Robust and Optimal Methods for Geometric Sensor Data Alignment

Dylan John Campbell

October 2018

A thesis submitted for the degree of
Doctor of Philosophy of The Australian National University

This thesis is an original work and has not been submitted to obtain a degree or diploma at any other university. To the best of my knowledge, it does not contain material previously published by another person, except where due reference is made.

Dylan John Campbell
28 October 2018

For Matthew

Acknowledgements

I would first like to thank my supervisors Lars Petersson, Laurent Kneip and Hongdong Li for their unstinting support, instructive guidance and dedication to the highest academic standards. I could not imagine a better or more complementary mix of talent, skills and experience for this research project and greatly value their contributions as supervisors, co-authors and mentors. Particular thanks are due to my primary supervisor, Lars Petersson, who guided my development as a researcher from the beginning. Your knack for assuaging any pessimism at negative results was much appreciated, as were your efforts to sustain the unity and morale of your research team.

More generally, I would like to acknowledge the computer vision community for making this area of research so dynamic and engaging. My thanks to those conference and journal reviewers whose conscientious work has significantly improved the quality of my research and to those authors who have generously responded to my emails about their work — it is much appreciated.

I would also like to acknowledge the various sources of funding that have assisted me throughout my studies. This research has been supported by an Australian Government Research Training Program (RTP) Scholarship and a NICTA PhD Supplementary Scholarship. It is inconceivable that this research project would have been attempted let alone completed without a reliable source of financial support. I am also grateful to NICTA, Data61 – CSIRO and the Australian National University for enriching my academic experience by providing conference travel funding.

Relocating to Canberra to pursue a research degree could have been a lonely experience. Fortunately, I moved to the close-knit community of Ursula Hall and met many people who have been an invaluable source of friendship, diversion and wisdom. Thank you also to my friends, colleagues and fellow tutors at ANU, NICTA and CSIRO who have shared the joys and frustrations of PhD and academic life — it has been a pleasure to work and learn with you. And to my coursework and research students, thank you for your curiosity and for making me a better teacher.

Finally, my heartfelt thanks to my family for supporting and encouraging me throughout my studies. And to Matthew, thank you for everything.

Abstract

Geometric sensor data alignment — the problem of finding the rigid transformation that correctly aligns two sets of sensor data without prior knowledge of how the data correspond — is a fundamental task in computer vision and robotics. It is inconvenient then that outliers and non-convexity are inherent to the problem and present significant challenges for alignment algorithms. Outliers are highly prevalent in sets of sensor data, particularly when the sets overlap incompletely. Despite this, many alignment objective functions are not robust to outliers, leading to erroneous alignments. In addition, alignment problems are highly non-convex, a property arising from the objective function and the transformation. While finding a local optimum may not be difficult, finding the global optimum is a hard optimisation problem. These key challenges have not been fully and jointly resolved in the existing literature, and so there is a need for robust and optimal solutions to alignment problems. Hence the objective of this thesis is to develop tractable algorithms for geometric sensor data alignment that are robust to outliers and not susceptible to spurious local optima.

This thesis makes several significant contributions to the geometric alignment literature, founded on new insights into robust alignment and the geometry of transformations. Firstly, a novel discriminative sensor data representation is proposed that has better viewpoint invariance than generative models and is time and memory efficient without sacrificing model fidelity. Secondly, a novel local optimisation algorithm is developed for nD–nD geometric alignment under a robust distance measure. It manifests a wider region of convergence and a greater robustness to outliers and sampling artefacts than other local optimisation algorithms. Thirdly, the first optimal solution for 3D–3D geometric alignment with an inherently robust objective function is proposed. It outperforms other geometric alignment algorithms on challenging datasets due to its guaranteed optimality and outlier robustness, and has an efficient parallel implementation. Fourthly, the first optimal solution for 2D–3D geometric alignment with an inherently robust objective function is proposed. It outperforms existing approaches on challenging datasets, reliably finding the global optimum, and has an efficient parallel implementation. Finally, another optimal solution is developed for 2D–3D geometric alignment, using a robust surface alignment measure.

Ultimately, robust and optimal methods, such as those in this thesis, are necessary to reliably find accurate solutions to geometric sensor data alignment problems.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Publications

The following publications have resulted from the work presented in this thesis:

- Campbell, D. and Petersson, L., 2015. An adaptive data representation for robust point-set registration and merging. In *Proceedings of the 2015 International Conference on Computer Vision* (Santiago, Chile, Dec. 2015), 4292–4300. IEEE. doi: 10.1109/ICCV.2015.488
- Yang, J.; Li, H.; Campbell, D.; and Jia, Y., 2016. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38, 11 (Nov. 2016), 2241–2254. doi: 10.1109/TPAMI.2015.2513405
- Campbell, D. and Petersson, L., 2016. GOGMA: Globally-Optimal Gaussian Mixture Alignment. In *Proceedings of the 2016 Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV, USA, Jun. 2016), 5685–5694. IEEE. doi: 10.1109/CVPR.2016.613 [Spotlight Presentation]
- Campbell, D.; Petersson, L.; Kneip, L.; and Li, H., 2017. Globally-optimal inlier set maximisation for simultaneous camera pose and feature correspondence. In *Proceedings of the 2017 International Conference on Computer Vision* (Venice, Italy, Oct. 2017), 1–10. IEEE. doi: 10.1109/ICCV.2017.10 [Oral Presentation]
- Campbell, D.; Petersson, L.; Kneip, L.; and Li, H., 2018. Globally-optimal inlier set maximisation for camera pose and correspondence estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (Jun. 2018), preprint. doi: 10.1109/TPAMI.2018.2848650

Abbreviations

BB	Branch-and-Bound
CPU	Central Processing Unit
DoF	Degrees of Freedom
GMM	Gaussian Mixture Model
GMA	Gaussian Mixture Alignment
GPU	Graphics Processing Unit
PnP	Perspective-n-Point
SVM	Support Vector Machine

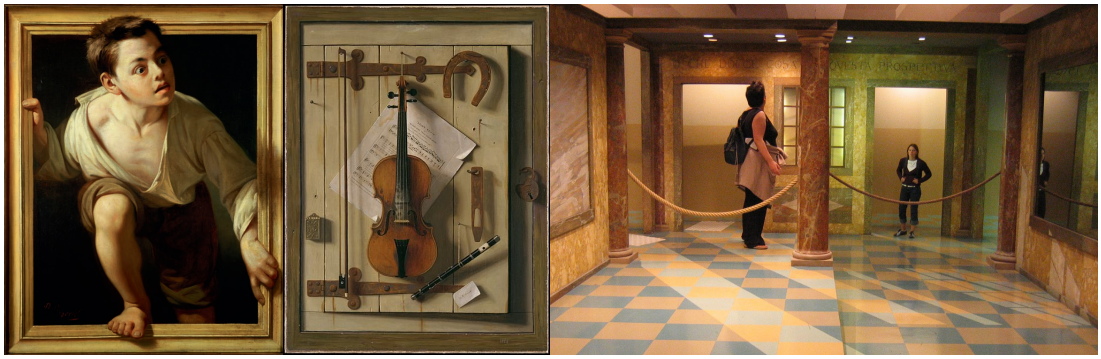
Chapter 1

Introduction

The human visual system is complex and multi-faceted, involving sensory apparatus — the retina containing the rods and cones in the eyes — and processing apparatus — a set of highly-connected neurons in the brain. It is capable of performing very sophisticated tasks, including detection, recognition, tracking, prediction, learning, colour constancy, noise removal, perception of depth and 3D structure, spatial awareness, and many other high-level tasks. Among the capabilities of the human visual system is three-dimensional perception and awareness. The ability to construct a model of the environment, situate oneself inside it, and comprehend how the 3D objects and structures therein fit together, corresponding to the tasks of mapping, localisation, route-planning and 3D interaction, are of significant evolutionary advantage.

While the human visual system is very well equipped to perform tasks that are required frequently, it has many limitations. As visual sensors, the eyes are restricted to the visible electromagnetic spectrum and 2D directional information, albeit including a sensitivity to both light and colour. As a visual processor, the visual cortex cannot directly perceive the 3D world, instead relying on a stereo pair of displaced 2D sensors, motion and past experience. This indirect 3D information may be incorrect, misleading or ambiguous, as certain illusions such as *trompe-l'œil* and forced perspective in Figure 1.1 illustrate. Other limitations of human vision processing arise from the fallibility of memory and recall, a propensity towards 'seeing' from memory rather than sensor stimulus, a tendency to take processing shortcuts and heuristics or make unwarranted extrapolations, and a susceptibility to optical illusions. Moreover, the visual system is subject to ocular and cortical deterioration and impairment, including the reduction of visual acuity, field-of-view, and colour perception, as well as disorders that affect higher-order visual processes such as face recognition.

In comparison to humans, several animals can perceive a wider range of the electromagnetic spectrum, including infrared (pit vipers) and ultraviolet (bees) frequencies. Other animals have additional sensory equipment which may complement or replace the visual system, including those that can sense the magnetic or electric fields, or use



(a) *Trompe-l'œil* artwork (b) Ames room

(c) Gallery in the Palazzo Spada (d) The Electric Brae

Figure 1.1: Geometric optical illusions exploit ambiguities in the human binocular vision system to subvert the viewer's perception of depth and 3D structure. (a) *Trompe-l'œil* artwork create a vivid illusion of depth by subverting viewer expectations and using perspective, foreshortening, and shading techniques. Attribution: Pere Borrell del Caso (left) and William Harnett (right). (b) An Ames room creates an illusion of size disparity by creating an apparent horizon that is not horizontal, typically by constructing a trapezoidal room that appears cubic from the viewing position. Attribution: mosso. (c) The forced perspective gallery by Francesco Borromini in the Palazzo Spada, Rome, is only 8m long but appears to be 37m due to the shrinking columns and sloping floor and ceiling. Attribution: Livioandronico2013. (d) The Electric Brae, a hill in Ayrshire, Scotland, has a road that slopes upwards but appears to be sloping downhill due to the geometry of the landscape. Attribution: Mary and Angus Hogg.

auditory input for depth and velocity perception, a form of sonar common to many bat species. While some of these fall outside the traditional visual system, the interconnection of different sensory inputs can make it difficult to separate the visual system from other sensor processing systems. Many animals, for example, incorporate information from their proprioceptive inner ears to make sense of visual input. However, there is always an energy trade-off between developing sophisticated sensory apparatus, especially those that perform active sensing, and other systems required for life and reproduction. This trade-off is, to an extent, also a feature of computer

vision and robotic systems. Nonetheless, the large variety of evolved visual and extra-visual systems indicates that advanced sensing can be a good strategy in the calculus of survival.

Computer vision systems attempt to emulate many of the capabilities of the human visual system, but do not have all the limitations inherent to it. For example, they have less fallible memory and recall processes; they can store much more data and expand their storage capacity when required; they can reason about the observable world in a mathematical framework without necessarily resorting to heuristics; they can make accurate measurements in 2D and 3D; and they can systematically identify ambiguities. Most importantly, they are not limited to the visible spectrum or the wider electromagnetic spectrum, and can perceive 3D information directly. Whatever can be detected by a sensor can be made accessible to a computer vision system. Common modalities include visual data, hyperspectral data, 3D positional data, proprioceptive data such as acceleration and angular velocity from an inertial measurement unit, and geospatial data such as absolute position from a GPS.

However, even with access to sources of information unavailable to the human visual system, current computer vision systems are in many ways inferior to that of humans and other animals. This is largely because visual information is a data-intensive and complex modality, despite the apparent ease with which humans process it. Visual data is information-rich, with even a single image containing a potentially enormous amount of information. For example, a one megapixel colour image consists of three million variables, each of which can typically attain one of 256 discrete values. Even with the processing power available to the modern computer, many computer vision algorithms find sensible ways to quickly reduce the amount of information in an image before performing the main computation. Furthermore, visual data is often highly complex, imaging cluttered scenes with multiple moving or stationary objects that occlude other objects or structures. Moreover, the appearance of objects may vary when viewed from different directions or with different illumination conditions. Another source of complexity is noise and distortions, a consequence of the physical measurements that constitute visual data. Finally, 2D images, the predominant form of visual data, are projections of 3D scenes into 2D, and the loss of dimensionality introduces geometric ambiguities, such as the optical illusions shown in Figure 1.1. Any of these commonplace complexities, which are inherent to the modality or violate an ideal vision model, may cause an insufficiently robust computer vision algorithm to fail. Other modalities used in computer vision share many of these same challenges, although the complexities of 3D reasoning can be simpler for 3D data such as point-sets or meshes.

A plurality of sensors have been used in computer vision systems, with the most common ones being the colour camera, the depth camera, and the laser rangefinder or

lidar sensor. The data types provided by these sensors have different properties: colour cameras generate images that collect structured directional measurements of colour and intensity at a low cost; depth cameras generate depth images that collect structured directional and depth measurements, which can be converted into positional information, using time-of-flight, structured light or stereopsis techniques; and lidar sensors generate point-sets that collect unstructured positional measurements and associated visual information such as reflectance, using time-of-flight or phase-shift techniques. The distinction between directional and positional sensor data is important for this thesis. To a lesser extent, so too is the distinction between structured data, such as an image, and unstructured data, such as a point-set, whose elements can be permuted without loss of information. Sensors and sensor data representations are discussed in more detail in Section 3.3.

There are two complementary sources of information provided by images and some point-sets: appearance and geometry. Appearance is the visual information, the way something looks, whereas geometry is the structural information, the shape and relative arrangement. While a robust computer vision system should integrate as many complementary sources of information as possible, it can be productive to investigate each in isolation. This thesis focuses on geometric information and geometric methods.

The problem considered in this dissertation is geometric sensor data alignment: the problem of finding the rigid transformation (rotation and translation) that correctly aligns one set of sensor data with another, without any prior knowledge about how the data correspond. In many cases this is undertaken by jointly solving for the transformation and correspondence set. The data may be of different dimensionality or captured using different sensors but must provide geometric information. That is, the spatial position or direction of each data element must be provided. The dual, interpreted loosely, of the alignment problem is the sensor pose estimation problem, where the focus is on recovering the pose instead of aligning the data. If one of the datasets, often denoted as the map, is registered to the world or reference coordinate frame, then the problem is absolute pose estimation. If there is no privileged frame of reference, it is relative pose estimation. An example alignment problem is shown in Figure 1.2. A vehicle has access to two sources of information: a pre-computed 3D model of the surrounding area, and a photo from a calibrated camera mounted on the vehicle. The problem is to align the 2D and 3D data in order to estimate the 6-DoF absolute pose of the camera and hence the vehicle.

In this chapter, the problem of geometric sensor data alignment is outlined in Section 1.1, the objective, scope and approach of this thesis are stated in Section 1.2, the key technical contributions are summarised in Section 1.3, and the thesis structure is outlined in Section 1.4.

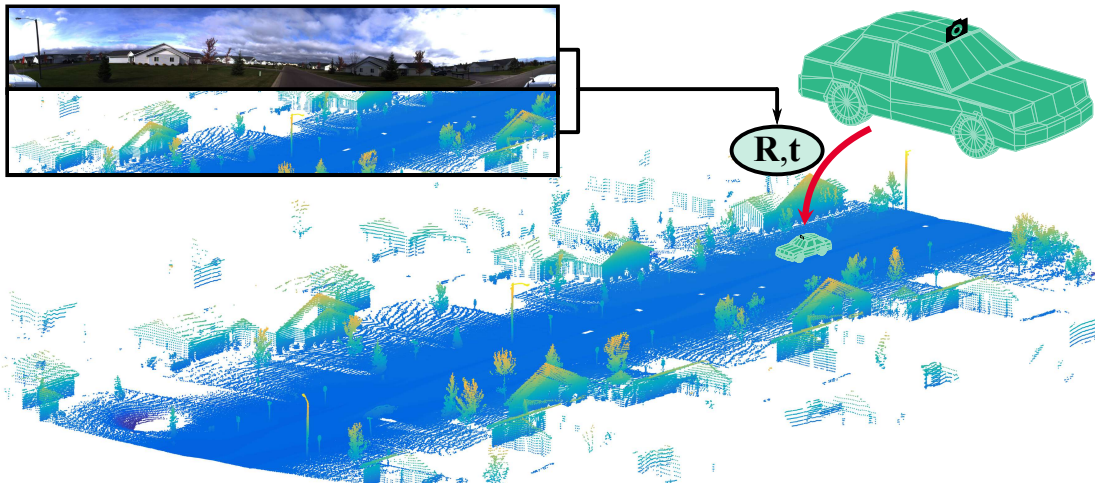


Figure 1.2: An example of geometric sensor data alignment. In this example, a vehicle has access to a pre-computed 3D model of the surrounding area and a photo from a calibrated camera mounted on the vehicle. The problem is to align the 2D and 3D data in order to estimate the pose of the camera and hence the vehicle.

1.1 The Geometric Sensor Data Alignment Problem

Geometric sensor data alignment is the problem of finding the rigid transformation (rotation and translation) that correctly aligns one set of sensor data with another, without any prior knowledge about how the data correspond. An ideal alignment solution would identify all outliers in the data and optimally align the inliers with respect to a geometric error criterion that accounts for noise, such as the L_2 error. Note that the terms alignment and registration are used interchangeably in this thesis.

The optimisation problem for geometric sensor data alignment can be written as follows. Given two sets of sensor data $\mathcal{X}_1$ and $\mathcal{X}_2$, a rigid transformation function T, and an objective function f that measures alignment quality, then

$$\begin{aligned} \underset{\mathbf{R},\mathbf{t}}{\text{optimise}} \quad & f(T(\mathcal{X}_1, \mathbf{R}, \mathbf{t}), \mathcal{X}_2) \\ \text{subject to} \quad & \mathbf{R} \in SO(n) \\ & \mathbf{t} \in \mathbb{R}^n \end{aligned} \tag{1.1}$$

where the rotation $\mathbf{R}$ and translation $\mathbf{t}$ are rigid transformations of nD Euclidean space. At the optimum, the arguments $\mathbf{R}^*$ and $\mathbf{t}^*$ constitute the aligning transformation or, from another perspective, the sensor pose. An example transformation function for a point-set $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^N$ is the application of the transformation $\mathbf{R}_\mathbf{r}\mathbf{p}_i + \mathbf{t}$ to each point. An example objective function for two point-sets $\mathcal{P}_1$ and $\mathcal{P}_2$ is the sum of the squared closest-point residuals $f(\mathcal{P}_1, \mathcal{P}_2) = \sum_{\mathbf{p}_1 \in \mathcal{P}_1} \min_{\mathbf{p}_2 \in \mathcal{P}_2} \|\mathbf{p}_1 - \mathbf{p}_2\|_2^2$.

1.1.1 Applications of Sensor Data Alignment

Geometric sensor data alignment is a fundamental task in computer vision, robotics, computer graphics and medical imaging. The underlying tasks of sensor data alignment and sensor pose estimation are themselves useful, and are frequently deployed as basic units in computer vision and robotic systems.

Applications of positional sensor data alignment in 2D and 3D are extensive. They include merging multiple partial scans into a complete model [Blais and Levine, 1995; Huber and Hebert, 2003]; using registration results as fitness scores for object recognition [Johnson and Hebert, 1999; Belongie et al., 2002]; registering a view into a global coordinate system for sensor localisation [Nüchter et al., 2007; Pomerleau et al., 2013]; fusing cross-modality data from different sensors [Makela et al., 2002; Zhao et al., 2005]; acquiring shape data [Gelfand et al., 2005; Aiger et al., 2008]; and finding relative poses between sensors [Yang et al., 2013a; Geiger et al., 2012]. Some higher-level applications include recording cultural heritage [Remondino, 2011], mapping underground mine sites [Magnusson et al., 2007], and Simultaneous Localisation And Mapping (SLAM) tasks in mobile robotics [Smith and Cheeseman, 1986; Leonard and Durrant-Whyte, 1991].

Applications of directional and positional sensor data alignment (2D–3D registration) are also numerous, since the ability to find the pose of a camera and map visual information onto a 3D model and vice versa is useful for many tasks. They include visual localisation [Nöll et al., 2011; Svärm et al., 2014, 2016]; camera pose estimation and tracking [Hartley and Kahl, 2009; Bazin et al., 2013; Kneip et al., 2015]; augmented reality [Marchand et al., 2016]; motion segmentation [Olson, 2001]; object recognition [Huttenlocher and Ullman, 1990; Mundy, 2006; Aubry et al., 2014]; automated cartography [Fischler and Bolles, 1981]; and hand–eye calibration for robotics [Horaud and Dornaika, 1995; Seo et al., 2009; Heller et al., 2012; Ruland et al., 2012]. Some higher-level commercial applications include autonomous vacuum cleaners such as the Dyson 360 Eye, and augmented reality platforms such as the Microsoft Hololens, the Oculus Rift, and the Google ARCore and Qualcomm Vuforia software development kits [Zia et al., 2016].

While not a focus of this thesis, directional sensor data alignment also has many applications. A commonly used application is panoramic image stitching [Bazin et al., 2013; Enqvist et al., 2015; Parra Bustos et al., 2016], where the homography relating two images can be obtained by rotation-only search if the camera is sufficiently distant from the scene. A commercial application in this domain is Google Photo Sphere, an application for creating 360° panoramas from photos.

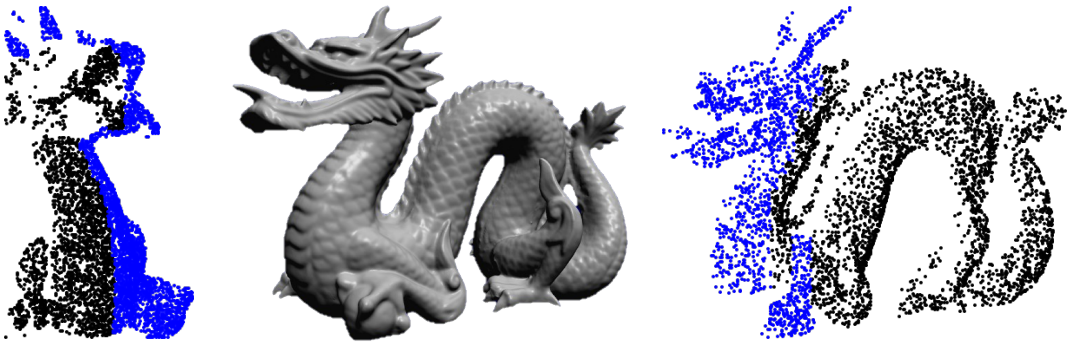


Figure 1.3: Two partially-overlapping observations (left and right) of the DRAGON model (middle) from the Stanford Computer Graphics Laboratory. The pair of point-sets has more structured outliers (in black) than inliers (in blue) because the overlapping region is small.

1.1.2 Key Challenges

There are two key challenges inherent to the alignment problem: outliers and non-convexity. The former arises from the sensor data, whereas the latter arises from the objective function and transformation.

Outliers are pervasive in sensor data and, for two sets of sensor data, consist of those data elements in each set that do not correspond to any element in the other set. They emanate from four major sources: sensor noise and error, sampling effects, changes in the scene and changes in the sensor viewpoint. The first two sources generate random outliers. For example, impulsive noise, multipath errors, and sparse or uneven sampling can produce random outliers. The last two sources generate structured outliers, which are typically more numerous. For example, a dynamic object may be absent in one dataset but present and occluding surfaces behind it in another, and parts of a scene may be visible from one viewpoint but absent or occluded from another. In real data, partially-overlapping observations are the most frequent and significant source of outliers, an example of which is shown in Figure 1.3. Outliers are problematic to alignment algorithms because alignment is a joint transformation and correspondence problem, and outliers invalidate the correspondence assumptions common to many alignment objective functions. That is, many objective functions do not model outliers or inadequately model them.

Non-convexity (or non-concavity for maximisation problems) is a property common to most useful alignment objective functions, as illustrated by Figure 1.4. Furthermore, rotation constraints also lead to non-convex optimisation problems. Hartley and Kahl [2007], for example, showed that many quasi-convex objective functions in multiple view geometry problems can be solved efficiently, unlike the many non-convex functions that arise when rotation parameters are to be solved. While it may be relatively

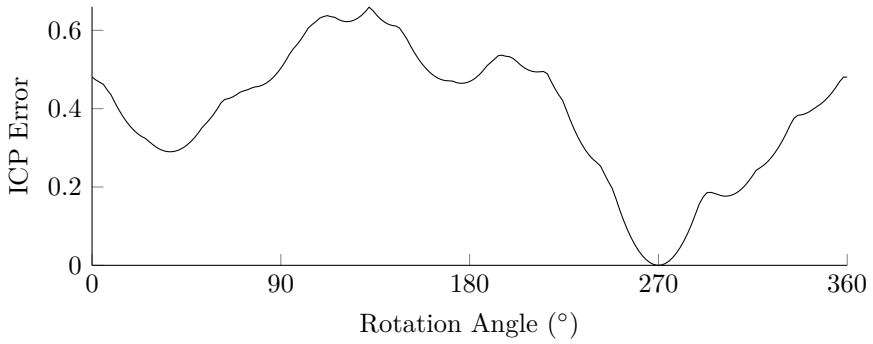


Figure 1.4: Alignment objective function non-convexity. In this example, the ICP objective function [Besl and McKay, 1992] is plotted for 10 random 2D points undergoing a rotation-only transformation with no outliers. The non-convexity of problems with outliers, sensor data of higher dimension, or transformations with higher degrees-of-freedom is even more pronounced.

straightforward to find a local optimum of a non-convex alignment function, finding the global optimum is a hard optimisation problem.

Although not necessarily inherent to the alignment problem, tractability is another challenge for alignment algorithms. In particular, many optimal algorithms developed to circumvent the non-convexity problem have exponential time complexity and therefore cannot be used for large datasets. However, recent work [Straub et al., 2017; Campbell et al., 2017] has shown that under a slightly weakened optimality condition, these algorithms can run in polynomial time. Moreover, sophisticated data structures, data representations and parallel processing [Yang et al., 2016; Parra Bustos et al., 2016; Campbell and Petersson, 2016] can greatly reduce the computational cost.

1.1.3 Existing Alignment Approaches

In this section, the state of the art in geometric alignment methods will be briefly outlined, with an emphasis on how the literature handles the key challenges of outliers and non-convexity as identified above. A full classification and review of the literature is given in Chapter 2.

Algorithms that solve the alignment problem can be classified into two groups: those that require a set of putative correspondences between elements of the sensor datasets [Horn, 1987; Fischler and Bolles, 1981; Enqvist et al., 2009; Lepetit et al., 2009; Svärm et al., 2016; Sattler et al., 2017] and those that do not [Besl and McKay, 1992; Fitzgibbon, 2003; Aiger et al., 2008; Breuel, 2003; Li and Hartley, 2007; Yang et al., 2016; David et al., 2004; Brown et al., 2015]. This thesis focuses on solutions to the more challenging and general problem of alignment without correspondences, designated as *geometric matching* problems by Breuel [2003].

Foundational solutions to geometric matching problems applied local optimisation techniques to non-robust objective functions and were therefore susceptible to local optima and outliers. That is, the methods were liable to find only a locally-optimal solution to a non-convex objective function whose measure of alignment quality was unreliable when outliers were present in the data. The Iterative Closest Point (ICP) algorithm, proposed by Besl and McKay [1992], became the technique *de rigueur* for aligning positional sensor data due to its conceptual simplicity, ease of use and good performance. It alternated between finding closest-point correspondences given the current transformation and then finding the least squares transformation given the current correspondences. For aligning directional and positional sensor data (2D–3D alignment), the SoftPOSIT algorithm, proposed by David et al. [2004], was considered the most computationally efficient approach [Moreno-Noguer et al., 2008]. It alternated between probabilistic correspondence assignment given the current transformation and iterative transformation update given the current correspondences, using a least squares objective function.

To improve the robustness of these methods to outliers, different alignment objective functions have been advanced. Fitzgibbon [2003] proposed Levenberg-Marquardt ICP (LM-ICP), extending the ICP algorithm into the LM optimisation framework [Moré, 1978]. Modelling registration as a general optimisation problem enabled the use of robust Lorentzian and Huber error functions that attenuate the influence of outlier correspondences. The family of probabilistic alignment approaches [Chui and Rangarajan, 2003; Myronenko and Song, 2010; Tsin and Kanade, 2004; Jian and Vemuri, 2011] also enabled the use of robust objective functions. In particular, modelling point-sets as probability distributions permits the closed-form L_2 distance between densities recommended by Jian and Vemuri [2011]. For 2D–3D alignment, Moreno-Noguer et al. [2008] proposed the BlindPnP algorithm, which represented the region of expected transformations (the pose prior) as a Gaussian mixture model from which a Kalman filter was initialised to guide a local pose search routine. It derived its robustness to outliers from a correspondence hypothesising step that was similar to the RANdom SAmple Consensus (RANSAC) algorithm [Fischler and Bolles, 1981] but was restricted to the subset of matches that were plausible from that pose guess.

Global optimisation approaches to geometric matching problems endeavour to avoid incorrect locally-optimal alignments by expanding the search domain. Furthermore, optimality can be guaranteed by applying the Branch-and-Bound (BB) paradigm [Land and Doig, 1960], conferring immunity to the non-convexity of the problem. Breuel [2003] introduced the use of BB to optimally solve 2D geometric alignment problems, proposing a family of bounding functions for different transformations and objective functions. However, he identified tractability as the primary impediment to extending

the work to 3D sensor data. Ceding outlier robustness was an early strategy to extend optimality to 3D alignment problems. For example, Li and Hartley [2007] minimised a non-robust Lipschitzised L_2 error function using BB, but assumed the transformation was pure rotation and that the point-sets were the same size with no outliers. More recently, Yang et al. [2016] proposed the Go-ICP algorithm for 6-DoF 3D–3D alignment, which found the optimal solution to the closest-point L_2 error between point-sets using a nested BB scheme. Brown et al. [2015] proposed a similar solution for 6-DoF 2D–3D alignment. The last two methods proposed a trimming strategy to compensate for their non-robust objective functions, however this required the inlier fraction to be specified, which can rarely be known in advance. If the inlier fraction is over- or under-estimated, the trimmed objective function may become distorted such that the location of the global optimum does not occur at the correct pose.

While RANSAC approaches [Fischler and Bolles, 1981; Grimson, 1990], which randomly hypothesise a minimal set of correspondences, compute a transformation and evaluate its quality, are global search methods that can confer outlier robustness, they do not guarantee optimality and quickly become intractable as the number of points and outliers increases. The first globally-optimal 6-DoF geometric matching methods for 3D sensor data with inherently robust objective functions were proposed in Parra Bustos et al. [2016], Campbell and Petersson [2016] and Campbell et al. [2017], the latter two of which are presented in Chapters 5 and 6 of this thesis. Using the Go-ICP framework, Parra Bustos et al. [2016] optimised the robust inlier set cardinality maximisation objective function for optimal 3D–3D alignment. They achieved efficient runtimes by exploiting stereographic projection techniques and sophisticated data structures including circular R-trees and matchlists.

This brief summary of the geometric alignment literature highlights the need for methods that are intrinsically robust to outliers and are not susceptible to local optima. In the next section, these considerations will be formalised into an objective to be pursued in this thesis.

1.2 Objective and Approach

In this section, the specific objective, scope and approach of this thesis will be outlined. That is, the section will address the questions of what will and will not be investigated, why the research is needed, and how it will be undertaken.

1.2.1 Objective

The objective of this thesis is *to develop tractable algorithms for geometric sensor data alignment that are robust to outliers and not susceptible to spurious local optima.*

This is a beneficial undertaking because alignment is a basic tool of computer vision and robotics, used for sensor localisation and fusing geometric sensor data. Furthermore, outliers are highly prevalent in sensor data and alignment problems are highly non-convex. As previously shown, these considerations are not fully addressed by the literature. Hence, robust and optimal methods are necessary for geometric sensor data alignment to handle unknown correspondences, outliers and non-convexity.

1.2.2 Scope

The scope of this thesis is restricted in several ways. Firstly, the alignment of directional sensor data, which corresponds to rotation search, relative camera pose and structure-from-motion problems, is not addressed in this thesis. Nonetheless, rotation search problems, such as image stitching, can be solved by the 2D–3D alignment algorithm developed in Chapter 6 by disabling translation search. In addition, the alignment of directional and positional sensor data is limited to the 2D–3D alignment problem and does not extend to arbitrary dimensions. Secondly, the raw sensor data types are restricted to colour or greyscale images, depth images and point-sets. Since these are common formats for raw data, this is not an onerous limitation. Finally, the scope is restricted to rigid transformations, to the exclusion of affine, projective, piecewise-rigid and non-rigid transformations. However, the nD–nD alignment algorithm developed in Chapter 4 can be trivially extended to non-rigid and other transformations, using the approach of Jian and Vemuri [2011]. Extending the algorithms of Chapters 5 and 6 to non-rigid transformations would not be tractable, since the dimensionality of the problem is already very high for a branch-and-bound approach.

1.2.3 Approach

The approach taken in this thesis is to consider the challenges presented by outliers and non-convexity from the outset when developing geometric alignment algorithms. That is, robustness to outliers is built into the algorithms through intrinsically robust objective functions, and susceptibility to incorrect local optima is reduced by expanding the basin of convergence (Chapter 4) or global optimisation (Chapters 5 and 6). In particular, the branch-and-bound algorithm is exploited to ensure that the optimal solution is found, which is shown to be highly desirable for many geometric alignment problems. As such, techniques for improving the efficiency of 6-DoF branch-and-bound search are developed in order to produce tractable algorithms. A variety of objective functions, algorithm structures, and implementation techniques are sampled in Chapters 4–6, which are not intended to be exhaustive but to demonstrate different possibilities for geometric sensor data alignment.

1.3 Summary of Contributions

The major contributions of this thesis are:

1. A novel positional sensor data representation, the Support Vector–parametrised Gaussian Mixture (SVGM), with a sparse parametrisation that is adaptive to local surface complexity. As a discriminative model, it is more invariant to viewpoint than a generative model since it does not model sampling artefacts, such as distance-dependent point density and occlusions. See Chapter 4.
2. A novel local optimisation algorithm, Support Vector Registration (SVR), for aligning positional sensor data under the robust L_2 distance between densities, which manifests strong robustness to outliers and sampling artefacts, and a wide region of convergence. See Chapter 4.
3. A novel global optimisation algorithm, Globally-Optimal Gaussian Mixture Alignment (GOGMA), for optimally aligning 3D positional sensor data under the robust L_2 distance between densities. GOGMA is the first optimal solution proposed for 3D–3D alignment with an inherently robust objective function. The pivotal contribution is the derivation of novel bounds on the objective function using the geometry of $SE(3)$. See Chapter 5.
4. A novel global optimisation algorithm, Globally-Optimal Pose And Correspondences (GOPAC), for optimally aligning 2D directional and 3D positional sensor data under the robust inlier set cardinality objective function. GOPAC is the first optimal solution proposed for 2D–3D alignment with an inherently robust objective function. The pivotal contribution is the derivation of novel bounds on the objective function using the geometry of $SE(3)$. See Chapter 6.
5. A novel global optimisation algorithm for optimally aligning 2D directional and 3D positional sensor data under the robust L_2 distance between densities. A novel projection of Gaussian mixture models onto the unit sphere is derived and analysed, and novel bounds on the closed-form objective function are found. See Chapter 6.

1.4 Thesis Outline

This thesis comprises seven chapters. In Chapter 2, the literature on geometric sensor data alignment is classified, outlined, and reviewed in order to determine the state of the art. The historical progression of alignment methods is charted to establish a context for the problem and the strengths and limitations of these methods are evaluated. Particular attention is paid to how these methods handle outliers and non-convexity. Chapter 3 presents the technical background for the geometric sensor data alignment

problem. This background material consists of elements that are common to many of the approaches proposed in later chapters, including rigid motion parametrisations, distance measures, sensor data representations, objective functions and optimisation techniques, and form a mathematical toolkit that will be referred to repeatedly.

The next three chapters propose novel geometric alignment algorithms for different types of sensor data. In Chapter 4, the problem of robustly aligning two sets of 2D or 3D positional sensor data, such as laser scans, is considered. An algorithm, Support Vector Registration (SVR), is proposed for robustly aligning positional sensor data using a Support Vector–parametrised Gaussian mixture data representation. Chapter 5 extends the investigation of robust data representations and objective functions to the optimal 3D–3D geometric alignment problem. An algorithm, Globally-Optimal Gaussian Mixture Alignment (GOGMA), is proposed for robust and optimal 3D positional sensor data alignment using the branch-and-bound framework with tight and novel bounds. In Chapter 6, the same framework is applied to the 2D–3D geometric alignment problem. Novel bounds are derived for the robust inlier set cardinality objective function and an algorithm, Globally-Optimal Pose And Correspondences (GOPAC), is proposed for solving the camera pose estimation problem. In addition, the theoretical framework is transferred to another robust objective function that measures the distance between mixture models on the sphere, linking back to mixture model approaches recommended in Chapter 4.

Finally, Chapter 7 summarises the main contributions of the thesis, collects the inferences made throughout, and discusses ongoing and future work stemming from this research.

Chapter 2

Literature Review

This chapter outlines and reviews the literature on geometric sensor data alignment. The two key challenges identified by the literature as inherent to the alignment problem are outliers and non-convexity. Outliers are pervasive and predominantly arise in sensor data from sampling scenes or objects from different viewpoints, resulting in occlusions and partial overlap. Non-convexity or non-concavity is a property common to all alignment objective functions and, while it may be relatively straightforward to find a local optimum of a non-convex alignment function, finding the global optimum is a hard optimisation problem. Accordingly, two clear trends emerge from the literature: towards a greater level of robustness to outliers and towards more sophisticated global optimisation strategies. Moreover, recent advances in applying global optimisation techniques to the alignment problem suggest that guaranteed optimal solutions can be both tractable and highly desirable. The aims of this chapter are to chart the historical progression of alignment methods to establish a context for the problem and to evaluate their strengths and limitations, with an especial focus on how they handled outliers and non-convexity. The techniques identified here to reduce the susceptibility of alignment algorithms to outliers and local optima form the technical background to the solutions proposed for several geometric alignment problems in Chapters 4, 5 and 6. In addition to this detailed survey, each of these chapters contain a summary of the work relevant to the chapter in order to motivate the specific problems addressed and re-introduce the state-of-the-art approaches.

The literature on geometric sensor data alignment can be grouped based on the type of geometric information provided by each sensor. Geometric measurements from a sensor can be positional or directional, that is, containing the spatial position or direction of the sample with respect to the sensor. Hence there are three logical subdivisions of geometric alignment problems: those where both sets of sensor data are positional, those where both sets are directional, and those where one set is positional and the other is directional. In this thesis, the first and last of these subdivisions are considered, which correspond to the point-set registration and absolute camera pose

problems. The alignment of directional sensor data, which corresponds to the relative camera pose and structure-from-motion problems, has a very large body of literature but is not the focus of this work.

The next level of structure arises from whether or not the alignment problem presupposes a set of correspondences between elements of the sensor datasets. For the first class of problem, methods for generating correspondences will be surveyed briefly before considering methods for aligning the data given the correspondences. The second class of problem, where a correspondence set is not available, is much more challenging. Methods that address this problem must simultaneously solve for the transformation and the correspondences between the sets of sensor data, although the correspondences need not be explicit. Moreover, this class of problem is more general than the former, with solutions to the first class being frequently used as sub-routines in approaches that handle the second class.

Finally, the third level of structure relates to the type of optimisation used by the alignment solution, including local, global and globally-optimal search techniques. Local optimisation methods apply local changes in parameter space to find the local optimum whose basin of convergence contains the parameter set at which the algorithm was initialised. Global optimisation methods endeavour to find the global optimum by searching over larger regions of parameter space. Globally-optimal methods are a subset of global optimisation methods that provide a guarantee that the global optimum will be attained within some specified precision.

2.1 Aligning Positional Sensor Data

The task of aligning two sets of positional sensor data has been studied extensively in the computer vision and robotics communities. While the general registration problem is not limited to 2D and 3D modalities, these modalities predominate in the literature due to their many practical applications, including merging multiple partial scans into a complete model [Blais and Levine, 1995; Huber and Hebert, 2003]; recognising objects using measures of registration quality [Johnson and Hebert, 1999; Belongie et al., 2002]; registering a single view into a global coordinate system for sensor localisation [Nüchter et al., 2007; Pomerleau et al., 2013]; fusing cross-modality data from different sensors [Makela et al., 2002; Zhao et al., 2005]; and finding the relative pose between sensors [Yang et al., 2013a; Geiger et al., 2012]. The general problem also encompasses both rigid and non-rigid registration, where the latter refers to the alignment of sensor data sampled from deformable objects such as human bodies. However, this review will focus primarily on the simpler case of finding the 3 or 6 degrees-of-freedom rigid transformation between sets of positional sensor data.

In this section, the literature will be divided into whether or not the alignment problem presupposes a set of correspondences between elements of the datasets. Methods for aligning sensor data given a potentially noisy correspondence set will be surveyed first, followed by methods for addressing the more challenging and general problem of simultaneously solving for the transformation and the correspondences. For both types, the progression towards methods that are more robust to noise, outliers and local optima will be highlighted.

2.1.1 Alignment With Correspondences

A large body of work exists for solving the problem of aligning positional sensor data when correspondences are available. For this problem, the literature identifies noise and outliers in the correspondence set as the primary confounding factors in real data. In addition, significant effort has gone towards developing optimal solutions that are insensitive to noise and outliers. However, before these approaches can be applied, a set of putative correspondences between the sensor data must be found.

Generating Correspondences

Keypoint or feature detection and feature description techniques provide a relatively robust and repeatable way to detect interest points such as edges or corners and compute likely correspondences between them. The recent literature on 3D keypoint detectors and feature descriptors is extensive, motivated in part by efforts to solve the correspondence problem for deformable or articulated objects such as the human face and body, and will be only briefly summarised here. For 3D sensor data, including point-sets, meshes and depth images, keypoint detectors and and feature descriptors were surveyed and thoroughly evaluated in Tombari et al. [2013] and Boyer et al. [2011] with respect to distinctiveness and repeatability, focussing on the applications of 3D object recognition and 3D shape retrieval respectively. Methods for keypoint or feature detection include Intrinsic Shape Signatures (ISS) [Zhong, 2009], Mesh-DoG [Zaharescu et al., 2009], Heat Kernel Signatures (HKS) [Sun et al., 2009], Normal Aligned Radial Features (NARF) [Steder et al., 2011], ShapeMSER [Litman et al., 2011], and those derived from 2D image detectors such as the Harris operator [Harris and Stephens, 1988; Sipiran and Bustos, 2010] and the Scale-Invariant Feature Transform (SIFT) [Lowe, 2004; Maes et al., 2010]. Several of these detectors (ISS, HKS, NARF) also provide feature descriptors. Methods for feature description include Spin Images [Johnson and Hebert, 1999], 3D Shape Context [Frome et al., 2004], Point Feature Histograms (PFH) [Rusu et al., 2008b], Fast PFH [Rusu et al., 2009], Mesh-HoG [Zaharescu et al., 2009], Signatures of Histograms of OrienTations (SHOT) [Tombari

et al., 2010], Scale-Invariant Heat Kernel Signatures [Bronstein and Kokkinos, 2010], Wave Kernel Signatures (WKS) [Aubry et al., 2011], Blended Intrinsic Maps (BIM) [Kim et al., 2011], Intrinsic Shape Context [Kokkinos et al., 2012], Scale-Invariant Spin Image [Darom and Keller, 2012] and Local Depth SIFT [Darom and Keller, 2012]. However, the performance of these handcrafted descriptors has been largely surpassed by machine learning approaches.

More recently, many learning-based approaches for feature description and correspondence have emerged [Taylor et al., 2012; Pons-Moll et al., 2015; Windheuser et al., 2014; Litman and Bronstein, 2014; Rodolà et al., 2014; Boscaini et al., 2016b; Masci et al., 2015; Boscaini et al., 2015, 2016a; Vestner et al., 2017]. Rodolà et al. [2014] proposed a random forest approach, applied to WKS features, to learn correspondences. In contrast, Litman and Bronstein [2014] generalised the HKS and WKS feature descriptors by learning optimal spectral descriptors, surpassing the performance of the handcrafted features. This was extended in Boscaini et al. [2016b] using anisotropic spectral kernels. Finally, Convolution Neural Networks (CNNs) were applied to this problem in the Geodesic CNN architecture [Masci et al., 2015], followed by Localised Spectral CNNs [Boscaini et al., 2015] and Anisotropic CNNs [Boscaini et al., 2016a], which can be applied to both meshes and point-sets.

The correspondence problem is solved inherently by many of these approaches [Rodolà et al., 2014; Masci et al., 2015; Boscaini et al., 2015, 2016a; Vestner et al., 2017]. However, the traditional approach, surveyed in Van Kaick et al. [2011], is to perform nearest neighbour matching in descriptor space using some similarity measure. For rigid alignment, additional constraints that preserve the distances between points can be applied. For non-rigid deformations, some measure of distortion between the shapes is minimised.

From Correspondences to Alignment

The minimum number of correspondences required to find the rigid transformation between two sets of positional sensor data is two for 2D data and three for 3D data [Horn, 1987]. Given point correspondences $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ that are related by the rigid transformation $\mathbf{y}_i = \mathbf{R}\mathbf{x}_i + \mathbf{t}$, finding the rotation $\mathbf{R}$ and translation $\mathbf{t}$ is equivalent to finding the transformation that relates the underlying Cartesian coordinate systems. For the 3D case, Horn [1987] proposed constructing a triad from three corresponding non-collinear points in each coordinate system and then the rotation that aligns the triads is also the rotation that relates the underlying coordinate systems. The triad

$(\hat{\mathbf{i}}_x, \hat{\mathbf{j}}_x, \hat{\mathbf{k}}_x)$ associated with the point-set $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ was defined as

$$\hat{\mathbf{i}}_x = \frac{\mathbf{x}_2 - \mathbf{x}_1}{\|\mathbf{x}_2 - \mathbf{x}_1\|} \tag{2.1}$$

$$\hat{\mathbf{j}}_x = \frac{\mathbf{x}_3 - \mathbf{x}_1 - [(\mathbf{x}_3 - \mathbf{x}_1) \cdot \hat{\mathbf{i}}_x]\hat{\mathbf{i}}_x}{\|\mathbf{x}_3 - \mathbf{x}_1 - [(\mathbf{x}_3 - \mathbf{x}_1) \cdot \hat{\mathbf{i}}_x]\hat{\mathbf{i}}_x\|} \tag{2.2}$$

$$\hat{\mathbf{k}}_x = \hat{\mathbf{i}}_x \times \hat{\mathbf{j}}_x. \tag{2.3}$$

The triad $(\hat{\mathbf{i}}_y, \hat{\mathbf{j}}_y, \hat{\mathbf{k}}_y)$ associated with the point-set $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^N$ was defined analogously. The rotation was then given by $\mathbf{R} = |\hat{\mathbf{i}}_y\hat{\mathbf{j}}_y\hat{\mathbf{k}}_y||\hat{\mathbf{i}}_x\hat{\mathbf{j}}_x\hat{\mathbf{k}}_x|^\intercal$ where the $|\cdot|$ terms are the matrices formed by adjoining the column vectors. The translation was found from any correspondence using $\mathbf{t} = \mathbf{y}_i - \mathbf{R}\mathbf{x}_i$. However, when the measurements are not exact, a least-squares solution is more accurate, since there are more constraints than unknown parameters even with the minimal number of correspondences.

When measurement noise is present in the data, a least-squares objective function can be used to find a more accurate solution. Least-squares methods minimise the sum of squared residuals, given by

$$\min_{\mathbf{R} \in SO(n),\, \mathbf{t} \in \mathbb{R}^n} \sum_{i=1}^{N} \|\mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i\|^2 \tag{2.4}$$

for dimension n. Arun et al. [1987] proposed a least-squares solution using a Singular Value Decomposition (SVD) of the cross-covariance matrix of the datasets. Another closed-form solution was proposed by Horn [1987] using eigendecomposition and unit quaternions to represent rotations. Although least-squares is useful when measurement noise is present, it is not robust to outliers in the correspondence set. A high outlier rate is common in real 3D data, since 3D feature detection and description techniques are still less accurate than their 2D counterparts [Tombari et al., 2013], particularly when the sampling resolution is insufficient or the overlapping region is small.

When outliers are present in the correspondence set, the L_2 norm used in least-squares can be replaced by a robust loss function. A recent solution was proposed by Zhou et al. [2016], where a scaled Geman–McClure loss function is used to attenuate the influence of outlier correspondences. A simulated annealing approach on the Geman–McClure parameter smooths the non-convex objective function at the start of the algorithm, allowing many correspondences to participate in the optimisation, before gradually introducing non-convexity to encourage a tight alignment.

An alternative approach to outliers in the correspondence set is to maximise the consensus set of correspondences, for which greedy algorithms [Johnson and Hebert, 1999; Rusu et al., 2008a], Hough transforms [Woodford et al., 2014], game theory

[Albarelli et al., 2010], the RANdom SAmple Consensus (RANSAC) framework [Fischler and Bolles, 1981; Torr and Zisserman, 2000; Chum and Matas, 2008], or robust global optimisation [Gelfand et al., 2005; Olsson et al., 2008; Hartley and Kahl, 2009; Enqvist et al., 2009, 2012; Bazin et al., 2012; Ask et al., 2013; Enqvist et al., 2015] can be used. Maximising the cardinality of the consensus set $\mathcal{I}$ for an inlier threshold θ with respect to a distance function d (typically the Euclidean distance) is given by

$$\begin{aligned} \max_{\mathbf{R},\mathbf{t}} \quad & |\mathcal{I}| \\ \text{subject to} \quad & \mathcal{I} = \{i \in [1, N] \mid d(\mathbf{R}\mathbf{x}_i + \mathbf{t}, \mathbf{y}_i) \leqslant \theta\} \\ & \mathbf{R} \in SO(n),\, \mathbf{t} \in \mathbb{R}^n. \end{aligned} \tag{2.5}$$

Greedy algorithms are commonly used to find a reasonable consensus set, although they do not guarantee optimality. Johnson and Hebert [1999] proposed a feature-based alignment method that exploited the transformation invariance of a local descriptor, the spin image, to build sparse feature correspondences using a greedy algorithm with geometric consistency constraints. A similar approach was used by Rusu et al. [2008a], where a subset of distinctive PFH descriptors were used as an input for a greedy alignment algorithm. Geometric constraints between the features were incorporated by choosing spatially-constrained correspondences. Hough transforms can also be used to determine the most likely transformation given a set of correspondences with outliers. Feature correspondences are converted into votes in a Hough space parametrised by the transformation and the modes in Hough space are selected, corresponding to the most likely transformations. Woodford et al. [2014] showed that the process can be both tractable and accurate for alignment tasks. Finding a consensus set of correspondences can also be cast in a game-theoretic framework. Albarelli et al. [2010] proposed a non-cooperative game that, starting with a set of feature correspondences, invoked a natural selection process where correspondence sets that satisfied a mutual rigidity constraint thrived while incompatible correspondences were eliminated.

The RANSAC framework was proposed by Fischler and Bolles [1981] for robust estimation in computer vision problems. It aims to maximise the cardinality of the consensus set by stochastically generating solution hypotheses from minimal sets. For 3D positional sensor data alignment, RANSAC can be applied by sampling $n = 3$ point correspondences, computing the transformation hypothesis using Horn's method [Horn, 1987], transforming the point-set and counting the number of inliers. The number of iterations K of this process can be estimated if the ratio of inliers to correspondences w is known and is given by

$$K = \frac{\log(1-p)}{\log(1-w^n)} \tag{2.6}$$

where p is the probability that at least one minimal set of size n contains only inlier correspondences. Hence, $K = 9206$ iterations are required to almost certainly ($p = 0.9999$) find a minimal set containing only inlier correspondences, for an inlier ratio of $w = 10\%$. It can be shown that the number of iterations increases exponentially with respect to the outlier ratio $(1 - w)$. Moreover, RANSAC is a stochastic method that does not provide any guarantee of optimality. Rusu et al. [2009] applied RANSAC to the alignment problem, incorporating geometric constraints on the pairwise distances between the points in the minimal set, a kd-tree to find nearest neighbour correspondences in feature descriptor space, and a robust Huber loss function to measure the quality of the transformation. Compared to a previously developed greedy algorithm [Rusu et al., 2008a], this approach was two orders of magnitude faster and converged to the global optimum more frequently.

Polynomial-time algorithms have also been proposed for the problem of optimally maximising the cardinality of the consensus set [Olsson et al., 2008; Enqvist et al., 2012; Ask et al., 2013; Enqvist et al., 2015]. The authors observed that the solution to the consensus set maximisation problem is identical to the solution of the same problem on a subset of the correspondences, of size $\leqslant d$, the dimensionality of the parameter space. The optimal transformation is hence found by enumerating all $\binom{N}{k}$ subsets of correspondences for $k \leqslant d$ and examining transformations at the Karush-Kuhn-Tucker (KKT) [Enqvist et al., 2012] or Fritz-John (FJ) [Ask et al., 2013; Enqvist et al., 2015] points related to the subset. However, full 3D–3D rigid registration is not addressed by these methods, with quasiconvexity requirements [Olsson et al., 2008] or tractability [Enqvist et al., 2012; Ask et al., 2013; Enqvist et al., 2015] precluding 6-DoF registration. Even for a small correspondence set with $N = 100$, the number of subsets to enumerate for 3D–3D registration ($d = 6$) is 1 271 427 896, although in practice not all subsets need to be examined for every problem. As a result, these methods are more appropriate for applications such as 2D registration, triangulation, image stitching and relative pose.

Branch-and-Bound (BB) algorithms for optimally maximising the cardinality of the consensus set have also been proposed. Gelfand et al. [2005] employed BB to assign each feature point in one point-set with the optimal corresponding feature point in the other set in order to minimise the overall pairwise distance error. To make the search more efficient, they used rigid transformation constraints and a measure of correspondence quality based on intrinsic quantities of the data (internal pairwise distances). Enqvist et al. [2009] similarly applied BB to find optimal correspondences by formulating consensus set maximisation as an NP-hard graph vertex cover problem using pairwise consistency constraints. Li [2009] reformulated the consensus set problem as a mixed integer program that was solved using BB, with bounds computed from convex

under-estimators of the mixed integer program. However, this approach was limited to a linear algebraic error function and linear constraints, and was therefore unable to handle nonlinear rotation models. A faster BB approach was proposed by Bazin et al. [2012], extending the rotation search algorithm of Hartley and Kahl [2009] to solve the consensus set maximisation problem. However, the bounds are not tight and the approach is only applicable for rotation search problems, such as line clustering and vanishing point detection. Moreover, these BB methods have exponential worst-case time complexity.

In contrast to the work in this section, the solutions to positional sensor data alignment proposed by this thesis in Chapters 4 and 5 do not require correspondences. As a consequence, they are not reliant on the robust detection and description of geometric features, which is an unsolved problem for unstructured 3D data, particularly when it is noisy, occluded or not densely sampled. The next section investigates the family of methods for positional sensor data alignment without correspondences, a family that includes the solutions developed in this thesis.

2.1.2 Alignment Without Correspondences

When a set of correspondences is not available and is difficult to obtain, the problem becomes much more challenging, as identified by Hartley and Kahl [2009]. Methods that address this problem must jointly solve for the transformation and the correspondences between the sets of positional sensor data. However, most algorithms do not explicitly search over correspondence and transformation space simultaneously, instead assuming that the correspondences determine the transformation or vice versa. Moreover, the correspondences themselves need not be explicit, with soft or probabilistic assignment being frequently applied.

In addition to being more challenging, the correspondence-free class of problem is more general than the class that assumes correspondences are known *a priori* and therefore can be applied in more situations. Furthermore, solutions to the problem of alignment given correspondences, such as Horn's method [Horn, 1987], can be used as time-efficient sub-algorithms in correspondence-free approaches after fixing the current correspondence set. This can be helpful for finding a locally-optimal transformation before re-optimising over the correspondences.

The alignment of positional sensor data without correspondences has previously been addressed for 2D–2D and 3D–3D geometric matching problems [Besl and McKay, 1992; Fitzgibbon, 2003; Myronenko and Song, 2010; Jian and Vemuri, 2011; Irani and Raghavan, 1999; Aiger et al., 2008; Yang et al., 2016]. While the non-rigid alignment of deformable objects has received significant attention [Van Kaick et al., 2011], this review

will focus on the 3- and 6-DoF rigid alignment problems in 2D and 3D respectively.

The methods for aligning positional sensor data without correspondences can usefully be classified based on the type of optimisation. The following sections will examine approaches that employ local, global and globally-optimal search techniques.

Local Optimisation

The Iterative Closest Point (ICP) algorithm [Besl and McKay, 1992; Chen and Medioni, 1992; Zhang, 1994] is the dominant solution for positional sensor data alignment without correspondences due to its conceptual simplicity, ease of use and good performance. Given an initial transformation, the algorithm alternates between constructing a correspondence set under the current transformation and estimating the transformation given these correspondences, until convergence. The correspondence set is generated by choosing, for each point in one point-set, its Euclidean nearest neighbour in the other point-set. While in general the nearest neighbour is not the real corresponding point, ICP nonetheless often converges to a reasonable solution. The transformation is estimated by minimising the sum of squared distances between corresponding points, using a closed-form solution such as Horn's method [Horn, 1987]. Usefully, ICP is able to work on raw sensor data in the form of point-sets, irrespective of their intrinsic properties, such as distribution, sampling density and noise intensity. However, the algorithm is limited by its assumption that closest-point pairs should correspond, which fails when the point-sets are not coarsely aligned or the moving 'model' point-set is not a proper subset of the static 'scene' point-set. The latter occurs frequently, since differing sensor viewpoints and dynamic objects lead to occlusion and partial-overlap. Moreover, this closest-point assumption means that ICP is susceptible to missing correspondences, which lead to incorrect data association, and local minima, in which the optimisation gets trapped, producing erroneous estimates without a reliable means of detecting failure.

The large quantity of work published on ICP, its variants and other local registration techniques precludes a comprehensive review. For additional background, the reader is directed to surveys on ICP variants [Rusinkiewicz and Levoy, 2001; Pomerleau et al., 2013] and 3D point-set and mesh registration techniques [Castellani and Bartoli, 2012; Tam et al., 2013]. To improve the speed of the ICP algorithm, Nüchter et al. [2007] used a kd-tree and caching for closest-point search, Chen and Medioni [1992] proposed the point-to-plane distance that typically reduces the number of iterations required, and Fitzgibbon [2003] proposed the use of a distance transform for constant-time nearest neighbour look-up. For this, the closest points in one point-set are pre-computed for all grid centres of a discretised volume. While this pre-processing step can be time-

consuming, it is amortised if many point-sets are to be aligned with the point-set that has been processed. To improve the robustness of ICP to outliers from occlusion and partial overlap, outlier rejection [Zhang, 1994; Granger and Pennec, 2002], trimming [Chetverikov et al., 2005], and robust error functions [Fitzgibbon, 2003] have been applied. These approaches perform robust statistical analysis of the residual errors, removing or reducing the influence of those most likely to be outlier correspondences. To enlarge the basin of convergence of ICP, Granger and Pennec [2002] proposed Expectation Maximisation ICP (EM-ICP) that used probabilistic correspondences with Gaussian weights and an annealing scheme on the variance, and Minguez et al. [2005] used a geometric distance measure for finding closest-point correspondences that simultaneously accounted for translational and rotational displacements, having made the observation that a small rotational displacement caused points far from the sensor to be displaced significantly from their correct correspondents. However, this distance measure prevents the use of data structures for expediting the search for nearest neighbours, such as a kd-tree or distance transform. Finally, to improve the speed, accuracy and basin of convergence of ICP, Fitzgibbon [2003] proposed Levenberg-Marquardt ICP (LM-ICP), applying the general-purpose LM optimisation algorithm [Moré, 1978]. This approach models registration as a general optimisation problem and is therefore quite versatile, enabling the use of robust error functions to attenuate the influence of points with large errors and distance transforms to compute the ICP error without establishing explicit point correspondences.

The family of probabilistic alignment approaches also seeks to improve the robustness of ICP to noise, outliers, and poor initialisations. Many of these approaches [Chui and Rangarajan, 2003; Myronenko and Song, 2010; Tsin and Kanade, 2004; Jian and Vemuri, 2011] can be used for both rigid and non-rigid registration, with non-rigid deformations modelled by thin-plate splines [Bookstein, 1989; Chui and Rangarajan, 2003] or Gaussian radial basis functions [Yuille and Grzywacz, 1989; Myronenko and Song, 2010]. Both Chui and Rangarajan [2003] and Myronenko and Song [2010] took a probabilistic approach to correspondence assignment using a Gaussian affinity matrix. Chui and Rangarajan [2003] proposed the Robust Point Matching algorithm that used soft assignment and deterministic annealing to alternately update the correspondences and estimate the transformation. Each point from one point-set is assumed to correspond to a weighted sum of the points from the other point-set using the kernelised pairwise distance affinity matrix. Myronenko and Song [2010] proposed the similar Coherent Point Drift algorithm that interpreted the alternating update strategy using the Expectation Maximisation (EM) framework [Dempster et al., 1977]. Horaud et al. [2011] extended this EM interpretation using the Expectation Conditional Maximisation (ECM) algorithm that shares the desirable convergence properties of EM but is

more suitable for use with anisotropic covariances. However, these algorithms used maximum likelihood estimation, which is sensitive to outliers, and therefore required an additional Gaussian component to model outliers.

A more versatile framework can be constructed by modelling both point-sets as probability distributions and minimising a discrepancy measure between them, obviating the need for establishing explicit point correspondences. Indeed, the ICP algorithm itself has been shown to be related to minimising the Kullback-Leibler divergence of two Gaussian Mixture Models (GMMs) [Jian and Vemuri, 2011]. Tsin and Kanade [2004] developed the Kernel Correlation algorithm that minimised an objective function that was proportional to the correlation of two kernel density estimates, implicitly modelling the point-sets as GMMs. In a similar way, Glaunes et al. [2004] modelled the point-sets as discrete distributions using weighted sums of Dirac measures and then estimated the optimal diffeomorphic transformation between the distributions. A more generic framework was proposed by Jian and Vemuri [2011] with the GMM Registration algorithm. It modelled the point-sets as GMMs with equally-weighted Gaussians centred at every point in the set with identical and isotropic covariances, and minimised the robust L_2 distance between densities. A very similar framework, the Normal Distributions Transform (NDT) method, was developed by Biber and Straßer [2003], Magnusson et al. [2007], and Stoyanov et al. [2012]. The method modelled the point-sets as structured GMMs with full data-driven covariances, by computing Gaussian parameters at each cell of a 3D grid, and one implementation minimised the L_2 distance between densities [Stoyanov et al., 2012]. The algorithm was shown to be faster and more robust to poor initial alignments than ICP [Magnusson et al., 2009]. While these L_2 methods are robust to outliers, they are not robust to some common sampling artefacts, including occlusions and variable sampling densities, due to the generative models used to construct the Gaussian mixtures.

The alignment solution proposed in Chapter 4 of this thesis, the Support Vector Registration (SVR) algorithm [Campbell and Petersson, 2015], belongs to this family of probabilistic approaches, exploiting the outlier robustness of the L_2 distance between probability densities. However, it also corrects a deficiency in existing approaches by considering robustness to sampling artefacts as a critical feature. This robustness is achieved by applying a discriminative model, a Support Vector Machine (SVM) classifier, to efficiently construct the Gaussian mixtures, which regularises the sampled points, creating a smooth, occlusion-resistant surface independent of point density and adaptive to local structural complexity. Robustness to occlusions and variable sampling densities improved the viewpoint-invariance of the models and therefore the alignment accuracy. However, while this and the other probabilistic methods are more robust to outliers and poor initialisations than ICP, they are still susceptible to local optima

and are dependent on a good transformation initialisation. The next section explores those works that are less susceptible to local optima through the application of global optimisation techniques.

Global Optimisation

Global optimisation endeavours to avoid incorrect locally-optimal alignments by expanding the search domain to cover a much greater region in correspondence or transformation space. No guarantees are given by algorithms in this category that the global optimum will be attained, although some approaches do specify the probability that a certain number of iterations will find the optimum. In addition, while these approaches solve for both the transformation and the correspondences, they typically alternate between the two spaces rather than solving them jointly. As such, the approaches can be classified into those where the search is led by correspondence space search or by transformation space search.

Methods for which correspondence search leads can be divided into the hypothesise-and-test and the hypothesise-and-vote frameworks. Many of these approaches can also be applied to subsets of the original point-sets, such as those extracted by feature detectors, to reduce their runtime. For the family of hypothesise-and-test algorithms, also known as sample-and-verify algorithms, exhaustive search [Huttenlocher and Ullman, 1990] can be performed by hypothesising a transformation from all possible pairs or triplets of points, for 2D or 3D respectively, in each dataset. As discussed in Section 2.1.1, these are the minimal number of correspondences required to find the rigid transformation between two sets of positional sensor data. Each hypothesis is tested by transforming one point-set and measuring how well the point-sets align, using a geometric distance or counting the number of inliers. Clearly the complexity of exhaustive search is higher when correspondences are not available, since every point in one point-set could correspond to every point in the other point-set. For the 3D case with point-sets of size M and N, the time complexity is $\mathcal{O}(M^4N^3 \log N)$ for correspondence sampling and transformation testing.

The RANSAC framework [Fischler and Bolles, 1981] can also be applied in the correspondence-free case, adding randomisation to the correspondence sampling step. Using a constant-sized set of random hypotheses for one point-set reduces the time complexity to $\mathcal{O}(MN^3 \log N)$. That is, the runtime required for a high probability of success scales polynomially with the size of the input. Nonetheless, the time complexity is high, limiting the approach to datasets with a small number of points. Irani and Raghavan [1999] further proposed the randomisation of the transformation testing step, testing only a constant number of random points in the transformed set except when

this initial test indicates a good quality match. This reduces the time complexity to $\mathcal{O}(N^3 \log N)$ for 3D data, although Irani and Raghavan [1999] only tested 2D datasets due to the still considerable time complexity. Chen et al. [1999] proposed improvements to the RANSAC framework for 3D alignment, including rigidity constraints and wide 3-point bases to reduce the number of potential correspondences and improve their robustness to noise and outliers.

Another set of approaches use geometric invariances to reduce the time complexity of these hypothesise-and-test strategies [Huttenlocher, 1991; Aiger et al., 2008; Mellado et al., 2014; Raposo and Barreto, 2017]. Huttenlocher [1991] observed that the ratio of distances between three collinear point was preserved by rigid and affine transformations and so, given a set of 4 coplanar points in one point-set and hence two invariant ratios, all sets of approximately congruent 4-points in the other point-set can be extracted efficiently. Aiger et al. [2008] extended this approach into 3D and, by also pre-processing the invariants and storing them in an appropriate data structure, reduced the time complexity of the problem to $\mathcal{O}(N^2 + k)$, where the number of congruent sets k in the second point-set is small in practice. Additional constraints for rigid transformations further reduced the set of candidate congruent 4-points. The approach used wide 4-point bases for noise and outlier resilience and operated on raw sensor data, although feature descriptors could be used to further reduce the runtime. More recently, a linear-time $\mathcal{O}(N)$ extension was proposed by Mellado et al. [2014] that exploited a hash-table-based data structure tailored to the problem to reduce the time complexity. Finally, Raposo and Barreto [2017] showed that the runtime can be reduced further by using 2-point bases if the normal vector of one of the points is also known. Moreover, using a line base instead of a quadrilateral base allowed wider bases when the overlapping area was small, improving the runtime and noise and outlier robustness.

Like hypothesise-and-test methods, hypothesise-and-vote or sample-and-vote algorithms [Ballard, 1981; Stockman, 1987; Olson, 1997; Wolfson and Rigoutsos, 1997] search stochastically over the set of correspondences to generate transformation hypotheses. However, instead of testing the hypotheses as they are generated, each hypothesis generates a vote for its transformation in a discretised Hough space. At the end, high-probability clusters in transformation space are identified, under the assumption that hypotheses with correct correspondences will vote consistently for the correct transformation. Pose clustering methods [Stockman, 1987; Olson, 1997] use an accumulation table for the voting and have a $\mathcal{O}(MN^3)$ time complexity when sampling is randomised sampling. A geometric hashing method was proposed by Wolfson and Rigoutsos [1997] to reduce the time complexity of voting methods to $\mathcal{O}(N^3 \log N)$ by pre-processing configurations of one point-set using a hash table.

Another class of correspondence-free global optimisation methods has transformation search lead. For this, heuristic or stochastic methods can be applied, although they are not guaranteed to converge to the correct alignment. Sandhu et al. [2010] employed a particle filtering strategy to expand the search domain of a local Gaussian mixture correlation optimiser. This principled but stochastic approach defined an uncertainty model for the transformation parameters to robustly predict the new distribution from which to sample particles. A related approach was proposed by Wachowiak et al. [2004] using particle swarm optimisation for the registration of 3D biomedical image data. Robertson and Fisher [2002] and Silva et al. [2005] applied genetic algorithms to the alignment problem, representing transformations as chromosomes with six genes corresponding to the transformation parameters. A population of individuals (transformation hypotheses) with these chromosomes underwent an evolutionary procedure, with fitter individuals having a greater chance of reproducing to form new transformation hypotheses. Finally, Blais and Levine [1995] and Papazov and Burschka [2011] used simulated annealing with robust loss functions to widen the basin of convergence significantly, reducing the likelihood that the search will become trapped in a local optimum near the point of initialisation. However, these methods may not find the correct alignment without a good transformation prior distribution or initialisation being provided, due to the stochastic nature of the search.

As with the local optimisation algorithms, some global optimisation methods rely on the statistical properties of the point-sets. Principal Component Analysis (PCA) has frequently been applied to coarsely align positional sensor data without correspondences or transformation prior. Dorai et al. [1997] and Chung et al. [1998] registered 3D data by aligning the centroids of the data and then aligning the principal axes found using PCA. This approach led to 180° rotation errors due to the sign ambiguity of the principal axes and failed for symmetric objects and incompletely overlapping data. An extension was proposed by Xiong et al. [2013b] to align eigenvectors in feature space using Kernel PCA, but still required substantial overlap and minimal occlusion. Frequency domain solutions, surveyed in Sun et al. [2014], provide an alternative approach. Makadia et al. [2006] decoupled rotation and translation search using the observation that the surface normal statistics are independent of translation. They obtained the rotation by maximising the convolution of the Extended Gaussian Images (EGI) [Horn, 1984] of the two surface normal sets, using the spherical Fourier Transform, and then estimated the translation using the fast Fourier Transform. However, discretisation artefacts were introduced by the use of histogram density estimates and the reliance on EGI peaks made the method susceptible to noise. Another 3D spectral registration method was proposed by Bülow and Birk [2013] for partially-overlapping data, using Phase Only Matched Filtering (POMF) to estimate the transformation parameters.

However, the use of interpolation for rotation estimation limited the approach to small angular deviations ($\pm 30°$).

Machine learning techniques have only had limited application to the problem of positional sensor data alignment. An indirect approach for aligning RGB-D images was proposed by Shotton et al. [2013], which used scene coordinate regression forests to infer the camera pose and thereby align the data. However, the method required a training set of RGB-D images and poses to localise new camera views and therefore would not be able to align, even indirectly, two arbitrary depth images. End-to-end deep learning architectures have not, as of yet, been applied to the 3D–3D registration problem, due to the unstructured, permutation-invariant and size-varying nature of 3D point-set data. However, neural networks have been applied to 3D feature detection and matching [Ai et al., 2017], feature description to encode local geometry using a dimensionality-reducing auto-encoder Elbaz et al. [2017], and a differentiable reformulation of the RANSAC algorithm [Brachmann et al., 2017], which shows promise for the correspondence-free alignment problem if the data problem can be solved.

In contrast to the work in this section, the solution to positional sensor data alignment proposed by this thesis in Chapter 5 provides a guarantee of optimality. Since typical alignment problems have a very large search space in the correspondences or transformations and a high level of non-convexity, it can be very difficult for methods that do not guarantee optimality to find the global optimum or even a sufficiently good local optimum. As a consequence, global-optimality is a very desirable and often necessary attribute for reliable alignment algorithms. The next section examines this class of globally-optimal algorithm and situates the work of this thesis in its research context.

Global Optimality

There is a relatively small body of literature that is concerned with providing optimality guarantees for the problem of aligning positional sensor data without correspondences. Globally-optimal methods find a transformation that is guaranteed to be an optimiser of a suitable objective function without requiring a transformation prior. The Branch-and-Bound (BB) paradigm, proposed by Land and Doig [1960], can be applied to provide such optimality guarantees for non-convex alignment problems. However, tractability has been the biggest challenge thus far for BB-based geometric alignment algorithms, particularly when scaling to 3D problems [Breuel, 2003].

Historically, only a small number of studies into what was known as the geometric matching problem provided optimality guarantees of any kind [Breuel, 1992; Mount et al., 1999; Breuel, 2003]. Breuel [1992] pioneered the use of BB to optimally solve

geometric alignment problems, proposing in Breuel [2003] a family of bounding functions for different transformations and objective functions. However, like the work of [Mount et al., 1999] and Pfeuffer et al. [2012] for satellite and biomedical imagery, the transformations investigated were predominantly from $\mathbb{R}^2 \to \mathbb{R}^2$, that is, 2D–2D alignment. A naïve extension to the six parameters required for 3D rigid transformations is challenging, due to the vastly increased volume of the search space and the nonlinearity of the rotation operator with respect to the rotation parameters. Breuel observed that 3D transformations were "often impractical because the complexity is too high" [Breuel, 2003, p. 24] — tractability being identified as the primary impediment.

More recently, branch-and-bound has also been applied to a variety of 3D geometric alignment problems to provide guaranteed optimal solutions. Existing 3D methods [Olsson et al., 2006, 2009; Bazin et al., 2013; Hartley and Kahl, 2009; Li and Hartley, 2007; Parra Bustos et al., 2014] are often very slow or make restrictive assumptions about the point-sets, correspondences or transformations. For example, Olsson et al. [2006] and Olsson et al. [2009] presented optimal algorithms for the 2D–3D (PnP) and 3D–3D registration problems respectively with known correspondences. In Olsson et al. [2009], BB and the bilinear relaxation of rotation quaternions was used to find optimal solutions to point-to-point, point-to-line, and point-to-plane registration. Bazin et al. [2013] used BB for aligning directional sensor data to find optimal correspondences between images using both geometry and appearance. Similarly, Hartley and Kahl [2009] used BB for optimal relative pose estimation by bounding the group of 3D rotations.

For 3D–3D registration problems without correspondences, Li and Hartley [2007] minimised a Lipschitzised L_2 error function using branch-and-bound with an octree data structure to implement the search. However, they assumed that the transformation was pure rotation and that the point-sets were the same size and had a one-to-one correspondence, and hence no random or structured outliers from partial overlap and occlusion. Moreover, the reported runtimes were quite high for the size of the problems solved. Parra Bustos et al. [2014] similarly assumed a pure rotation transformation but did not restrict the point-sets for their optimal 3D–3D registration algorithm. They achieved efficient runtimes by exploiting stereographic projection techniques, circular R-trees and matchlists for optimal inlier set cardinality maximisation, a robust objective function. Moreover, they also improved the rotation bound from Hartley and Kahl [2009] that restricted a point perturbed by a set of angle-axis rotations to lie within a certain ball, with the observation that the point must also lie on a sphere, and hence on a spherical patch.

Within the last few years, there have been a number of new methods proposed for full 6-DoF 3D–3D rigid alignment without correspondences or restrictions on the

point-sets [Yang et al., 2013b, 2016; Parra Bustos et al., 2016; Campbell and Petersson, 2016; Straub et al., 2017]. Yang et al. [2016] proposed the Go-ICP algorithm, which found the optimal solution to the closest-point L_2 error between point-sets, the error measure used in the ICP algorithm. Further, they accelerated the algorithm by encapsulating ICP as a local optimisation subroutine inside a nested BB scheme over the translation and rotation domains. However, Go-ICP was sensitive to outliers from occlusion and partial overlap, due to its non-robust objective function. The proposed trimming strategy went some way to alleviating this, but increased the runtime, required an estimate of the overlap ratio and led to potential ambiguities in the solutions. Moreover, the implementation used a distance transform to make the problem tractable. This approximation meant that ϵ-suboptimality could not be guaranteed unless the resolution of the distance transform was sufficiently high. Parra Bustos et al. [2016], extending their rotation search approach in Parra Bustos et al. [2014], embedded their rotation search kernel into the nested Go-ICP framework for full 6-DoF registration with a robust inlier set cardinality maximisation objective function. They improved upon the time efficiency of Go-ICP with a tighter bounding function and sophisticated projections and data structures that can not be easily exploited under the least squares objective of Go-ICP.

Concurrently, the probabilistic mixture model approach developed in Chapter 5 of this thesis was proposed in Campbell and Petersson [2016] for robust and optimal 3D–3D registration using branch-and-bound. Unlike Yang et al. [2016], the work that immediately preceded it, the Globally-Optimal Gaussian Mixture Alignment (GOGMA) algorithm optimised an inherently robust objective function, the L_2 distance between probability densities. Tight bounds on the objective function were derived using the geometry of $SE(3)$, with the rotation bound being tighter than, and directly transferable to, the rotation bound in the Go-ICP algorithm. Both GOGMA and the concurrent method of Parra Bustos et al. [2016] demonstrated that there was a need for inherently robust objective functions to handle outliers in the data.

Subsequent to this work, another mixture model approach was proposed by Straub et al. [2017]. They decoupled rotation and translation search by first rotationally aligning the translation-invariant surface normal distributions, and then aligning the Gaussian mixtures to estimate the translation given rotation. Tight bounds on the robust L_2 distance objective function were derived for a rectangular tessellation of translation space $\mathbb{R}^3$ and a near-uniform tetrahedral tessellation of rotation space $SO(3)$, which is more efficient to optimise over than angle-axis tessellations. Decoupling rotation and translation search improved the optimisation efficiency significantly, since the complexity scales exponentially in the search space dimension, and made it possible to use full covariance Gaussian mixtures for the translation search. However, decoupling

meant that the solutions for rotation and translation were not jointly optimal, creating alignment failure modes as shown in their results. Moreover, the method required surface normals, which limits the general applicability of the algorithm to smoother, densely-sampled surfaces.

Collectively, the literature on aligning positional sensor data highlights the need for tractable, robust and optimal solutions that solve for both the transformation and the correspondences. Such solutions must consider robustness to outliers as a key tenet, since outliers are pervasive and inherent to the problem. Also inherent to all useful alignment objective functions is non-convexity. Therefore, global search, preferably with optimality guarantees, is also a key tenet, to avoid converging on highly prevalent local optima. The next section changes the focus to a different class of alignment problem, that of aligning directional and positional sensor data/

2.2 Aligning Directional and Positional Sensor Data

The task of aligning directional and positional sensor data combines data that contains the spatial position of the samples and data that contains only the direction of the samples with respect to the sensor. Consequently, the task has some unique challenges, as reflected in the depth and breadth of the existing literature. While the general problem is not limited to 2D and 3D modalities, 2D–3D alignment problems for directional and positional data predominate in computer vision. These can be grouped into the linked 2D–3D geometric matching and absolute camera pose problems.

In this section, the literature will be divided into whether or not the alignment problem presupposes a set of correspondences between elements of the datasets. Methods for aligning sensor data given a potentially noisy correspondence set will be surveyed first, followed by methods for addressing the more challenging and general problem of simultaneously solving for the transformation and the correspondences.

2.2.1 Alignment With Correspondences

A large body of work exists for solving the problem of aligning directional and positional data when correspondences are available. However, before these approaches can be applied, a set of putative correspondences between the sensor data must be found.

Generating Correspondences

For the 2D–3D alignment of an image and a point-set, it can often be difficult to establish putative correspondences between 2D image features and 3D features. This arises from the significantly different modalities of the 2D and 3D data since features

that are salient in the appearance space of an image may not be salient in the structural space of a point-set.

Keypoint or feature detection and feature description techniques provide a relatively robust and repeatable way to detect interest points such as edges or corners within each modality and compute likely correspondences between them. For images, these techniques include Canny edges [Canny, 1986], Harris corners [Harris and Stephens, 1988], Scale-Invariant Feature Transform (SIFT) features [Lowe, 2004] and Maximally Stable Extremum Regions (MSER) blob features [Matas et al., 2004]. For 3D sensor data, keypoint detectors and feature descriptors include Intrinsic Shape Signatures (ISS) [Zhong, 2009], Normal Aligned Radial Features (NARF) [Steder et al., 2011], Spin Images [Johnson and Hebert, 1999], 3D Shape Context [Frome et al., 2004], Point Feature Histograms (PFH) [Rusu et al., 2008b], Fast PFH [Rusu et al., 2009], and Signatures of Histograms of OrienTations (SHOT) [Tombari et al., 2010]. See Tombari et al. [2013] for an evaluation of the 3D keypoint detectors, including those derived from 2D image detectors such as the Harris operator [Harris and Stephens, 1988; Sipiran and Bustos, 2010] and the Scale-Invariant Feature Transform (SIFT) [Lowe, 2004; Maes et al., 2010], with respect to distinctiveness and repeatability.

Nonetheless, finding correspondences across the two modalities is much more challenging. In some cases the point-set has visual information associated with it, such as colour, reflectance or SIFT features, making the cross-modality correspondence problem simpler [Sattler et al., 2017]. For these problems, the correspondence problem becomes unimodal and a correspondence set can be generated using some similarity measure between visual features. More recent approaches use machine learning techniques to learn cross-modality correspondences from the data [Shotton et al., 2013; Kendall et al., 2015; Brachmann et al., 2017]. However, the correspondence problem remains inherently non-trivial. What is more, some image pixels, such as sky pixels, will never correspond to 3D features since they are not geometrically meaningful.

From Correspondences to Alignment

Once a correspondence set has been generated, there are many algorithms that can solve for the transformation between the sets of sensor data. For 2D–3D alignment, Perspective-n-Point (PnP) solvers [Grunert, 1841; Haralick et al., 1994; Gao et al., 2003; Olsson et al., 2006; Lepetit et al., 2009; Kneip et al., 2011; Hesch and Roumeliotis, 2011; Penate-Sanchez et al., 2013; Kneip et al., 2014] are able to estimate the pose of a calibrated camera given a set of noisy image points and their corresponding 3D points. The minimal case (P3P), reviewed in Haralick et al. [1994], requires $n = 3$ 2D–3D correspondences, for which there may be up to four solutions that can

be disambiguated using a fourth point. An efficient closed-form solution was derived for the P3P problem in Kneip et al. [2011]. Linear solutions were proposed for $n = 4$ and $n = 5$ 2D–3D correspondences in Quan and Lan [1999], and the Direct Linear Transform (DLT) can be used for $n \geqslant 6$ correspondences [Sutherland, 1963; Hartley and Zisserman, 2003] when the 3D points are in a general configuration. However, these linear approaches optimised an algebraic error and are susceptible to local optima. For an arbitrary number of correspondences $n \geqslant 3$, the general PnP problem can be solved, with greater robustness to noise evident for larger values of n [Lepetit et al., 2009; Hesch and Roumeliotis, 2011; Penate-Sanchez et al., 2013; Kneip et al., 2014]. In particular, EPnP [Lepetit et al., 2009] is suitable for use with deformable objects and UPnP [Penate-Sanchez et al., 2013] relaxes the calibration requirements by also solving for focal length, as does the DLT method. Each of these methods can be followed by non-linear optimisation to refine the camera pose, typically applying the Levenberg–Marquardt algorithm [Levenberg, 1944]. Finally, to address the problem of local optima, prevalent when the number of correspondences is small or the level of noise is high, Olsson et al. [2006] proposed the first globally-optimal branch-and-bound algorithm for the PnP problem. It used a geometric error norm and guaranteed that the global minimum of the L_2 norm of the reprojection errors would be attained. A more efficient optimal algorithm was proposed by Hartley and Kahl [2009] using the L_∞ norm and solving a series of second-order cone programs. However, neither of these optimal strategies were robust to outlier correspondences and therefore they may converge to an incorrect pose, as with most PnP algorithms.

When outliers are present in the correspondence set, the RANdom SAmple Consensus (RANSAC) framework [Fischler and Bolles, 1981; Chum and Matas, 2008] or robust global optimisation [Enqvist and Kahl, 2008; Li, 2009; Enqvist et al., 2012; Ask et al., 2013; Svärm et al., 2014; Enqvist et al., 2015; Svärm et al., 2016] can be used to find the inlier set. RANSAC can be applied by sampling three 2D–3D point pairs, computing the pose hypothesis using P3P, transforming the point-set and counting the number of inliers [Kneip and Furgale, 2014], but does not provide any guarantee of optimality. In contrast, Enqvist and Kahl [2008] proposed a globally-optimal algorithm that extended the L_∞ norm approach of Hartley and Kahl [2009] to handle outliers and reported runtimes an order of magnitude faster than the previous approaches. The key insight was that their pairwise 'pumpkin' constraints were independent of camera rotation, so branch-and-bound search over $\mathbb{R}^3$ was sufficient to determine the camera translation. However, their formulation was unable to guarantee the convergence of the bounds on the optimal solution. Nonetheless, this algorithm, and the mixed integer formulation of Li [2009], were both capable of discarding outliers, improving the quality of the fitted solution. More recent works address optimality for model estimation

problems such as image stitching, triangulation, relative pose and rigid 2D registration, generating polynomial-time algorithms that optimise a robust loss function such as the number of inlier or the truncated L_2 norm [Enqvist et al., 2012; Ask et al., 2013; Svärm et al., 2014; Enqvist et al., 2015]. Finally, Svärm et al. [2016] extended this body of work to the problem of absolute camera pose estimation for a large-scale model, maximising the number of inliers in polynomial time. However, in order to achieve a polynomial-time optimal algorithm, the vertical direction and height (within an interval) of the camera is assumed to be known.

An alternative approach is afforded by outlier removal schemes [Sim and Hartley, 2006; Li, 2007; Ke and Kanade, 2007; Olsson et al., 2008, 2010; Yu et al., 2011; Parra Bustos and Chin, 2015; Chin et al., 2016] that can make the problem more tractable and are often used in conjunction with the global optimisation methods surveyed above. Sim and Hartley [2006] proposed a method for detecting outliers in quasiconvex problems using the L_∞ framework, however the absolute calibrated camera pose problem is not quasiconvex and the method removes all measurements in the support set, including any inliers. Li [2007], Ke and Kanade [2007] and Olsson et al. [2008] also presented methods for removing outliers in quasiconvex problems, such as triangulation and camera resectioning, but all have high computational complexity, restricting them to situations where outliers are rare. More recently, Parra Bustos and Chin [2015] and Chin et al. [2016] derived methods that guaranteed that only true outliers were removed, albeit only for quasiconvex problems.

Large-scale camera localisation, with its significant demands on outlier robustness and computational efficiency, has received a lot of attention recently [Li et al., 2010; Sattler et al., 2011, 2012; Li et al., 2012; Zeisl et al., 2015; Enqvist et al., 2015; Svärm et al., 2016; Sattler et al., 2017]. These methods develop sophisticated matching strategies to avoid outlier correspondences at the outset and may also incorporate RANSAC, global optimisation and outlier removal stages in their sparse feature pipeline. A recent state-of-the-art approach is Active Search [Sattler et al., 2017], which prioritises those SIFT features that are more likely to yield inlier 2D–3D correspondences, and achieves very high camera pose accuracy in feature-rich outdoor environments. Like these methods, state-of-the-art Simultaneous Localisation and Mapping (SLAM) systems [Klein and Murray, 2007; Newcombe et al., 2011; Engel et al., 2014; Mur-Artal et al., 2015] also solve the absolute camera pose estimation problem, however they work most effectively in controlled environments because they are unable to handle large changes in viewpoint or appearance. However, these methods share the assumption that there is a reasonable expectation that 2D–3D correspondences can be found. For this reason, they are often only practical for 3D models that have been constructed using stereopsis or Structure-from-Motion (SfM). These models associate an image feature with each

3D point, facilitating inter-modal feature matching. Generic point-sets do not have this property; a point may lie anywhere on the underlying surfaces in a laser scan, not just where strong image gradients occur. Moreover, these large databases of features have their own disadvantages, being computationally expensive to generate, not scaling well and lacking robustness to appearance changes due to environmental conditions.

It should be observed that some of these approaches [Fischler and Bolles, 1981; Enqvist and Kahl, 2008] can be extended to the correspondence-free case by providing the algorithm with all possible permutations of the correspondence set. However, this leads to a hard combinatorial problem that quickly becomes infeasible.

In contrast to the work in this section, the solutions to aligning directional and positional sensor data proposed by this thesis in Chapter 6 do not require correspondences. As a consequence, they are not reliant on the robust detection and description of cross-modal features, which is an unsolved problem for 2D and 3D data, particularly when only geometric information is known. The next section investigates the family of methods for aligning directional and positional sensor data without correspondences, a family that includes the solutions developed in this thesis.

2.2.2 Alignment Without Correspondences

When a set of correspondences is not available and is difficult to obtain, the problem becomes much more challenging. Methods that address this problem must jointly solve for the transformation and the correspondences between the sets of directional and positional sensor data. However, most algorithms do not explicitly search over correspondence and transformation space simultaneously, instead assuming that the correspondences determine the transformation or vice versa. Moreover, the correspondences themselves need not be explicit, with soft or probabilistic assignment being frequently applied.

In addition to being more challenging, the correspondence-free class of problem is more general than the class that assumes correspondences are available and can be applied in more situations. Furthermore, solutions to the problem of alignment given correspondences are regularly used as sub-algorithms in correspondence-free approaches since they can be more time efficient.

The alignment of 2D directional sensor data without correspondences has previously been addressed for problems such as correspondence-free Structure-from-Motion (SfM) [Dellaert et al., 2000; Makadia et al., 2007; Lin et al., 2012] and relative camera pose [Hartley and Kahl, 2009; Bazin et al., 2013; Fredriksson et al., 2016]. While SfM and relative camera pose problems have received a lot of attention historically, there has been a parallel investigation into 2D–3D geometric matching and absolute camera

pose problems. These are alignment problems involving both directional and positional sensor data, not directional data alone.

The alignment of 2D directional and 3D positional sensor data without correspondences has previously been addressed for 2D–3D geometric matching and correspondence-free absolute camera pose problems [Huttenlocher and Ullman, 1990; Cass, 1997; Jacobs, 1997; Olson, 2001; Jurie, 1999; Breuel, 2003; David et al., 2002; Moreno-Noguer et al., 2008; Brown et al., 2015]. While these approaches do not assume that a correspondence set is available, many of them use simplified camera models such as linear affine approximations [Huttenlocher and Ullman, 1990; Cass, 1997; Jacobs, 1997; Jurie, 1999; Breuel, 2003], which are only reasonable when the distances from the camera to the 3D points are large in comparison to the relative depths of those points.

Some approaches sidestep the full 2D–3D problem by utilising a collection of images [Paudel et al., 2015b] or multiple cameras [Paudel et al., 2015a] to first obtain 3D positional information from the 2D data, which is then registered against a 3D point-set. Paudel et al. [2015b] align a scanned 3D point-set with a sparse SfM point-set of unknown scale, generated from a collection of images. The method is restricted to scenes with predominantly planar surfaces, a Manhattan world assumption, for which an optimal assignment of SfM points to extracted planar surfaces is performed using the branch-and-bound paradigm. However, a general solution must be able to handle the case of aligning 2D points from a single image or camera with a 3D point-set.

The methods for aligning directional and positional sensor data without correspondences can be usefully classified based on the type of optimisation. The following sections will examine approaches that employ local, global and globally-optimal search techniques.

Local Optimisation

For the most general problem of aligning directional and positional sensor data without correspondences, there are several approaches that employ local optimisation [Beveridge and Riseman, 1995; Wunsch and Hirzinger, 1996; David et al., 2002; Moreno-Noguer et al., 2008]. These methods share a common iterative approach that alternates between searching over the transformation and correspondence spaces and use a full perspective model. They also all require a transformation prior and may only find a locally-optimal solution within the convergence basin of the prior. To alleviate this, these methods are frequently used within a global optimisation framework, with the methods of Beveridge and Riseman [1995] and Moreno-Noguer et al. [2008] incorporating global search natively.

Wunsch and Hirzinger [1996] optimised a 2D–3D alignment objective function that combined constraints on the camera pose with constraints on the correspondences. The algorithm proceeded by finding the point on a line of sight of a 2D feature that was closest to each 3D feature, analogously to the iterative closest point algorithm. Then a 3D–3D pose problem was solved to find the pose that best aligned these points with the 3D features. These two steps were iterated until some convergence criteria were satisfied.

This approach is not dissimilar to the SoftPOSIT algorithm of David et al. [2002], which also solves the camera pose estimation problem from a single image using local optimisation. However, they represented the correspondence constraints analytically, solving the full 2D–3D problem at each iteration. SoftPOSIT alternates correspondence assignment using SoftAssign [Gold and Rangarajan, 1996] with an iterative pose update algorithm POSIT [Dementhon and Davis, 1995], applying deterministic annealing to encourage a large basin of convergence. A least squares objective function is used in the pose update step and is therefore not robust to outliers in the data. The time complexity of the algorithm is $\mathcal{O}(MN^2)$ for M 3D points and N image points. However, both of these methods use non-robust objective functions and are susceptible to local optima, require a pose prior and cannot guarantee global optimality.

In contrast to the work in this section, the solutions to aligning directional and positional sensor data proposed by this thesis in Chapter 6 are not susceptible to local optima and do not require a pose prior. Since a good estimate of the pose is not known in advance for many alignment problems other than tracking, local optimisation algorithms are often unsuitable. Hence, global optimisation techniques that search beyond the local region are often required. The next section examines the application of these techniques to the geometric sensor data alignment problem.

Global Optimisation

Global optimisation endeavours to avoid incorrect locally-optimal alignments by expanding the search domain to cover a much greater region in correspondence or transformation space. No guarantees are given by algorithms in this category that the global optimum will be attained, although some approaches do specify the probability that a certain number of iterations will find the optimum.

Grimson [1990] applied a hypothesise-and-test approach to the simultaneous pose and correspondence problem, removing the need for a pose prior. The approach hypothesised a small set of 2D–3D correspondences from which the transformation was computed. The 3D points were then back-projected into the image using this transformation and the quality of the pose hypothesis was measured. This approach searched

the entire domain of feasible transformations, but did not guarantee that the optimal pose would be found and quickly became intractable as the number of points increased.

The hypothesise-and-test principle is shared by the more general RANdom SAmple Consensus (RANSAC) algorithm proposed in Fischler and Bolles [1981]. Unlike Grimson [1990], the RANSAC algorithm can be applied even when no information is available to constrain the correspondences in the hypothesis generation step. To do so, sets of three 2D–3D correspondences can be drawn randomly from the set of all possible correspondences in order to determine the hypothesis transformation. The time complexity of RANSAC applied to the 2D–3D correspondence-free alignment problem is $\mathcal{O}(MN^3 \log N)$ for M 3D points and N image points [David et al., 2004]. That is, the runtime required for a high probability of success scales polynomially with the size of the input. Heuristic criteria to terminate the search early have been introduced to address this prohibitive time complexity [Ayache and Faugeras, 1986; Grimson, 1991], but the approach remains limited to small numbers of points.

Like hypothesise-and-test methods, pose clustering or generalised Hough transform approaches [Stockman, 1987; Breuel, 1992; Cass, 1997; Olson, 1997] search stochastically over the full set of correspondences to generate pose hypotheses. However, instead of testing the pose hypotheses as they are generated, these methods generate all hypotheses before identifying high-probability clusters in 6D pose space, under the assumption that these will contain only hypotheses with correct correspondences. Due to the highly combinatorial nature of searching the set of 2D–3D correspondences, these methods are limited to small input sizes. This can be seen by the $\mathcal{O}(MN^3)$ time complexity of the method of Olson [1997], which is one of the most efficient algorithms of this type.

Global search can also be achieved by applying heuristic and stochastic optimisation techniques. Beveridge and Riseman [1995] use random-start local search to find the optimal correspondences with a certain confidence, initialising a hybrid pose estimation algorithm at randomly sampled points in the transformation domain. The algorithm searches in the direction of the greatest gradient in the space of 2D–3D line segment correspondences. It uses a weak-perspective camera model to rank neighbouring points in this space and a full-perspective model to update the pose given the correspondences. The time complexity of the local search is $\mathcal{O}(M^2N^2)$ for M 3D points and N image points. In a similar way, a stochastic global search variant of SoftPOSIT was proposed by David et al. [2004], named Random Start SoftPOSIT. It extends the search domain of the original local optimisation algorithm by initialising multiple runs at different randomly sampled points in the transformation domain and terminating when a threshold of inliers is exceeded. A more sophisticated approach was proposed by Moreno-Noguer et al. [2008] with the BlindPnP algorithm, which represents the region of expected

transformations (the pose prior) as a Gaussian mixture model from which a Kalman filter is initialised. This guides a local correspondence hypothesising step incorporating the PnP algorithm, which explores the space of correspondences within the subset of matches that are plausible from that pose guess. BlindPnP has been shown to outperform SoftPOSIT when large amounts of clutter, occlusions and repetitive patterns are present but is otherwise comparable in accuracy and time complexity. However, these stochastic techniques are still susceptible to local optima and require a pose prior that contains the true transformation. Moreover they cannot guarantee that the optimal solution was found.

Research into learning-based approaches for 2D–3D alignment has a long history [Lamdan and Wolfson, 1988; Burns et al., 1993; Beis and Lowe, 1999]. These indexing methods require a training set of images and camera poses to learn 2D–3D correspondences and thereby estimate pose. Specifically, they learn groupings of hand-crafted 2D features for each 3D object and store the associated vectors in hash tables [Lamdan and Wolfson, 1988; Burns et al., 1993] or kd-trees [Beis and Lowe, 1999]. At runtime, feature groupings are extracted from the test image and are used to index into the data structure, extracting the learned 2D–3D correspondence hypotheses. These hypotheses are used for pose estimation and verification. The geometric hashing approach [Lamdan and Wolfson, 1988; Burns et al., 1993] is limited to planar scenes due to the requirements that the hashing metric is viewpoint invariant and the image features are viewpoint invariant for a general 3D point-set. The approach of Beis and Lowe [1999] does not have this restriction. However, these techniques are not able to handle large pose variations.

In the years since these early learning approaches, there has been some work on applying more sophisticated machine learning techniques to the problem of RGB-D camera pose estimation [Shotton et al., 2013; Glocker et al., 2013]. Shotton et al. [2013] proposed scene coordinate regression forests to infer the pose of an RGB-D camera. The algorithm used a training set of depth images to generate scene coordinate labels that map pixels from the camera coordinate frame to the global coordinate frame. A regression forest was trained with these labels to regress over the labels and thus localise the camera accurately. However, these approaches require RGB-D cameras, limiting their applicability in outdoor environments.

More recently, there have been a number of works that introduce deep learning to the problem of camera pose estimation and 2D–3D alignment [Kendall et al., 2015; Brachmann et al., 2017; Kendall and Cipolla, 2017]. These approaches require a large training set of images and camera poses to learn 2D–3D correspondences and thereby regress pose. Unlike the hand-crafted 2D features of the indexing methods, these approaches learn the correspondences directly from the data. Kendall et al. [2015]

introduced a Convolutional Neural Network (CNN) for regressing the six degree of freedom pose of a camera from a single RGB image. It localised the camera using high-level features and was robust to lighting conditions, motion blur and other situations where point-based SIFT registration methods fail. However, it was unable to achieve the metric accuracy of geometry-based methods, largely because it used a naïve loss function that did not consider geometry. To remedy this, Kendall and Cipolla [2017] introduced a geometric loss function, learning an optimal weighting between position and orientation, and thereby achieving improved localisation accuracy. Concurrently, Brachmann et al. [2017] showed how to reformulate the RANSAC algorithm such that it is differentiable and hence end-to-end trainable in a deep learning pipeline. This formulation was applied to the problem of camera localisation, providing robust estimation of the camera poses. However, these CNN-based approaches, in addition to requiring a large training set of images and poses, do not estimate the pose with respect to an explicit 3D model. 3D point-set data is notoriously difficult to handle for deep learning architectures, due to its lack of regular structure, permutation invariance and size variation.

Unlike these global optimisation algorithms, the solutions to aligning directional and positional sensor data proposed by this thesis in Chapter 6 provide a guarantee of optimality and do not require training data. Since typical alignment problems have a very large search space in the correspondences or transformations and a high level of non-convexity, it can be very difficult for methods that do not guarantee optimality to find the global optimum or even a sufficiently good local optimum. Consequently, global-optimality is a very desirable and often necessary attribute for reliable alignment algorithms. The next section examines this class of globally-optimal algorithm and situates the work of this thesis in its research context.

Global Optimality

There is a relatively small body of literature that is concerned with providing optimality guarantees for the problem of aligning positional and directional sensor data without correspondences. Globally-optimal methods find a transformation that is guaranteed to be an optimiser of a suitable objective function without requiring a transformation prior. The Branch-and-Bound (BB) paradigm, proposed by Land and Doig [1960], can be applied to provide such optimality guarantees. However, tractability has been the biggest challenge thus far for BB-based geometric alignment algorithms, particularly when scaling to 3D problems.

Historically, only a small number of studies into what was known as the geometric matching problem provided optimality guarantees of any kind [Jurie, 1999; Breuel,

2003]. Jurie [1999] used a probabilistic approach with similarities to BB for 2D–3D alignment under a Gaussian error model. However, the method does not provide a strong optimality guarantee and used a simplified camera model that linearly approximated perspective projection. Breuel [2003] used BB to optimally solve geometric alignment problems, proposing a family of bounding functions for different transformations and objective functions. However, the transformations investigated were predominantly from $\mathbb{R}^2 \rightarrow \mathbb{R}^2$, that is, 2D–2D alignment. While the author also proposed a weak bound for 2D–3D geometric alignment under the simplified orthogonal projection model, they observe that 2D–3D point alignment "is often impractical because the complexity is too high" [Breuel, 2003, p. 24] – tractability being identified as the primary impediment.

More recently, branch-and-bound has also been applied to a variety of other geometric alignment settings to provide guaranteed optimal solutions. For example, Bazin et al. [2013] used BB for aligning directional sensor data to find optimal correspondences between images, Hartley and Kahl [2009] for optimal relative pose estimation by bounding the group of 3D rotations, Li and Hartley [2007] for rotation-only 3D–3D registration, Olsson et al. [2006, 2009] for 2D–3D or 3D–3D registration with known correspondences, Yang et al. [2016] for full 3D–3D registration and Campbell and Petersson [2016] for robust 3D–3D registration.

However, only Brown et al. [2015] and Campbell et al. [2017] have explored the problem of optimally aligning directional and positional sensor data (2D–3D alignment) without correspondences. Brown et al. [2015] proposed a global and ϵ-suboptimal method using the branch-and-bound framework. Their approach found a camera pose whose geometric error, the sum of angular distances between the bearings and their rotationally-closest 3D points, was within ϵ of the global minimum. This objective function is not inherently robust to outliers and therefore a trimming strategy was applied, as used in Yang et al. [2013b]. Such a strategy has the disadvantage of requiring that the inlier fraction be specified in order to trim outliers. This can rarely be known in advance and, if it is over- or under-estimated, the global optimum of the function may not occur at the correct pose. In contrast, an ideal approach would optimise the untrimmed geometric error after having identified and removed the outliers. In addition, the use of ϵ annealing in their framework invalidates the guarantee of ϵ-suboptimality, since branches containing the correct pose may be pruned early.

In contrast, the alignment solutions presented in Chapter 6 of this thesis optimise inherently robust objective functions. The Globally Optimal Pose And Correspondences (GOPAC) algorithm for camera pose estimation [Campbell et al., 2017] is guaranteed to find the exact optimum of the robust inlier set cardinality objective function. Tight bounds on the cardinality of the inlier set for branches of rotation and translation

space are derived in Chapter 6 and an algorithm that integrates local optimisation to accelerate convergence is developed. Moreover, many of the results derived in this thesis can be directly transferred to other geometric alignment algorithms that used BB [Brown et al., 2015; Yang et al., 2016; Campbell and Petersson, 2016] to improve the quality of their bounds and the runtime of their implementations.

Collectively, the literature highlights the need for tractable, robust and optimal solutions to the 2D–3D geometric alignment problem, simultaneously solving for the transformation and the correspondences. Such solutions must consider robustness to (structured) outliers as a key tenet, since outliers are pervasive in this domain. Another key tenet is global search, preferably with optimality guarantees, to avoid converging on local optima that are highly prevalent in alignment objective functions.

The following chapter will present the technical background for the geometric sensor data alignment problem. This background material includes basic elements such as the parametrisations of rigid motions, distance measures and sensor data representations, in addition to objective functions and optimisation techniques for geometric alignment problems.

Chapter 3

Geometric Sensor Data Alignment

This chapter presents the background material for the geometric sensor data alignment problem. The technical background consists of elements that are common to many or all of the approaches proposed in later chapters, forming a mathematical toolkit that will be referred to repeatedly. For completion, this chapter will also cover related concepts that are not used in the work, but are nonetheless important to discuss. The areas of mathematics that are covered span geometry, statistics and optimisation. These areas are linked by their utility to the problem of sensor data alignment.

The chapter starts with a discussion of ways to parametrise the space of rigid motions in two and three dimensions, that is, the space of translations and rotations. The limitations of each approach with respect to computation and memory efficiency is also discussed. Next, the different distance measures in each space are introduced formally, followed by an overview of common sensor data representations. The central part of the chapter introduces objective functions for alignment when correspondences are not available. The discussion tracks a progression from non-robust objective functions to those that are more robust to noise and outliers. Next, local optimisation methods are briefly outlined, followed by methods for optimising over the global domain. The chapter ends with a detailed discussion of the branch-and-bound algorithm, a formalism for optimal global search.

While this chapter will provide the tools for understanding the research field of geometric sensor data alignment and will precisely define the problem in each instance, it is worthwhile to state the problem in general terms at the outset. The problem of geometrically aligning sensor data is the problem of finding the rigid transformation that correctly aligns two sets of sensor data without any prior knowledge about how the data corresponds. Where the concept of corresponding data elements is meaningful, the problem can profitably be thought of as jointly solving for the transformation and correspondence set. The data may be of different dimensionality or captured using

different sensors but must provide positional or directional information. That is, each data element must contain the spatial position of the element or the direction of the element with respect to the sensor.

While some of this material may apply to alignment problems where some or all of the correspondences between elements in the datasets are known, this thesis does not focus on those methods. It is important to distinguish between the two problem types, since they are related and sometimes interlinked. For example, methods that solve for transformation using correspondences are often used as subroutines in geometric alignment algorithms. However, the geometric alignment problem is the more general and challenging form, and does not assume that a satisfactory set of correspondences can be extracted from the sensor data.

3.1 Parametrisations of Rigid Motions

A rigid motion is a transformation that can be undertaken by a rigid body, that is, any combination of translations and rotations, excluding scaling and reflections. Also referred to as rigid transformations or isometries, rigid motions are elements of the Special Euclidean group $SE(n) = \mathbb{R}^n \times SO(n)$ of dimension n under the group operation of matrix multiplication. In this chapter, only rigid motions in two and three dimensions are considered, since they correspond to the physical motions relevant to the sensor data alignment problem. For example, while the intrinsic properties of a camera may change, the sensor itself is only subject to 3D translations and rotations. Hence, rigid motions include transformations undertaken by a sensor, such as a camera or laser scanner, or a rigid multi-sensor system. Rigid transformations also relate all Cartesian coordinate systems in $\mathbb{R}^n$ to one another, independent of a sensor. This becomes useful in the application of map merging, where the original sensor viewpoints may no longer be relevant.

Non-rigid transformations of sensor data are also relevant to many alignment problems, since objects that are observed by sensors may deform and the intrinsic parameters of the sensors may also change. However, these types of transformations will not be considered in this thesis. Instead, the rigid transformation of the sensor or the coordinate system is the focus. For a detailed survey of non-rigid alignment techniques, the reader is directed to Van Kaick et al. [2011].

In this section, parametrisations of translation and rotation space will be treated separately. Since the parametrisation of translation space is trivial, the majority of this section introduces the different rotation parametrisations and discusses the advantages and disadvantages of each.

3.1.1 Parametrisations of Translation

Translation space $\mathbb{R}^n$ may be parametrised in several ways, depending on the dimension. The 2D Euclidean space $\mathbb{R}^2$ may be parametrised by 2-vectors using a Cartesian, polar or other coordinate system. In a similar way, the 3D Euclidean space $\mathbb{R}^3$ may be parametrised by 3-vectors using a Cartesian, spherical or other coordinate system. In this work, Cartesian coordinates are used predominantly, being in many cases the most appropriate choice for geometry problems. Nonetheless, polar and spherical coordinate systems can simplify the equations in certain situations.

The domain of all translations is unbounded. However, some optimisation approaches require a bounded transformation domain. For these approaches, the space of translations is restricted to be within the bounded set Ω_t, which is typically a cuboid, for ease of manipulation.

3.1.2 Parametrisations of Rotation

Rotation space $SO(n)$ has a wide variety of parametrisations, which can be applied in different contexts and for different purposes. Three important considerations are computational efficiency, memory efficiency and theoretical insight. Computational efficiency refers to how long it takes a computer to perform operations with a given rotation representation, such as composing rotations together or applying the rotation to a vector or set of vectors. Memory efficiency refers to how much memory is required to store the rotation and how much temporary memory is required to store additional variables when the rotation is applied. Theoretical insight is more nebulous and refers to the observation that mathematical relationships may be more apparent in one parametrisation than another.

The Special Orthogonal group $SO(n)$ specifically refers to the group of $n \times n$ orthogonal matrices with determinant equal to 1 under the group operation of matrix multiplication. That is, an element of $SO(3)$ is a rotation in matrix representation form. In this work, the notation $SO(3)$ is often used as a shorthand to denote the group of (proper) rotations in the abstract, regardless of the rotation representation. This is justified by the existence of mappings between the matrix representation and the other rotation representations.

Three parametrisations are presented in this section: matrices, angle-axis vectors and quaternions. The last two only apply to three dimensional rotations. There are many other parametrisations that will not be discussed, including the scalar angle θ for 2D rotations and the Euler and Tait-Bryan angles for 3D rotations. The former is self-evident and the latter two provide no advantage over the other representations, are ambiguous without careful definition and are susceptible to singularities. Known

as gimbal lock, this susceptibility occurs because the map from the Euler angles to the rotations is not a covering map, therefore there are points in the parameter space where a change in rotation space cannot be realised by a change in the parameter space. As a generalisation, matrix or quaternion parametrisations are most suitable for applications that require computational efficiency and angle-axis or quaternion parametrisations are most suitable for applications that require memory efficiency. Any representation, however, may be useful for facilitating theoretical insight.

Matrix Representation

A rotation is frequently represented by an $n \times n$ orthogonal matrix with a determinant equal to 1. Matrices of this form are elements of the Special Orthogonal group $SO(n)$ under the group operation of matrix multiplication. Therefore, the set of all rotations is given by

$$\{\mathbf{R} \in \mathbb{R}^{n\times n} \mid \mathbf{R}^\intercal\mathbf{R} = \mathbf{R}\mathbf{R}^\intercal = \mathbf{I},\, \det(\mathbf{R}) = 1\} \tag{3.1}$$

where $\mathbf{I}$ is the $n \times n$ identity matrix and $\det(\mathbf{R})$ denotes the determinant of the matrix $\mathbf{R}$. While all orthogonal matrices have a determinant of ± 1, the condition on the determinant ensures that reflections in 2D and inversions and rotoinversions in 3D are not admitted.

In two dimensions, an arbitrary rotation matrix can be written as

$$\mathbf{R} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \tag{3.2}$$

where θ is the angle of (anti-clockwise) rotation. In three dimensions, there are many ways of writing a given rotation. One common form uses Euler angles, an example of which is given by

$$\mathbf{R} = \begin{pmatrix} \cos\varphi\cos\psi - \cos\theta\sin\varphi\sin\psi & -\cos\varphi\sin\psi - \cos\theta\sin\varphi\cos\psi & \sin\varphi\sin\theta \\ \sin\varphi\cos\psi + \cos\theta\cos\varphi\sin\psi & -\sin\varphi\sin\psi + \cos\theta\cos\varphi\cos\psi & -\cos\varphi\sin\theta \\ \sin\psi\sin\theta & \cos\psi\sin\theta & \cos\theta \end{pmatrix} \tag{3.3}$$

where the Euler angles are (φ, θ, ψ) and the rotation can be decomposed as the product of three elemental intrinsic rotations about the X and Z axes: $\mathbf{R} = \mathbf{Z}(\varphi)\mathbf{X}(\theta)\mathbf{Z}(\psi)$.

The matrix representation of rotations is related to the angle-axis and quaternion representations using results from Lie group theory. Restricting the analysis to three dimensions, $SO(3)$ is a Lie group, a manifold of dimension 3 embedded in $\mathbb{R}^{3\times3}$, that is associated with a Lie algebra $\mathfrak{so}(3)$ of all 3×3 skew-symmetric matrices. Any member of $\mathfrak{so}(3)$ can be represented by a 3-vector $\mathbf{x} \in \mathbb{R}^3$, with the mapping from $\mathbb{R}^3 \to \mathfrak{so}(3)$

given by the cross-product matrix

$$\mathbf{x} \mapsto [\mathbf{x}]_\times = \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix} \tag{3.4}$$

with $\mathbf{x} = [x, y, z]^\mathsf{T}$. The mapping from $\mathfrak{so}(3) \to SO(3)$ is given by the exponential map

$$\mathbf{A} \mapsto \exp(\mathbf{A}) = \sum_{k=0}^{\infty} \frac{1}{k!}\mathbf{A}^k \tag{3.5}$$

where $\exp(\cdot)$ is the matrix exponential function and $\mathbf{A}^0$ is defined to be the identity matrix $\mathbf{I}$ with the same dimensions as $\mathbf{A}$. For any skew-symmetric matrix $\mathbf{A} \in \mathfrak{so}(3)$, the matrix exponential has a closed form arising from Rodrigues' rotation formula. The mapping from $\mathfrak{so}(3) \to SO(3)$ is therefore given in closed form by

$$\mathbf{A} \mapsto \exp(\mathbf{A}) = \exp(\|\mathbf{a}\|\hat{\mathbf{A}}) = \mathbf{I} + (\sin\|\mathbf{a}\|)\hat{\mathbf{A}} + (1 - \cos\|\mathbf{a}\|)\hat{\mathbf{A}}^2. \tag{3.6}$$

where $[\mathbf{a}]_\times = \mathbf{A}$, $\|\mathbf{a}\|$ is the Euclidean norm of $\mathbf{a}$, and $\hat{\mathbf{A}} = \mathbf{A}\|\mathbf{a}\|^{-1}$. This formula will be revisited in the context of the angle-axis representation. The maps (3.4) and (3.6) together form a many-to-one mapping from $\mathbb{R}^3 \to SO(3)$.

The inverse mapping from $SO(3) \to \mathfrak{so}(3)$ is given by the logarithm map, where $\mathbf{A}$ is the matrix logarithm of $\mathbf{R}$ if $\exp\mathbf{A} = \mathbf{R}$. Observing that the first and third terms of the right-hand side of Rodrigues' rotation formula (3.6) are symmetric, the skew-symmetric part of both sides can be isolated. From this a closed form of the matrix logarithm for any rotation matrix $\mathbf{R} \in SO(3)$ can be found. The mapping from $SO(3) \to \mathfrak{so}(3)$ is therefore given in closed form by

$$\mathbf{R} \mapsto \log(\mathbf{R}) = \begin{cases} \dfrac{\arcsin(\|\mathbf{b}\|)}{\|\mathbf{b}\|}\mathbf{B} & \text{if } 1 \leqslant \operatorname{trace}(\mathbf{R}) < 3 \\ \dfrac{\pi - \arcsin(\|\mathbf{b}\|)}{\|\mathbf{b}\|}\mathbf{B} & \text{if } -1 < \operatorname{trace}(\mathbf{R}) < 1 \\ \mathbf{0} & \text{if } \operatorname{trace}(\mathbf{R}) = 3 \\ \pi \operatorname{sgn}(\mathbf{R}) \circ \left[\sqrt{\dfrac{1}{2}\operatorname{diag}(\mathbf{R} + \mathbf{I})}\right]_\times & \text{if } \operatorname{trace}(\mathbf{R}) = -1 \end{cases} \tag{3.7}$$

where $\mathbf{B} = \frac{1}{2}(\mathbf{R} - \mathbf{R}^\mathsf{T})$ is the skew-symmetric part of the rotation matrix, $\mathbf{B} = [\mathbf{b}]_\times$ for the 3-vector $\mathbf{b} = [b_1, b_2, b_3]^\mathsf{T}$, $\mathbf{0}$ is the zero matrix, $\operatorname{sgn}(\cdot)$ is the element-wise sign function, $\circ$ is the Hadamard (element-wise) matrix product and $\operatorname{diag}(\mathbf{M})$ is a function that extracts the diagonal elements of $\mathbf{M}$ into a vector $[m_{11}, m_{22}, m_{33}]^\mathsf{T}$. The last case

occurs when $\mathbf{R}$ has two eigenvalues equal to -1 and therefore represents a rotation of $\pm\pi$, for which there is not a unique logarithm. A simpler approach for this case is to find the unit eigenvector $\hat{\mathbf{v}}$ of $\mathbf{R}$ corresponding to the eigenvalue $\lambda = 1$ and then the matrix logarithm is given by $\pi[\hat{\mathbf{v}}]_\times$. An alternative, more compact form of the mapping from $SO(3) \to \mathfrak{so}(3)$, incorporating eigendecomposition, is given by

$$\mathbf{R} \mapsto \log(\mathbf{R}) = \begin{cases} \dfrac{\theta(\mathbf{R} - \mathbf{R}^\intercal)}{2\sin(\theta)} & \text{if } -1 < \text{trace}(\mathbf{R}) < 3 \\ \mathbf{0} & \text{if } \text{trace}(\mathbf{R}) = 3 \\ \pi[\hat{\mathbf{v}}]_\times & \text{if } \text{trace}(\mathbf{R}) = -1 \end{cases} \tag{3.8}$$

where $\theta = \arccos\Big(\frac{1}{2}(\text{trace}(\mathbf{R}) - 1)\Big)$ is the rotation angle. Irrespective of which form is used, the matrix $\log(\mathbf{R})$ from (3.7) or (3.8) is skew-symmetric in all cases. Therefore, together with the inverse mapping of (3.4), there exists a mapping from $SO(3) \to \mathbb{R}^3$.

The matrix representation is the most computationally efficient choice in many situations, particularly when applied to a set of vectors, such as a point-set. In three dimensions, it requires $9N$ multiplications and $6N$ additions when applied to a set of N vectors. However, it is less efficient than quaternions for composing rotations, requiring 27 multiplications and 18 additions. The matrix representation is over-parametrised, containing n^2 parameters in an n-dimensional space. The orthogonality constraints reduce the degrees of freedom to $\frac{1}{2}n(n-1)$. Therefore, the memory efficiency of this representation is poor, requiring $\frac{2n}{n-1}$ times the degrees of freedom in storage. For 2D space, 1 rotational degree of freedom requires 4 parameters. For 3D space, 3 rotational degrees of freedom require 9 parameters. In terms of theoretical insight, the matrix representation has been studied extensively and has well-known properties. In particular, the rotation angle of a rotation matrix is well-defined using the trace, as is the angle between two rotation matrices. Moreover, several theorems exist that relate the angle between rotation matrices to other angle measures. See Section 3.2.2 for further details.

Angle-Axis Representation

The angle-axis representation arises from Euler's rotation theorem that proves the equivalence of a sequence of rotations in three-dimensional space to a single rotation about a fixed axis. The representation provides an intuitive way of thinking about rotations while also being a minimal parametrisation that is not susceptible to singularities, unlike the minimal Euler angle parametrisations.

Therefore, every rotation in $SO(3)$ can be represented as an angle-axis 3-vector $\mathbf{r}$ with a rotation angle $\theta = \|\mathbf{r}\|$, the Euclidean norm of $\mathbf{r}$, about a unit rotation axis

$\hat{\mathbf{r}} = \mathbf{r}/\|\mathbf{r}\|$, encoded in $\mathbb{R}^3$ as

$$\mathbf{r} = \theta\hat{\mathbf{r}}. \tag{3.9}$$

The angle-axis representation can also be written as an ordered scalar–vector pair $(\theta, \hat{\mathbf{r}})$. The parameters of the angle-axis representation, also known as the Rodrigues parameters, are in $\mathbf{S}^1 \times \mathbf{S}^2$, that is a scalar angle (parametrising a vector on the unit 1-sphere) multiplied by a rotation axis (a vector on the unit 2-sphere).

The mapping from angle-axis vectors to rotation matrices $\mathbb{R}^3 \to SO(3)$ is given by the exponential map over the Lie algebra matrix induced by the angle-axis vector, as previously discussed. That is, matrix exponentiation is applied to the skew-symmetric matrix $[\mathbf{r}]_\times$ induced by $\mathbf{r}$ in order to retrieve the rotation matrix, denoted $\mathbf{R}_\mathbf{r}$. For angle-axis vectors, the Rodrigues' rotation formula can be used to efficiently calculate the exponential map in closed form [Hartley and Zisserman, 2003]. The mapping from $\mathbb{R}^3 \to SO(3)$ is therefore given as

$$\mathbf{r} \mapsto \mathbf{R}_\mathbf{r} = \exp([\theta\hat{\mathbf{r}}]_\times) = \mathbf{I} + (\sin\theta)[\hat{\mathbf{r}}]_\times + (1 - \cos\theta)[\hat{\mathbf{r}}]_\times^2 \tag{3.10}$$

where $\mathbf{I}$ is the 3×3 identity matrix and $[\hat{\mathbf{r}}]_\times$ is the cross-product matrix of $\hat{\mathbf{r}}$.

The mapping from $SO(3) \to \mathbb{R}^3$ involves the matrix logarithm and is given as

$$\mathbf{R}_\mathbf{r} \mapsto \mathbf{r} = \begin{cases} \dfrac{\theta\mathbf{b}}{\sin(\theta)} & \text{if } -1 < \text{trace}(\mathbf{R}) < 3 \\ \mathbf{0} & \text{if } \text{trace}(\mathbf{R}) = 3 \\ \pi\hat{\mathbf{v}} & \text{if } \text{trace}(\mathbf{R}) = -1 \end{cases} \tag{3.11}$$

where $\theta = \arccos\Big(\frac{1}{2}(\text{trace}(\mathbf{R}) - 1)\Big)$ is the rotation angle, $\mathbf{b}$ is the 3-vector whose corresponding cross-product matrix $[\mathbf{b}]_\times = \frac{1}{2}(\mathbf{R} - \mathbf{R}^\intercal)$ is the skew-symmetric part of the rotation matrix, $\mathbf{0}$ is the zero 3-vector, and $\hat{\mathbf{v}}$ is the unit eigenvector of $\mathbf{R}$ corresponding to the eigenvalue $\lambda = 1$, that is, satisfying $\mathbf{R}\hat{\mathbf{v}} = \hat{\mathbf{v}}$. Indeed, the rotation axis can always be solved by eigendecomposition, up to a sign ambiguity.

Rotations are not uniquely represented in angle-axis form, since the rotation encoded by $\theta\hat{\mathbf{r}}$ is equivalent to $(2k\pi + \theta)\hat{\mathbf{r}}$ and $(2k\pi - \theta)(-\hat{\mathbf{r}})$ for $k \in \mathbb{Z}$. To ensure that most of the encoded rotations are unique, the angle θ can be restricted to $[0, \pi]$. As a result, the space of all 3D rotations can be represented as a solid, closed ball of radius π in $\mathbb{R}^3$, denoted as B^3_π. The mapping from $B^3_\pi \to SO(3)$ is one-to-one on the interior of the π-ball and two-to-one on the surface. The redundancy occurs at antipodal points on the surface of the ball, since for a rotation of π radians, the direction of the rotation axis is immaterial. The trade-off for restricting the parameter values is that a path through B^3_π must jump to a different region when it reaches the surface.

In order to be composed with another rotation or applied to a set of vectors, an angle-axis vector must first be converted to its matrix or quaternion representation. As a result, they are not computationally efficient and are often only used as an intermediate step. However, since they are minimal parametrisations, they are optimally memory-efficient, requiring only three parameters to entirely define the rotation. Thus, they are an appropriate choice for storing rotations when memory is constrained.

Critically, the angle-axis representation provides useful theoretical intuitions. For example, it can be used to visualise rotations in 3D space, which may provide geometric insight. It also facilitates the subdivision of rotation space using standard techniques such as octrees. Moreover, it is naturally decomposable into a scalar rotation angle and a vector rotation axis which makes it useful for measuring and bounding changes in rotation angle and direction separately. Finally, manipulating $\mathbb{R}^3$ instead of $SO(3)$ admits addition, commutativity and scaling, since $\mathbb{R}^3$ is a vector space, unlike $SO(3)$.

One disadvantage of the angle-axis representation is that a uniform subdivision of angle-axis space $B^3_\pi \in \mathbb{R}^3$ does not correspond to a uniform subdivision of rotation space $SO(3)$. Visualising $SO(3)$ as a hemisphere of the unit 3-sphere S^3, the corresponding subdivision is warped and the elements are of non-uniform size. This occurs because the Euclidean distance between angle-axis vectors is only an approximation of the distance between the equivalent rotations on the rotation manifold [Li and Hartley, 2007]. Indeed, Hartley and Kahl [2009] observe that the angle-axis representation can be thought of as an azimuthal-equidistant projection of S^3, flattening the upper hemisphere and causing tangential stretching at the periphery. While an exactly uniform subdivision of S^3 is non-trivial, a more uniform subdivision was investigated for the rotation search problem in Straub et al. [2017], who demonstrated that uniformity was advantageous for the efficiency of their search algorithm.

Quaternion Representation

Quaternions, conceived by Hamilton [1844], extend the concept of complex numbers and comprise four real numbers (w, x, y, z) and four basis elements $(1, i, j, k)$ and can be written as $q = w1 + xi + yj + zk$. The basis elements i, j, k are unit imaginary numbers with the property that $i^2 = j^2 = k^2 = ijk = -1$. Alternatively, a quaternion can be represented as a 4-vector $\mathbf{q} = w\mathbf{1} + x\mathbf{i} + y\mathbf{j} + z\mathbf{k} = [w, x, y, z]^\intercal$ with respect to the unit basis vectors $(\mathbf{1}, \mathbf{i}, \mathbf{j}, \mathbf{k})$ in $\mathbb{R}^4$ or as a scalar–vector pair $\mathbf{q} = (w, \mathbf{v})$ where $\mathbf{v}$ is the imaginary vector $[x, y, z]^\intercal$.

Some useful mathematical operations arise from these definitions. The norm, conjugate and reciprocal of a quaternion $\mathbf{q} = (w, \mathbf{v})$ are given by the Euclidean norm $\|\mathbf{q}\| = \sqrt{w^2 + x^2 + y^2 + z^2}$, the quaternion conjugate $\mathbf{q}^* = (w, -\mathbf{v})$ and the quater-

nion inverse $\mathbf{q}^{-1} = \mathbf{q}^* \|\mathbf{q}\|^{-2}$ respectively. The non-commutative Hamilton product for two quaternions $\mathbf{q}_1 = (w_1, \mathbf{v}_1)$ and $\mathbf{q}_2 = (w_2, \mathbf{v}_2)$ is given by

$$\mathbf{q}_1\mathbf{q}_2 = (w_1 w_2 - \langle \mathbf{v}_1, \mathbf{v}_2 \rangle, w_1\mathbf{v}_2 + w_2\mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2) \tag{3.12}$$

where $\langle \cdot, \cdot \rangle$ is the inner product operator and $\times$ is the vector cross product. Finally, the conjugation of $\mathbf{q}_2$ by $\mathbf{q}_1$ is given by $\mathbf{q}_1\mathbf{q}_2\mathbf{q}_1^{-1}$.

A 3D rotation can be represented as a unit quaternion $\hat{\mathbf{q}}$ where $\|\hat{\mathbf{q}}\| = 1$. The unit Euclidean norm constraint reduces the degrees of freedom to the required three for rotation. The set of all unit quaternions is the 3-sphere S^3, a smooth manifold embedded in $\mathbb{R}^4$. Under the quaternion product, the unit quaternions form a Lie group which is a double cover of the rotation group $SO(3)$. Rotations $\hat{\mathbf{q}}_1$ and $\hat{\mathbf{q}}_2$ can be composed using the quaternion product $\hat{\mathbf{q}}_2\hat{\mathbf{q}}_1$, which corresponds to the rotation $\hat{\mathbf{q}}_1$ followed by the rotation $\hat{\mathbf{q}}_2$. To rotate a vector $\mathbf{p}$ in $\mathbb{R}^3$ by a unit quaternion, the vector $\mathbf{p} = [x, y, z]^\intercal$ is notated as a quaternion with real part equal to zero, that is $\mathbf{q}_\mathbf{p} = (0, \mathbf{p})$. The conjugation of this quaternion by a unit quaternion $\hat{\mathbf{q}} = (w, \mathbf{v})$ is given by $\hat{\mathbf{q}}\mathbf{q}_\mathbf{p}\hat{\mathbf{q}}^{-1}$ and corresponds to the rotation of vector $\mathbf{p}$ about the axis $\mathbf{v}$ by an angle $2\arccos(w)$. A more efficient but less compact formula for rotating a vector $\mathbf{p}$ by the unit quaternion is given by $\mathbf{p} + 2\mathbf{v} \times (\mathbf{v} \times \mathbf{p} + w\mathbf{p})$.

The scalar-vector pair notation provides an intuition as to how unit quaternions are related to the angle-axis representation. For an angle-axis vector $\mathbf{r} = \theta\hat{\mathbf{r}}$, the mapping from $\mathbb{R}^3 \to \mathbb{R}^4$ is given by

$$\mathbf{r} \mapsto \hat{\mathbf{q}} = \left(\cos\frac{\theta}{2}, \sin\frac{\theta}{2}\hat{\mathbf{r}}\right). \tag{3.13}$$

For a unit quaternion $\hat{\mathbf{q}} = (w, \mathbf{v})$, the mapping from $\mathbb{R}^4 \to \mathbb{R}^3$ is given by

$$\hat{\mathbf{q}} \mapsto \mathbf{r} = \begin{cases} 2\arccos(|w|)\mathbf{v}\|\mathbf{v}\|^{-1} & \text{if } |w| \neq 1 \\ \mathbf{0} & \text{if } |w| = 1 \end{cases} \tag{3.14}$$

where $\mathbf{0}$ is the zero 3-vector and $\arccos(|w|)$ can be evaluated using the two-argument arctangent function $\operatorname{atan2}(\|\mathbf{v}\|, |w|)$ for greater numerical stability. Taking the absolute value of w accounts for the sign ambiguity in the unit quaternion with the positive sign chosen to ensure that the angle θ is in $[0, \pi]$, as required. The mappings between quaternions and rotation matrices can be defined using the angle-axis mappings previously described. However, for a rotation matrix $\mathbf{R}$, an explicit mapping from $SO(3) \to \mathbb{R}^4$ is given by

$$\mathbf{R} \mapsto \hat{\mathbf{q}} = \begin{cases} \left(w, \dfrac{1}{2w}\mathbf{b}\right) & \text{if } w \neq 0 \\ (0, \hat{\mathbf{u}}) & \text{if } w = 0 \end{cases} \tag{3.15}$$

where $w = \frac{1}{2}\sqrt{1 + \text{trace}(\mathbf{R})}$, $\mathbf{b}$ is the 3-vector whose corresponding cross-product matrix $[\mathbf{b}]_\times = \frac{1}{2}(\mathbf{R} - \mathbf{R}^\intercal)$ is the skew-symmetric part of the rotation matrix, and $\hat{\mathbf{u}}$ is the unit eigenvector of $\mathbf{R}$ corresponding to the eigenvalue $\lambda = 1$, satisfying $\mathbf{R}\hat{\mathbf{u}} = \hat{\mathbf{u}}$. Finally, for a unit quaternion $\hat{\mathbf{q}}$, the mapping from $\mathbb{R}^4 \to SO(3)$ is given by

$$\hat{\mathbf{q}} \mapsto \mathbf{R} = \Big(w^2 - \mathbf{v}^\intercal\mathbf{v}\Big)\mathbf{I} + 2\mathbf{v}\mathbf{v}^\intercal + 2w[\mathbf{v}]_\times \tag{3.16}$$

where $\mathbf{I}$ is the 3×3 identity matrix and $[\mathbf{v}]_\times$ is the cross-product matrix of $\mathbf{v}$.

A significant advantage of the quaternion representation is its computational efficiency. In particular, it requires 17 fewer operations to compose two unit quaternions than to compose two rotation matrices, using 16 multiplications and 12 additions for the Hamilton product. However, quaternions are slightly less efficient than matrices at rotating vector sets, requiring $15N$ multiplications and $15N$ additions when applied to a set of N vectors, in contrast to $9N$ and $6N$ for matrices. Indeed, it is more efficient to first convert the quaternion into a rotation matrix for $N > 1$, requiring only $12+9N$ multiplications and $12 + 6N$ additions in this case. The quaternion representation also provides a meaningful mechanism to reduce numerical errors during computation. Normalising the quaternion ensures that the object is a valid rotation and is more computationally efficient than Gram-Schmidt orthonormalisation of a rotation matrix. This helps reduce numerical errors caused by repeated rounding in floating point arithmetic. Another advantage is its memory efficiency. It is a compact representation, requiring only four parameters to be stored in memory. By this measure, it is preferable to the matrix representation and is comparable to the angle-axis representation. Finally, the quaternion representation provides several useful theoretical insights. In particular, they provide a way to visualise $SO(3)$ as a hemisphere of S^3. The hypersphere S^3 in $\mathbb{R}^4$ is a double covering of $SO(3)$ in which antipodal points represent the same rotation. For rotation angles in $[0, \pi]$, rotations are mapped one-to-one to the upper 'northern' hemisphere except on the 'equator'. The 'North Pole' of this representation is the identity rotation $\hat{\mathbf{q}} = [1, 0, 0, 0]^\intercal$.

3.2 Distance Measures for Rigid Transformations

This section provides a general overview of the distance measures or metrics that are used to measure transformations in $SE(n)$. It is not exhaustive, but instead focuses on those that are frequently used in geometric sensor data alignment and this work in particular. The treatment is divided into translation and rotation measures as before, with the predominant focus being on rotation measures. For the group of translations, $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$ is a distance measure that maps two vectors in $\mathbb{R}^n$ to a non-

negative real number. Similarly, for the group of rotations, $d : SO(n) \times SO(n) \to \mathbb{R}^+$ is a distance measure that maps two elements of the rotation group in $SO(n)$ to a non-negative real number.

In this section, L_p-norm functions are written with their appropriate subscripts. However, whenever a norm subscript is not notated in this document, the norm is to be understood as an L_2 norm unless otherwise specified. In addition, the subscript notation L_p is used instead of the more common superscript notation L^p for consistency with the computer vision literature.

3.2.1 Euclidean Distance

The natural distance measure for translations is the Euclidean distance d_{E}, not least because the translation group is isomorphic to Euclidean space. For the group of translations, the Euclidean distance between two n-vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^n$ is given by the L_2-norm of the vector difference, that is

$$d_{\mathrm{E}}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2 = \sqrt{(\mathbf{x} - \mathbf{y})^\intercal (\mathbf{x} - \mathbf{y})} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3.17}$$

where x_i and y_i are the components of $\mathbf{x}$ and $\mathbf{y}$ respectively. This metric is suitable for both 2D and 3D translations. When the only relevant information is the relative ordering of translations from a fixed translation, the squared Euclidean distance may be used instead, which is more efficient to calculate.

3.2.2 Angular Distance

The natural distance measure for rotations is the angular distance $d_\angle$ that takes values in the range $[0, \pi]$. The angular distance between a rotation $\mathbf{R}$ and the identity rotation $\mathbf{I}$ is given by the angle of rotation $\angle(\mathbf{R})$ of the matrix $\mathbf{R}$, chosen such that $0 \leqslant \angle(\mathbf{R}) \leqslant \pi$. In 2D, this corresponds to the (unsigned) rotation angle about the origin, reversing the direction of rotation if necessary. In 3D, this corresponds to the rotation angle about an axis, reversing the direction of the axis if necessary. In both cases, the rotation angle is the angle between $\mathbf{v}_\perp$ and $\mathbf{R}\mathbf{v}_\perp$ where $\mathbf{v}_\perp$ is a vector perpendicular to the axis of rotation. For two rotation matrices $\mathbf{R}_1$ and $\mathbf{R}_2$, the angular distance $d_\angle(\mathbf{R}_1, \mathbf{R}_2)$ is defined as the angle of rotation $\angle(\mathbf{R}_1^\intercal \mathbf{R}_2)$ of the matrix $\mathbf{R}_1^\intercal \mathbf{R}_2$, or equivalently $\mathbf{R}_1 \mathbf{R}_2^\intercal$, $\mathbf{R}_2^\intercal \mathbf{R}_1$ or $\mathbf{R}_2 \mathbf{R}_1^\intercal$, again chosen to be in the range $[0, \pi]$ [Hartley and Kahl, 2009].

For $\mathbf{R} \in SO(2)$ or $\mathbf{R} \in SO(3)$, the angular distance is given by

$$d_\angle(\mathbf{R}_1, \mathbf{R}_2) = d_\angle(\mathbf{R}_1^\intercal \mathbf{R}_2, \mathbf{I}) = \frac{1}{\sqrt{2}} \| \log(\mathbf{R}_1^\intercal \mathbf{R}_2) \|_{\mathrm{F}} \tag{3.18}$$

where $\|\mathbf{A}\|_{\mathrm{F}}$ denotes the Frobenius norm given by $\sqrt{\sum\sum |a_{ij}|^2}$. The factor of $(\sqrt{2})^{-1}$ arises from the skew-symmetry of $\log(\mathbf{R})$ and the observation that the rotation angle is given by $\|\mathbf{r}\|_2$, for $[\mathbf{r}]_\times = \log(\mathbf{R})$. Taking the Frobenius norm of both sides gives the expression $\|\log(\mathbf{R})\|_{\mathrm{F}} = \|[\mathbf{r}]_\times\|_{\mathrm{F}} = \sqrt{2}\|\mathbf{r}\|_2$.

In 2D, the angular distance can be expressed explicitly as

$$d_\angle(\mathbf{R}_1, \mathbf{R}_2) = \arccos\left(\frac{\operatorname{trace}(\mathbf{R}_1^\intercal \mathbf{R}_2)}{2}\right). \tag{3.19}$$

It may also be calculated using $\min(|\theta_1 - \theta_2|, 2\pi - |\theta_1 - \theta_2|)$ where $\theta_i = \angle(\mathbf{R}_i)$. In 3D, the angular distance can be written as

$$d_\angle(\mathbf{R}_1, \mathbf{R}_2) = \arccos\left(\frac{\operatorname{trace}(\mathbf{R}_1^\intercal \mathbf{R}_2) - 1}{2}\right). \tag{3.20}$$

These trace expressions make use of the properties of the eigenvalues of two- and three-dimensional rotation matrices. An alternative approach for 3D rotations is to compute the angle from the quaternion representation of the rotation matrices. If $\hat{\mathbf{q}}_1$ and $\hat{\mathbf{q}}_2$ are unit quaternions representing the same rotations as $\mathbf{R}_1$ and $\mathbf{R}_2$ respectively, then

$$d_\angle(\mathbf{R}_1, \mathbf{R}_2) = 2\arccos(|w|) = 2\arccos|\langle \hat{\mathbf{q}}_1, \hat{\mathbf{q}}_2 \rangle| \tag{3.21}$$

where $(w, \mathbf{v}) = \hat{\mathbf{q}}_2^{-1}\hat{\mathbf{q}}_1$ using the Hamilton product, $\langle \cdot, \cdot \rangle$ is the quaternion inner product and the positive absolute value is taken to ensure that the angle lies in the range $[0, \pi]$ as required [Hartley et al., 2013].

3.2.3 Chordal Distance

Another rotation distance measure is found by calculating the Euclidean distance between two rotation matrices in their embedding space $\mathbb{R}^{n\times n}$. Designated the chordal distance d_{C} between two matrices $\mathbf{R}_1$ and $\mathbf{R}_2$, it is given by

$$d_{\mathrm{C}}(\mathbf{R}_1, \mathbf{R}_2) = \|\mathbf{R}_1 - \mathbf{R}_2\|_{\mathrm{F}} \tag{3.22}$$

where the Frobenius norm $\|\cdot\|_{\mathrm{F}}$ is an extension of the Euclidean norm to matrices. An advantage of this measure is that is computationally inexpensive, without matrix multiplications or trigonometric functions.

For rotations in both $SO(2)$ and $SO(3)$, Hartley et al. [2013] show that the angular distance is related to the chordal distance by

$$d_{\mathrm{C}}(\mathbf{R}_1, \mathbf{R}_2) = 2\sqrt{2}\sin\left(\frac{1}{2}d_\angle(\mathbf{R}_1, \mathbf{R}_2)\right). \tag{3.23}$$

3.2.4 Quaternion Distance

Similar to the definition of the chordal distance, the quaternion distance is found by calculating the Euclidean distance between two unit quaternions in their embedding space $\mathbb{R}^4$. However, since antipodal points of the unit quaternion sphere are identified, $\hat{\mathbf{q}}$ and $-\hat{\mathbf{q}}$ represent the same rotation. The problem of choosing the correct sign is resolved by defining the quaternion distance between two unit quaternions $\hat{\mathbf{q}}_1$ and $\hat{\mathbf{q}}_2$, the quaternion representations of rotation matrices $\mathbf{R}_1$ and $\mathbf{R}_2$ respectively, as

$$d_{\mathrm{Q}}(\mathbf{R}_1, \mathbf{R}_2) = \min\{\|\hat{\mathbf{q}}_1 - \hat{\mathbf{q}}_2\|_2, \|\hat{\mathbf{q}}_1 + \hat{\mathbf{q}}_2\|_2\} \tag{3.24}$$

where both the positive and negative branches of $\hat{\mathbf{q}}_2$ are considered. Unlike the angular and chordal distances, the quaternion distance does not exist for 2D rotations.

As obtained in Hartley et al. [2013], the angular distance is related to the quaternion distance by

$$d_{\mathrm{Q}}(\mathbf{R}_1, \mathbf{R}_2) = 2\sin\left(\frac{1}{4} d_{\angle}(\mathbf{R}_1, \mathbf{R}_2)\right). \tag{3.25}$$

3.2.5 Angle-Axis Distance

The angle-axis distance is also defined using the Euclidean distance, in this case between two angle-axis vectors in $\mathbb{R}^3$. However, if the angle-axis vectors are restricted to lie within the ball B^3_π of radius π, as previously defined, then this measure would have discontinuities. For example, rotations close to π radians about opposite axes are close by the angular distance measure but not close by the Euclidean distance measure in B^3_π, since they are almost antipodal. To remove these discontinuities, the angle-axis distance d_{AA} between two rotation matrices $\mathbf{R}_1$ and $\mathbf{R}_2$ in $SO(3)$ considers all equivalent angle-axis vectors $\mathbf{r}_1$ and $\mathbf{r}_2$, including those outside B^3_π, and is defined as

$$d_{\mathrm{AA}}(\mathbf{R}_1, \mathbf{R}_2) = \min_{\mathbf{r}_1, \mathbf{r}_2} \|\mathbf{r}_1 - \mathbf{r}_2\|_2 \tag{3.26}$$

where $\exp([\mathbf{r}_1]_\times) = \mathbf{R}_1$ and $\exp([\mathbf{r}_2]_\times) = \mathbf{R}_2$. Like the quaternion distance, the angle-axis distance does not exist for 2D rotations.

As observed in Hartley et al. [2013], the angle-axis distance is not bi-invariant. Therefore, there is no equality relationship between the angular distance and the angle-axis distance. However, the useful inequalities

$$d_{\angle}(\mathbf{R}_1, \mathbf{R}_2) \leqslant d_{\mathrm{AA}}(\mathbf{R}_1, \mathbf{R}_2) \leqslant \frac{\pi}{2} d_{\angle}(\mathbf{R}_1, \mathbf{R}_2) \tag{3.27}$$

can be shown for this distance measure [Li and Hartley, 2007; Hartley and Kahl, 2009].

3.3 Sensor Data Representations

Sensor data can be represented in many different ways, each with concomitant advantages and disadvantages. It is important to recognise that sensor data has been sampled from real (or simulated) surfaces. At macroscopic scales, the observable part of real-world scenes are primarily composed of a single continuous surface, with airborne objects forming additional disjoint surfaces. Despite the complexity of real-world scenes, they should in principle be able to be well-represented by a 2D manifold. However, when data is collected from the environment, it is invariably sampled from this continuous surface at discrete locations, regardless of the sensor used.

In addition to the loss of information engendered by discretely sampling a continuous surface, the samples themselves are subject to sensor noise. Sensor data taken from a single viewpoint at a single instant of time can be viewed as a signal that may contain additional undesired signals representing information that is not present in the observed scene. These noise signals consist of small-scale errors and can be modelled using probability distribution functions. Sensor noise is typically assumed to be Gaussian, additive and independent at each measurement, caused primarily by thermal noise. Other noise sources are modelled using the uniform distribution for quantisation noise, the Poisson distribution for shot noise, and fat-tailed distributions for impulsive salt-and-pepper noise. This problem is exacerbated by data collected over a finite time period, particularly when the sensor or objects in the scene are moving. These time-displaced signals result in motion blur, ghosting and other inconsistencies.

The other major problem associated with sensor data arises when two or more sets (frames) of data are considered together. Outliers are points or pixels that do not correspond between sets of sensor data taken from different viewpoints or at different times. Outliers may be random or structured, with random outliers often being caused by significant sensor noise (such as salt-and-pepper noise) and structured outliers being caused by occlusion. In the latter case, parts of a scene may be occluded from one viewpoint but not another, resulting in partially-overlapping observations. In the context of alignment problems where correspondences are provided, the term 'outlier' often denotes an outlier (incorrect) correspondence. However, this terminology will not be used without clarification in this work.

As previously mentioned, motion can be another confounding factor when scenes are observed over multiple instances of time. Under certain assumptions, a scene can be decomposed into a static part and a dynamic part. The static part consists of infrequently-changing surfaces such as buildings, in contrast to the dynamic part that consists of moving objects such as vehicles. A static assumption is reasonable for many man-made structures and elements of the natural world, but is less reasonable for fixed,

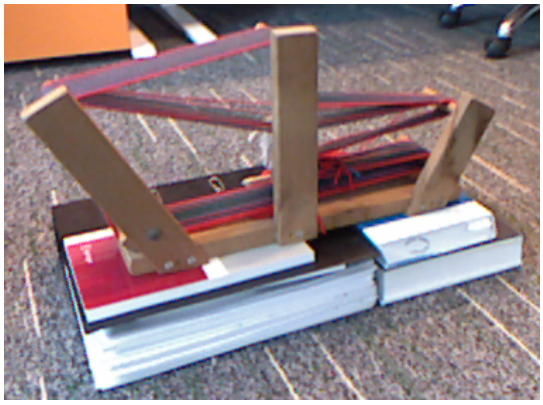
deformable objects such as trees. Even rigid man-made structures cannot be considered static parts of a scene indefinitely, since they are liable to be removed or altered at some point. Nonetheless, sensor motion is calculated with respect to the static part of a scene, which is assumed to be fixed. In the context of this work, dynamic objects are treated in a twofold way. Within a single frame, the motion blur of a moving object is considered to be intra-frame noise. Between several frames, a moving object is considered to consist of structured outliers, points or pixels without correspondences in another frame of sensor data. Dynamic sensors also generate noise and structured outliers, however these are present in both the dynamic and static portions of the scene.

The entities to be aligned can take many forms and range from discrete point-sets or images to continuous surfaces or probability densities. Geometric sensor data can be classified as positional or directional, that is, containing the spatial position or direction of the sample with respect to the sensor. In addition, sensor data representations can be classified as raw or processed, with processed representations having been generated for some advantage, being more compact, continuous, visually-appealing or amenable to calculation than the raw data. For this classification, the first category of data representation is the raw output of the sensor. The term raw is perhaps a misnomer for this data type, since some low-level processing has already occurred by this stage, however these data types are typically available directly from the sensor. This category includes colour or greyscale images, depth images and 2D or 3D point-sets. The second category involves higher-level processing of the sensor data. The types of processed sensor data representations are multifarious, and include bearing vector sets, meshes, primitives, occupancy grids, spherical harmonic images and mixture models.

An ideal sensor data representation would model the underlying surface accurately and completely. However, given noisy and incomplete observations, a good sensor data representation should be

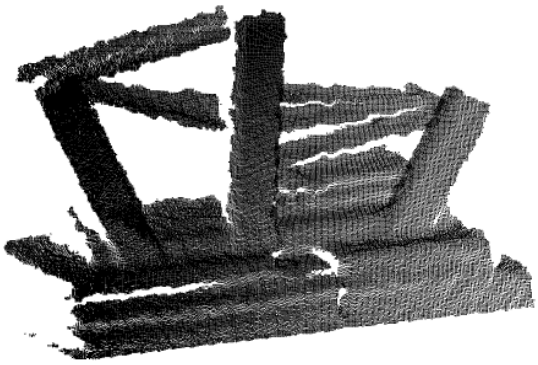
1. compact for memory efficiency;
2. robust or invariant to sensor noise;
3. robust to structured outliers;
4. adaptive to local surface complexity without over-smoothing; and
5. data-driven, imposing minimal structure on the recovered surface.

These criteria are a guide to selecting a good representation, but should not be taken as being necessary or sufficient. In the final analysis, the best representation is the one that generates the best results for the task of geometric alignment. Robustness to structured outliers is an important criterion, because occlusions and missing data are extremely prevalent in data captured from the real world. Using knowledge about the sensor pose to model unobserved regions or using a probabilistic Bayesian approach that does not put too much credence on any one observation can improve the outlier

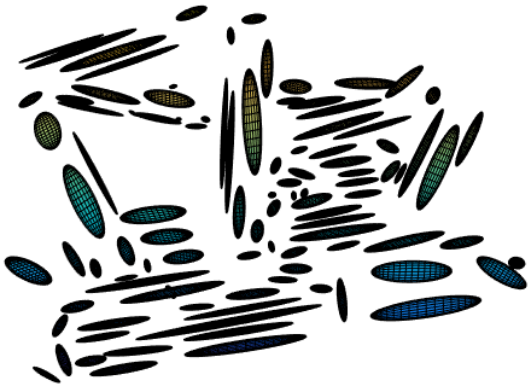


(a) Greyscale Image

(b) Depth Image



(c) 3D Point-Set



(d) Gaussian Mixture Model

Figure 3.1: Four sensor data representations of a loom, captured by an RGB-D camera.

robustness of a sensor data representation. Overall, the representation should attempt to discern the underlying structure without imposing a strong prior.

An overview of the raw and processed sensor data representations used in this work follows. For the raw data types, the sensors used to collect the data will be discussed, including their predominant noise characteristics. The advantages and disadvantages of the data types will also be discussed, in the context of the criteria outlined above. Examples of various sensor data representations are shown in Figure 3.1.

3.3.1 Point-Sets

Point-sets, also known as point clouds, are a discrete spatial data representation that at minimum contain 2D or 3D positional information. That is, each data element must contain the spatial position of the observed element but may also contain additional information, such as reflectance, colour, labelling, mesh structure or a feature descriptor. This data type is very prevalent, since it is a standard low-level representation provided by many sensors. An example is shown in Figure 3.1(c).

Point-sets can be captured with a diverse range of sensors. One class of sensor used frequently in surveying and robotics is the lidar sensor. A portmanteau of 'light' and 'radar', lidar is an active sensing technique that makes point-wise depth measurements with a range finder, based on the time-of-flight, wavelength shift or triangulation of laser pulses. In a scanning lidar system, the device changes the viewing direction of the range finder to take measurements across its field-of-view. In a scannerless lidar system such as a time-of-flight or depth camera, the device consists of an array of depth sensors. Due to its mode of operation, lidar is susceptible to noise caused by reflective or transparent surfaces. In particular, transparent or translucent materials such as glass, water and airborne particles can cause non-Gaussian and potentially structured noise, with ghosting effects being common. Moreover, since most scanning lidar systems produce a 2D or 3D point-set by reflecting the light pulses on a rotating mirror, the signals are not received simultaneously, generating a form of motion blur. In addition, the scale of the noise is distance-dependent, with small angular errors corresponding to large spatial errors for points far from the sensor. Finally, the point density is also distance-dependent, with surfaces near the sensor being sampled much more densely than distant surfaces. Lidar technology is used in many commercial sensors, including the Velodyne systematic scanning sensor, the Zebedee unsystematic scanning sensor [Bosse et al., 2012] and the Microsoft Kinect V2 scannerless sensor.

Another class of active sensor for capturing point-sets is the structured light sensor. This technology makes depth measurements by projecting a pattern of light on the scene and quantifying the deformation of the pattern when viewed by a camera that is offset from the projector. While noise associated with reflective and transparent surfaces is also common for these sensors, they are not subject to the same problems of motion distortion as lidar systems since they capture the entire scene at one instant of time. However, interference from light sources other than the projector can cause incorrect or inaccurate detections, with outdoor operation being a challenge for early systems. Many commercial structured light systems exist, with the original Microsoft Kinect (V1) sensor being a well-known example.

Passive sensors, such as RGB cameras, can also be used to construct point-sets. While this data representation is not the raw output of the sensor, Structure-from-Motion (SfM) and stereopsis algorithms can extract 3D structure from 2D images using the principles of multiple view geometry. In many cases, the scale of these point-sets is unknown and must be ascertained separately before distance measurements can be made. Also, each point in an SfM point-set is associated with an image feature, such as a SIFT feature, from which correspondences were found. This attribute can be very useful for alignment algorithms. A unique property of the noise characteristics for these point-sets is that they may contain reconstruction errors from incorrect or imprecise

correspondences. SfM and stereopsis are mature technologies that are deployed in commercial applications such as Google's Tango platform for smartphones and tablets.

The point-set data representation has several advantages. Firstly, this data type is a standard raw output of many lidar and structured light sensors. In addition, the data points of these active sensors represent physical measurements of real surfaces that can be very accurate and do not impose external structure on the underlying surface. Moreover, the point-set data type is well-supported by 3D visualisation and processing software, and code libraries for point-set processing are prevalent, including Point Cloud Library [Rusu and Cousins, 2011]. However, the point-set representation also has many shortcomings. When captured by an active sensor, the point-set is a non-uniform discrete sampling of a continuous surface. It has significant redundancy, with high point densities near the sensor and more samples of many surfaces than parameters needed to minimally represent them, particularly for planar surfaces. For a given level of detail, point-sets are therefore not memory-efficient. Moreover, the representation is not robust to sensor noise or structured outliers and does not adapt to local surface complexity.

In this work, set notation is used to describe point-sets. That is, the point-set $\mathcal{P}$ containing the 2D or 3D points $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N$ is given by

$$\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^{N}. \tag{3.28}$$

3.3.2 Depth Images

A depth image is a structured 2D array of depth values, where each pixel records the distance to the closest surface in that direction. They are a subset of the 3D point-set data representation, since the depth measurements at each pixel of a depth image can be converted into a structured point-set if the intrinsic camera calibration parameters are known. Accordingly, depth images are positional sensor data representations. An example is shown in Figure 3.1(b).

Depth images can be captured by several sensors, including time-of-flight, structured light and stereo cameras. These previously-discussed sensors are distinct from general 3D sensors because all the depth measurements within the field-of-view are made simultaneously and are structured in a grid pattern. This is similar to a normal camera and therefore the natural data representation is an image, with a depth channel instead of the colour channels.

The depth image data representation has many of the same advantages and disadvantages as point-sets. For example, depth images are a low-level output of depth cameras, consist of physical measurements that can be quite accurate and are well-supported by software packages and code libraries. However, they do have several

unique advantages. Firstly, they are not subject to the same level of motion distortion as asynchronous sensors such as lidar scanners because all the depth measurements are made simultaneously. Additionally, the grid structure can be exploited by many algorithms originally designed to process image data. For example, a depth image can be used as an input to a convolutional neural network, whereas this is non-trivial for an unstructured point-set. Depth images also have many disadvantages. Like lidar sensors, depth cameras sample a continuous surface, have a higher sampling density on surfaces near the sensor and over-parametrise simple surfaces. These redundant samples mean that depth images are not memory-efficient. Moreover, the representation is not robust to sensor noise or structured outliers and does not adapt to local surface complexity. In particular, structured light depth cameras are very susceptible to structured outliers caused by projecting the pattern onto reflective or transparent surfaces. For example, these cameras are often unable to distinguish objects from their reflections and reconstruct a Carrollian mirror world.

In this work, matrix notation is used to describe depth images. That is, the depth image $\mathbf{D}$ is an $m \times n$ matrix of mn pixels, where the value of each element is the depth measurement or a special value representing "no measurement".

3.3.3 Greyscale Images

A greyscale digital image is a structured 2D array of intensity values, where each pixel records the intensity measurement in that direction. If the intrinsic camera calibration parameters are known, each pixel can be converted to a bearing vector. Hence, images are directional sensor data representations. An example is shown in Figure 3.1(a).

Greyscale images can be captured using a digital camera, possibly with an additional processing step to convert colour information to intensity. For most cameras, the intensity values at every pixel are measured simultaneously.

The greyscale image data representation has many advantages. Firstly, it is a ubiquitous representation produced by an inexpensive sensor, with extremely large datasets available. It is also information-rich, with even a single image containing a large amount of information about scene geometry and appearance. The geometry cues from a single image can be very useful for alignment algorithms, in addition to the geometric information available from multiple images. As with depth images, the grid structure is algorithmically convenient and images are used pervasively as inputs to convolutional neural networks. However, the image representation has several disadvantages for alignment algorithms. Significantly, images from calibrated cameras contain only directional information, not positional information. Missing one dimension of spatial information provides an additional level of challenge for alignment algorithms since the

problem can be under-constrained and susceptible to ambiguities. For example, with depth unknown, geometric cues taken from images may be optical illusions. Moreover, visual images are a very complex modality, with imaged objects having varying appearances when viewed from different directions or with different illumination.

In this work, matrix notation is used to describe greyscale images. That is, the image $\mathbf{Y}$ is an $m \times n$ matrix of mn pixels, where the value of each element is the intensity measurement. For a linear RGB space, the formula to convert a colour image to a greyscale image $\mathbf{Y}$ while preserving luminance is

$$\mathbf{Y} = 0.2126\mathbf{R} + 0.7152\mathbf{G} + 0.0722\mathbf{B} \tag{3.29}$$

where $\mathbf{R}$, $\mathbf{G}$ and $\mathbf{B}$ are the red, green and blue channels of the image in matrix form.

3.3.4 Bearing Vector Sets

Now that the predominant raw sensor data representations have been discussed, the processed representations used in this work will be introduced. Bearing vector sets are generated from greyscale images captured by a calibrated camera using rudimentary processing and consist of a set of 3D vectors. To make the 2D directional nature of the data more explicit, bearing vectors are typically normalised to have unit length, reducing the degrees of freedom by one. Each data element must contain a vector directed towards the observed element but may also contain additional information, such as intensity, colour, labelling or a feature descriptor. While predominantly generated from images, they are not limited to structured data of that type.

An important aspect of the conversion from image pixels to bearing vectors is the selection of which pixels, known as keypoints or feature points, to include. Not all pixels contain geometric information useful for alignment. For example, while a skyline may be useful for alignment, the majority of sky pixels are unlikely to provide useful cues about the geometric structure of the scene. Therefore, the selection of relevant pixels is a challenge that must be addressed by this data representation. Some possibilities for pixel selection criteria include whether they are identified as visual edges or corners, which may be geometrically meaningful, or whether they have been labelled with a 'structural' class, such as 'building'.

Bearing vector sets have some advantages over images for the alignment problem. They are a natural representation for directional data and make it explicit that a calibrated camera is an angle measurement device. In addition, many alignment algorithms, such as most perspective-n-point algorithms, operate directly on bearing vectors. However, many disadvantages of images persist when they are converted to bearing vector sets. The lack of depth information leads to alignment ambiguities with

this data type and changes in appearance can affect the repeatability of pixel selection. Keypoint selection is non-trivial and can lead to significant difficulties with alignment.

In this work, set notation is used to describe bearing vector sets. That is, the set $\mathcal{F}$ containing the 3D bearing vectors $\mathbf{f}_1$, $\mathbf{f}_2$, …, $\mathbf{f}_N$ is given by

$$\mathcal{F} = \{\mathbf{f}_i\}_{i=1}^{N}. \tag{3.30}$$

The conversion of a 2D image point $\mathbf{x}$ to a bearing vector $\mathbf{f}$ is given by

$$\mathbf{f} \propto \mathbf{K}^{-1}\hat{\mathbf{x}} \tag{3.31}$$

where $\mathbf{K}$ is the matrix of intrinsic camera parameters and $\hat{\mathbf{x}}$ is the homogeneous image point $\hat{\mathbf{x}} = (\mathbf{x}, 1)^\intercal$. Bearing vectors are typically normalised to have unit length.

3.3.5 Gaussian Mixture Models

One processed sensor data representation that can be constructed from low-level positional representations, such as point-sets and depth images, is the Gaussian Mixture Model (GMM). This positional data representation can be used to model the continuous underlying surfaces of a scene from a discrete sampling of those surfaces. Specifically, a GMM models observations, such as 3D points, as being generated from a mixture of finitely many Gaussian distributions. They can admit arbitrarily accurate estimates of many noisy surface densities [Devroye, 1987, Theorem 2.2]. An example GMM is shown in Figure 3.1(d).

There are a plethora of ways in which a GMM can be generated from point-set data. These include Kernel Density Estimation (KDE) [Jian and Vemuri, 2011; Detry et al., 2009; Comaniciu, 2003; Xiong et al., 2013a], Expectation Maximisation (EM) [Dempster et al., 1977; Deselaers et al., 2010], Dirichlet Process (DP) estimation [Antoniak, 1974; Straub et al., 2017] or mixture-mapped Support Vector Machines (SVMs) [Campbell and Petersson, 2015]. With the exception of the last method, these are generative models that attempt to find the distribution from which the samples were generated. This is different to modelling the distribution of surfaces in the scene, since the sampled data also incorporates information about the sensor. That is, a naïve generative model will attempt to model the scene *as observed by the sensor*, including the sensor-specific artefacts such as discretisation and distance-dependent point density.

The most general, expressive and least constrained GMM has non-identical mixture weights, full and non-identical covariance matrices, and an adaptive number of Gaussian components. However, more restrictive conditions are often imposed to expedite GMM generation or simplify the expressions. These include:

- tied (identical) mixture weights;
- tied (identical) covariance matrices;
- constrained covariance matrices; and
- a fixed or restricted number of components.

Any or all of these restrictions can be selected. Two common constraints on the covariance matrices are the requirement that they be diagonal or scalar matrices. Off-diagonal elements are zero for both constraints, encoding the assumption that the variates are uncorrelated. Scalar matrices have the additional constraint that all diagonal elements be equal, that is, having the form $\sigma^2\mathbf{I}$. This constraint is often referred to as an isotropic or spherical covariance condition, since the surfaces of constant likelihood about the Gaussian mean value are spheres.

The Gaussian mixture model data representation has several advantages. Firstly, it is a probabilistic model that has been estimated from the data, providing a rich statistical representation that can be mathematically interrogated, such as with Bayesian inference. For example, the question 'how likely is this new observation?' has a precise probabilistic answer. Moreover, the model can incorporate sampling effects and measurement uncertainty, reflecting the inherent uncertainty in the real sensing process. Importantly, the GMM representation attempts to model the underlying surfaces of the scene, since these are the entities that are being measured. As previously noted, for the GMM to accurately model these surfaces, parameter estimation must include an understanding of the non-uniformity of the sampling.

Another advantage of the GMM representation is that it is continuous, like the physical surfaces that were measured. Significantly, the data association problem for continuous representations is implicit, not combinatorial like for discrete point-sets. In addition, the class of objective functions admitted for continuous data representations is different than those for discrete representations and tend to have a wider region of convergence. Moreover, the representation is very memory efficient, since many points may be represented by a single Gaussian density. For example, points densely sampled on a plane can be replaced entirely by a single Gaussian whose covariance matrix has two large eigenvalues and a third eigenvalue equal to zero. GMMs are also robust to Gaussian sensor noise, since this is modelled implicitly by the representation. They are also robust to structured outliers since the probability of a point being sampled at any location is non-zero and the outlier distribution, if it is known, can be added to the mixture. Finally, the representation can be very adaptive to local surface complexity and is inherently data-driven, typically imposing no structure on the recovered surface (see Magnusson et al. [2007] for a counter-example).

However, the GMM representation also has some disadvantages. Inferring a continuous surface from a discretely sampled one is an inherently challenging problem

and methods used to solve this by generating GMMs from sensor data are not equally successful and all involve trade-offs. In addition, the GMM representation is an abstraction that has its own set of algorithmic challenges. Directly operating with the raw sensor data can often be faster than interposing an additional data processing step. Finally, the representation does not explicitly model free or unobserved regions. Free regions include the line between a laser rangefinder and each sampled point, which is known to be 'empty' or transparent to the laser, such as air. As a result, information that could be used to reason about structured outliers is lost.

Let $\boldsymbol{\theta} = \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \phi_i\}_{i=1}^{n}$ be the parameter set of an n-component GMM generated from the point-set $\mathcal{P}$, with means $\boldsymbol{\mu}_i$, covariance matrices $\boldsymbol{\Sigma}_i$, and mixture weights $\phi_i \geqslant 0$, where $\sum_{i=1}^{n} \phi_i = 1$. The probability density function given these parameters is

$$p(\mathbf{p}|\boldsymbol{\theta}) = \sum_{i=1}^{n} \phi_i \mathcal{N}(\mathbf{p}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{3.32}$$

where

$$\mathcal{N}(\mathbf{p}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{p}-\boldsymbol{\mu})^{\intercal}\boldsymbol{\Sigma}^{-1}(\mathbf{p}-\boldsymbol{\mu})\right) \tag{3.33}$$

is the probability density function for a Gaussian random variable and $|\cdot|$ is the determinant. In a slight abuse of notation, $\mathcal{N}(\mathbf{p}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is used to denote the probability density function $p_X(\mathbf{p})$ of a Gaussian random variable X, whereas the notation $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is used to denote the normal distribution itself.

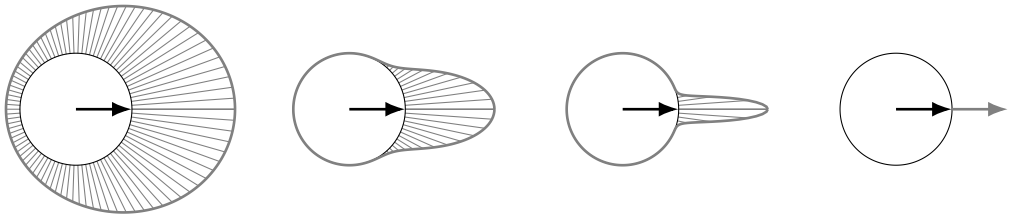
A useful identity [Petersen and Pedersen, 2012, §8.1.8] is given by

$$\int \mathcal{N}(\mathbf{p}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)\mathcal{N}(\mathbf{p}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)\,\mathrm{d}\mathbf{p} = \mathcal{N}(\mathbf{0}|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2). \tag{3.34}$$

where the left hand side is integrated over the entire domain of $\mathbf{p}$, that is $\mathbb{R}^3$. This identity is helpful in deriving closed-form equations for distance measures between Gaussian densities.

3.3.6 Mixture Models on the Sphere

By analogy to Gaussian mixture models in $\mathbb{R}^D$, mixture models may also be defined on the sphere S^{D-1}. Where GMMs can provide a probabilistic model of positional data such as 2D or 3D points, mixture models on the sphere can provide a probabilistic model of directional data, such as bearing vectors. In the field of directional statistics, many probability distributions on the sphere have been defined, several of which are relevant to the sensor data alignment problem: the von Mises–Fisher [Fisher, 1953], Fisher–Bingham [Kent, 1982] and Projected Normal [Mardia, 1972; Watson, 1983] distributions.



(a) $\kappa = 1$ (b) $\kappa = 10$ (c) $\kappa = 100$ (d) $\kappa \to \infty$

Figure 3.2: Visualisation of a von Mises–Fisher distribution as the concentration parameter κ increases. A 2D slice of the 3D distribution is shown for clarity. As $\kappa \to \infty$, the distribution approaches a delta function on the sphere.

The von Mises–Fisher (vMF) distribution, visualised in Figure 3.2, is closely related to the isotropic Gaussian distribution on the sphere S^{D-1} in $\mathbb{R}^D$ and is frequently used to describe directional data [Gopal and Yang, 2014; Straub et al., 2015]. In two dimensions, it is a close and tractable approximation to the wrapped normal distribution, the circular analogue of the normal distribution. There are several methods for generating a von Mises–Fisher Mixture Model (vMFMM) from a bearing vector set. These include k-means [Dhillon and Modha, 2001], Expectation Maximisation (EM) [Banerjee et al., 2005], and the posterior of a Dirichlet process [Straub et al., 2015]. The last approach is nonparametric and automatically infers the number of components.

Let $\boldsymbol{\theta} = \{\boldsymbol{\mu}_i, \kappa_i, \phi_i\}_{i=1}^n$ be the parameter set of an n-component von Mises–Fisher Mixture Model (vMFMM) generated from the bearing vector set $\mathcal{F}$, with mean directions $\boldsymbol{\mu}_i \in S^{D-1}$, concentrations $\kappa_i > 0$, and mixture weights $\phi_i \geqslant 0$, where $\sum_{i=1}^n \phi_i = 1$. The probability density function given these parameters is

$$p(\mathbf{f}|\boldsymbol{\theta}) = \sum_{i=1}^{n} \phi_i \, \mathrm{vMF}(\mathbf{f}|\boldsymbol{\mu}_i, \kappa_i) \tag{3.35}$$

where

$$\mathrm{vMF}(\mathbf{f}|\boldsymbol{\mu}, \kappa) = C_D(\kappa) \exp(\kappa \boldsymbol{\mu}^\mathsf{T} \mathbf{f}) \tag{3.36}$$

is the probability density function for a von Mises–Fisher random variable [Fisher, 1995] and $C_D(\kappa)$ is the normalisation constant for dimension D is given by

$$C_D(\kappa) = \frac{\kappa^{D/2-1}}{(2\pi)^{D/2} I_{D/2-1}(\kappa)} \tag{3.37}$$

where I_α denotes the modified Bessel function of the first kind of order α, defined as

$$I_\alpha(x) = \sum_{n=0}^{\infty} \frac{1}{n! \, \Gamma(n + \alpha + 1)} \left(\frac{x}{2}\right)^{2n+\alpha} \tag{3.38}$$

where Γ is the gamma function [Abramowitz and Stegun, 1964]. For $D = 3$, the normalisation constant simplifies to

$$C_3(\kappa) = \frac{\kappa}{4\pi \sinh \kappa} = \frac{\kappa}{2\pi(\exp(\kappa) - \exp(-\kappa))} \tag{3.39}$$

and the probability density function for the vMFMM simplifies to

$$p(\mathbf{f}|\boldsymbol{\theta}) = \sum_{i=1}^{n} \phi_i \frac{\kappa_i \exp(\kappa_i(\boldsymbol{\mu}_i^\intercal \mathbf{f} - 1))}{2\pi(1 - \exp(-2\kappa_i))}. \tag{3.40}$$

The Fisher–Bingham (FB) or Kent distribution [Kent, 1982] is closely related to the non-isotropic (bivariate) Gaussian distribution on the sphere S^{D-1} in $\mathbb{R}^D$. As with the vMF distribution, a Fisher–Bingham mixture model can be estimated from a bearing vector set. The probability density function for an FB random variable is

$$\mathrm{FB}(\mathbf{f}|\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{\gamma}_2, \kappa, \beta) = (C(\kappa, \beta))^{-1} \exp\Big(\kappa \boldsymbol{\gamma}_1^\intercal \mathbf{f} + \beta\Big(\big(\boldsymbol{\gamma}_2^\intercal \mathbf{f}\big)^2 - \big(\boldsymbol{\gamma}_3^\intercal \mathbf{f}\big)^2\Big)\Big) \tag{3.41}$$

where $\boldsymbol{\gamma}_1$ is the mean direction, $\boldsymbol{\gamma}_2$ and $\boldsymbol{\gamma}_3$ are the major and minor axes of the elliptical contours of equal probability, $\kappa > 0$ is the concentration or spread of the distribution and $\beta < \kappa/2$ controls the ellipticity. The normalising constant $C(\kappa, \beta)$ involves an infinite sum of modified Bessel functions of the first kind I_α and is given by

$$C(\kappa, \beta) = 2\pi \sum_{n=0}^{\infty} \frac{\Gamma\big(n + \frac{1}{2}\big)}{\Gamma(n+1)} \beta^{2n} \left(\frac{\kappa}{2}\right)^{-2n-\frac{1}{2}} I_{2n+\frac{1}{2}}(\kappa). \tag{3.42}$$

While more expressive than the vMF distribution, the parameters of the FB distribution are substantially more difficult to estimate because the FB distribution does not have a closed form.

The Projected Normal (PN) distribution [Mardia, 1972; Wang and Gelfand, 2013] is the projection of a non-isotropic Gaussian distribution in $\mathbb{R}^D$ onto the sphere S^{D-1}. Interestingly, the resulting distribution is not necessarily unimodal, depending on the length of the mean vector and the covariance matrix. This distribution is useful for modelling a 3D scene as observed by a 2D sensor, such as a camera. Unlike the previous distributions, a Projected Normal mixture model can be estimated from a point-set, not a bearing vector set. Consider a vector $\mathbf{p} \in \mathbb{R}^D$ that follows a multivariate normal distribution, that is $\mathbf{p} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, which has zero probability of being the zero vector, that is $p(\mathbf{p} = \mathbf{0}) = 0$. Then the bearing vector $\mathbf{f} = \mathbf{p}/\|\mathbf{p}\|$ follows a projected normal distribution $\mathrm{PN}_D(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, also known as the offset normal [Mardia, 1972] or angular Gaussian distribution [Watson, 1983; Pukkila and Rao, 1988]. The probability density function for a Projected Normal random variable, in a formulation derived by Pukkila

and Rao [1988], is given by

$$\mathrm{PN}(\mathbf{f}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = |2\pi\boldsymbol{\Sigma}|^{-1/2} Q_3^{-D/2} \exp\Big(-\frac{1}{2}\Big(Q_1 - Q_2^2 Q_3^{-1}\Big)\Big) I_D\Big(Q_2 Q_3^{-1/2}\Big) \tag{3·43}$$

where $Q_1 = \boldsymbol{\mu}^\intercal\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$, $Q_2 = \boldsymbol{\mu}^\intercal\boldsymbol{\Sigma}^{-1}\mathbf{f}$, $Q_3 = \mathbf{f}^\intercal\boldsymbol{\Sigma}^{-1}\mathbf{f}$, and the function I_D is given by

$$I_3(x) = \sqrt{2\pi}\Phi(x)\Big(1 + x^2\Big) + x\exp\Big(-\frac{1}{2}x^2\Big) \tag{3·44}$$

for $D = 3$, where $\Phi(x) = \Big(\sqrt{2\pi}\Big)^{-1}\int_{-\infty}^{x}\exp\Big(-\frac{t^2}{2}\Big)\,\mathrm{d}t$ is the cumulative distribution function of the standard normal distribution. A simpler form can be found for an isotropic Gaussian distribution in $\mathbb{R}^3$ with $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$, given by

$$\mathrm{PN}(\mathbf{f}|\boldsymbol{\mu},\sigma^2) = \frac{1}{(2\pi)^{\frac{3}{2}}}\exp\left(-\frac{\|\boldsymbol{\mu}\|^2}{2\sigma^2}\right)\left[\frac{\boldsymbol{\mu}^\intercal\mathbf{f}}{\sigma} + \sqrt{2\pi}\Phi\left(\frac{\boldsymbol{\mu}^\intercal\mathbf{f}}{\sigma}\right)\exp\left(\frac{1}{2}\left[\frac{\boldsymbol{\mu}^\intercal\mathbf{f}}{\sigma}\right]^2\right)\left(1 + \left[\frac{\boldsymbol{\mu}^\intercal\mathbf{f}}{\sigma}\right]^2\right)\right]. \tag{3·45}$$

3.4 Objective Functions for Sensor Data Alignment

The objective functions in this section consider 2D or 3D sensor data or both, depending on the suitability of the measure for each modality. For 3D, only positional (spatial) information is examined, such as a point-set of 3-vectors or a 3D Gaussian mixture model. For 2D, two types of sensor data are examined: positional information, such as a point-set of 2-vectors from a 2D laser scan or image pixels, and directional information, such as a set of bearing 3-vectors with unit norm. The latter may correspond to 2D points imaged by a calibrated camera, which is an angle-measuring device.

A good objective function will attain an optimum at the 'correct' alignment of the sensor data. However, this simply means that the true rigid transformation between the two sets of sensor data has been found. Consequently, the objective function should enable the discovery of a transformation that is as close to the ground truth transformation as possible, on all relevant datasets. Clearly this is not well-defined, particularly in the presence of sensor noise and outliers, making alignment a non-trivial problem. Therefore, a good objective function needs to be robust to noise and outliers in the data if it is to attain an optimum at the true alignment across many datasets. While random outliers can be problematic, the more pervasive and challenging type are structured outliers, such as those caused by occlusion or sampling. Finally, a good objective function will also be cheap to evaluate on a computer, with closed-form expressions and efficiently computable functions being advantageous. Differentiability is also beneficial for many optimisation algorithms, particularly when closed-form derivatives exist.

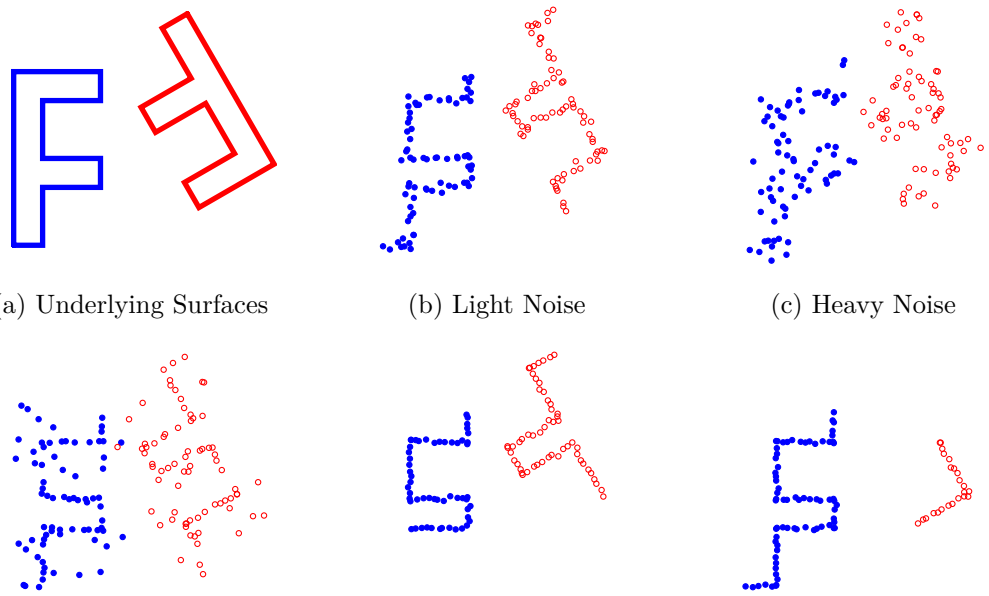
Some examples of challenging 2D point-set alignment problems are shown in Figure 3.3. The underlying surfaces from which the samples were drawn are shown in Figure 3.3(a). A common source of error for many objective functions is sensor noise. In Figure 3.3(b), the correct alignment is obvious to a human observer, but the Gaussian noise may cause an algorithm using a non-robust objective function to find an incorrect minimum. In Figure 3.3(c), the correct alignment is nearly impossible for a human or computer to distinguish. It is clear that there is a breakdown point for humans and computers, but a good objective function should tolerate a low signal-to-noise ratio. Another common source of error is outliers. In Figure 3.3(d), the random outliers may be problematic for a non-robust objective function but are trivial for a human to handle, even in large quantities. In Figure 3.3(e), the point-sets overlap incompletely (partial-to-partial alignment), inducing structured outliers. For lower amounts of overlap, the problem becomes increasingly challenging, for both humans and computers. Even when one point-set is a subset of the other (partial-to-full alignment), as shown in Figure 3.3(f), repeated structures and other symmetries can be confounding factors. A good objective function should mimic the human response of considering multiple transformations as equally likely. Finally, structured outliers can also arise from sampling itself, because data elements in one set may not have a corresponding element in the other set. This can pose a significant challenge for sparsely sampled surfaces.

It is apparent that selecting an appropriate objective function is one of the most crucial steps in sensor data alignment. Among the considerations referred to so far, robustness to noisy sensor data with many outliers is particularly critical. Therefore, one of the purposes of this section is to survey some of the most frequently used objective functions and to outline the advantages and disadvantages of each. Moreover, this section is restricted to objective functions that do not require correspondences. That is, the functions can all be used to solve the general geometric alignment problem. In addition, objective functions that are used or referred to in later chapters of this work will be treated in more detail than other functions.

3.4.1 Least Squares

Least squares objective functions consist of the sum of the squared residuals between elements of the data sets. While the least squares minimiser is not robust to outliers, it can be useful for refining an approximate alignment.

A frequently used nD–nD least squares objective function is the function minimised by the Iterative Closest Point (ICP) algorithm [Besl and McKay, 1992; Chen and Medioni, 1992]. The function uses closest-point residuals, that is, the Euclidean distance between a point in the transformed set and its closest-point in the other set.



(a) Underlying Surfaces (b) Light Noise (c) Heavy Noise

(d) Outliers: Random (e) Outliers: Partial-Overlap (f) Outliers: Subset

Figure 3.3: Examples of challenging 2D point-set alignment problems. The datasets sampled from the underlying surfaces (a) have different noise characteristics (b)–(c), random outliers (d) or structured outliers from partially-overlapping observations (e)–(f).

The ICP algorithm iteratively alternates between finding closest-point correspondences and finding the least squares transformation parameters given those correspondences. The second step can be solved in closed-form using the method of Horn [1987] or singular value decomposition [Arun et al., 1987]. Let $\mathbf{p} \in \mathbb{R}^n$ be an nD point and $\mathcal{P}_k = \{\mathbf{p}_{ki}\}_{i=1}^{N_k}$ be a set of N_k points. The non-symmetric ICP objective function is

$$f(\mathbf{R}, \mathbf{t}) = \sum_{\mathbf{p}_1 \in \mathcal{P}_1} \min_{\mathbf{p}_2 \in \mathcal{P}_2} d_{\mathrm{E}}^2(\mathbf{R}\mathbf{p}_1 + \mathbf{t}, \mathbf{p}_2) = \sum_{\mathbf{p}_1 \in \mathcal{P}_1} \min_{\mathbf{p}_2 \in \mathcal{P}_2} \|\mathbf{R}\mathbf{p}_1 + \mathbf{t} - \mathbf{p}_2\|_2^2 \tag{3.46}$$

for a rotation $\mathbf{R} \in SO(n)$ and a translation $\mathbf{t} \in \mathbb{R}^n$. The symmetric ICP objective function that treats both point-sets identically is given by

$$f(\mathbf{R}, \mathbf{t}) = \sum_{\mathbf{p}_1 \in \mathcal{P}_1} \min_{\mathbf{p}_2 \in \mathcal{P}_2} \|\mathbf{R}\mathbf{p}_1 + \mathbf{t} - \mathbf{p}_2\|_2^2 + \sum_{\mathbf{p}_2 \in \mathcal{P}_2} \min_{\mathbf{p}_1 \in \mathcal{P}_1} \|\mathbf{R}\mathbf{p}_1 + \mathbf{t} - \mathbf{p}_2\|_2^2. \tag{3.47}$$

These objective functions use point-to-point residuals [Besl and McKay, 1992], however other residuals such as point-to-plane [Chen and Medioni, 1992] have also been proposed. In addition, while the original ICP algorithm was a local optimisation method, requiring a good initial transformation estimate, the ICP objective function has also been used in a global optimisation framework [Yang et al., 2016].

Another least squares objective function used to globally solve the rotation-only alignment problem was given in Li and Hartley [2007]. The main observation was that if $N_1 = N_2$ and there is a one-to-one correspondence (bijection) between the elements of $\mathcal{P}_1$ and $\mathcal{P}_2$, then the geometric alignment problem can be treated as a joint transformation and correspondence problem. That is, if the correspondences are known, the least squares solution for the transformation parameters can be found optimally and in closed-form, and if the transformation is known, the correspondences can be found optimally by solving a linear assignment problem [Papadimitriou and Steiglitz, 1982]. The objective function is given by

$$f(\mathbf{R}, \mathbf{t}, \mathbf{P}) = \sum_{i=1}^{N_1} \|\mathbf{R}\mathbf{p}_{1i} + \mathbf{t} - \mathbf{p}_{2j_{\mathbf{P}_i}}\|_2^2 \tag{3.48}$$

where the index of $\mathbf{p}_2$ is

$$j_{\mathbf{P}_i} = \arg\max_j \mathbf{P}_{ij} \tag{3.49}$$

where $\mathbf{P}_{ij}$ is the Boolean value at the i^{th} row and j^{th} column of the permutation matrix $\mathbf{P} \in \mathbb{P}^n$. The permutation matrix enforces the one-to-one correspondence between data elements. However, the one-to-one assumption is incorrect for most geometric alignment problems, where elements in one set of data do not have correspondences in the other set and vice versa, due to outliers from occlusion, partial overlap or sampling. The ICP objective function does not assume a bijection between $\mathcal{P}_1$ and $\mathcal{P}_2$.

For 2D–3D alignment, a least squares objective function can be constructed analogously to the ICP criterion using the angular distance measure. Let $\mathbf{f} \in \mathbb{R}^n$ and $\mathbf{p} \in \mathbb{R}^n$ be an nD bearing vector and point respectively, $\mathcal{F} = \{\mathbf{f}_i\}_{i=1}^M$ be a set of M bearing vectors and $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^N$ be a set of N points. Then the non-symmetric 'angular ICP' objective function is given by

$$f(\mathbf{R}, \mathbf{t}) = \sum_{\mathbf{f} \in \mathcal{F}} \min_{\mathbf{p} \in \mathcal{P}} d_{\angle}^2(\mathbf{f}, \mathbf{R}\mathbf{p} + \mathbf{t}). \tag{3.50}$$

However, least squares is not frequently used for this problem, since the sum of the angular distances is easier to compute than the squares and is more robust to outliers.

3.4.2 Robust Least Squares Alternatives

While least squares objective functions are ideal for sensor data that has been corrupted by Gaussian noise, they are not robust to outliers. In particular, data elements in one set that do not have a corresponding element in the other set can bias the least squares parameter estimate significantly. Efforts to address this problem seek to reduce the

weight assigned to incorrect correspondences or prune them entirely. Many heuristic approaches have been used to identify incorrect correspondences, including reasoning about the Euclidean distance between correspondences or the angular distance between the normals of the correspondences [Rusinkiewicz and Levoy, 2001]. However, more rigorous approaches use robust statistics.

M-estimators are a general class of robust estimators that typically have a high breakdown point and good efficiency [Huber, 1981]. In addition, they do not necessarily relate to a probability density function and so can be used with raw, discrete sensor data. For geometric alignment, M-estimators are the parameters $\mathbf{R}^*$ and $\mathbf{t}^*$ that satisfy

$$\arg\min_{\mathbf{R},\mathbf{t}} \sum_{i=1}^{n} \rho(\epsilon_i(\mathbf{R},\mathbf{t})) \tag{3.51}$$

for a function ρ and a set of n residuals $\epsilon(\mathbf{R},\mathbf{t})$, being the Euclidean or angular distances between corresponding data elements.

The least squares estimator is itself an M-estimator with $\rho(\epsilon) = \epsilon^2$, however more robust estimators have been proposed. The ρ functions associated with more robust estimators include the L_p norms $\rho(\epsilon) = |\epsilon|^p$, the Huber or Winsorised loss function and Tukey's biweight function [Huber, 1981]. The key feature is that large (outlier) residuals are not penalised at the same rate as small residuals. The Huber loss function for example is quadratic for small residuals and linear for large residuals. The optimisation problem (3.51) can be solved for these ρ functions with the iteratively reweighted least squares algorithm. However, correspondences need to be known or estimated for this formulation. As with the ICP objective function, closest-point correspondences are typically used to find the residuals.

One disadvantage of many M-estimators, including those that use the Winsorised or biweight functions, is that they require a threshold to be set. This defines at what size a residual should be considered an outlier, which may not be known in advance. In contrast, sparsity-inducing functions such as L_p norms for $p \in (0, 1]$ do not require a threshold, reward inliers and only weakly penalise outliers. As such, their M-estimators are related to the estimators found by inlier set cardinality maximisation.

Two robust alternatives that can be used for raw, discrete sensor data will be discussed in more detail in the following sections: least trimmed squares and inlier set cardinality. The least trimmed squares estimator can be formulated as an M-estimator with an adaptive threshold value from analysing the distribution of residuals. In contrast, the inlier set cardinality estimator is not typically analysed as an M-estimator. After this, robust objective functions for processed sensor data in the form of probability density functions will be introduced. These objective functions downweigh observations probabilistically, not geometrically like the aforementioned functions.

3.4.3 Least Trimmed Squares

Least trimmed squares objective functions consist of the sum of the K smallest squared residuals between elements of the data sets. Trimming is a technique used in statistics to obtain a more robust statistic by excluding outlier values. However, it is predicated upon knowing the number of data elements K from one set that have a corresponding element in the other set, which is non-trivial in practise.

The trimmed ICP objective function minimised in Chetverikov et al. [2005] uses a subset $\mathcal{T}$ of cardinality K of the closest-point residuals at each iteration for the transformation calculation. Let $\mathbf{p} \in \mathbb{R}^n$ be an nD point, $\mathcal{P}_k = \{\mathbf{p}_{ki}\}_{i=1}^{N_k}$ be a set of N_k points, and $\mathcal{T} \subseteq \mathcal{P}_1$ be the subset of $\min\{K, |\mathcal{P}_1|\}$ points in $\mathcal{P}_1$ with the smallest closest-point residuals for a given rotation and translation. Then the non-symmetric trimmed ICP objective function is given by

$$f(\mathbf{R}, \mathbf{t}) = \sum_{\mathbf{p}_1 \in \mathcal{T}} \min_{\mathbf{p}_2 \in \mathcal{P}_2} d_{\mathrm{E}}^2(\mathbf{R}\mathbf{p}_1 + \mathbf{t}, \mathbf{p}_2) = \sum_{\mathbf{p}_1 \in \mathcal{T}} \min_{\mathbf{p}_2 \in \mathcal{P}_2} \|\mathbf{R}\mathbf{p}_1 + \mathbf{t} - \mathbf{p}_2\|_2^2 \tag{3.52}$$

for a rotation $\mathbf{R} \in SO(n)$ and a translation $\mathbf{t} \in \mathbb{R}^n$. This objective function has also been used in a global optimisation framework [Yang et al., 2016].

For 2D–3D alignment, a least trimmed squares objective function can be constructed analogously. Let $\mathbf{f} \in \mathbb{R}^n$ and $\mathbf{p} \in \mathbb{R}^n$ be an nD bearing vector and point respectively, $\mathcal{F} = \{\mathbf{f}_i\}_{i=1}^{M}$ be a set of M bearing vectors, $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^{N}$ be a set of N points, and $\mathcal{T} \subseteq \mathcal{F}$ be the subset of $\min\{K, |\mathcal{F}|\}$ bearing vectors in $\mathcal{F}$ with the smallest closest ray residuals for a given rotation and translation. Then the non-symmetric trimmed 'angular ICP' objective function is given by

$$f(\mathbf{R}, \mathbf{t}) = \sum_{\mathbf{f} \in \mathcal{T}} \min_{\mathbf{p} \in \mathcal{P}} d_{\angle}^2(\mathbf{f}, \mathbf{R}\mathbf{p} + \mathbf{t}). \tag{3.53}$$

A similar objective function, which does not square the angular residuals, was used in a global optimisation framework by Brown et al. [2015].

While trimming improves the robustness of a function to outliers, it also requires the user to specify the inlier fraction, which can rarely be known in advance. It is also less intuitive to select than other geometrically meaningful thresholds. However, the main problem with this approach is that if the inlier fraction is over- or under-estimated, the global optimum of the function may not occur at the correct pose. Figure 3.4 demonstrates how a global optima of a trimmed objective function, as used by Brown et al. [2015] and Yang et al. [2016], may not occur at the true pose, a problem that is exacerbated when the inlier fraction is guessed incorrectly.

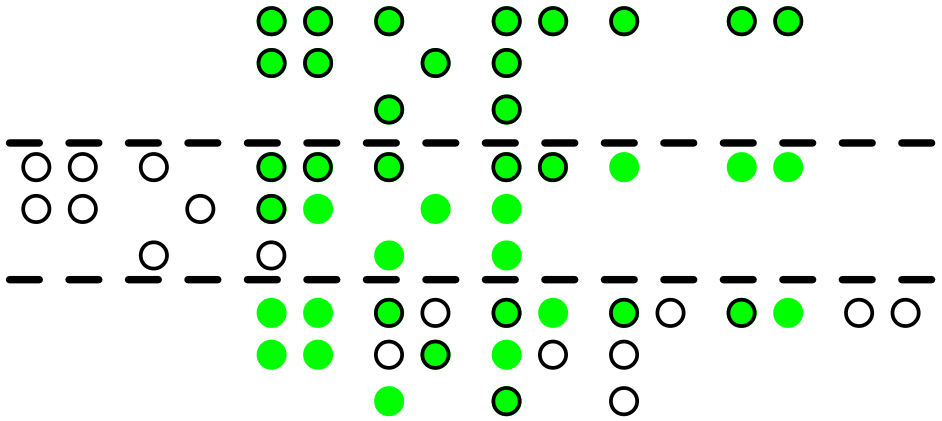


Figure 3.4: Three zero-error 1D alignments of 2 point-sets with 8 trimmed 'outliers'. For the correct transformation (top), the point-sets overlap completely and the trimmed objective function attains the global minimum (zero). For the two incorrect transformations (middle and bottom), the trimmed objective function also attains the global minimum but the alignment is incorrect. With noise, the global optimum of a trimmed objective function may not even occur at the true pose, particularly if an incorrect trimming fraction is selected. The problem is exacerbated with higher dimensions and degrees of freedom.

3.4.4 Inlier Set Cardinality

The cardinality of the inlier set is a robust objective function that is frequently maximised in geometric sensor data alignment problems [Breuel, 2003; Aiger et al., 2008; Parra Bustos et al., 2016]. It can be viewed as a discrete function that counts the number of inliers given a specific rigid transformation. For this objective function, an *inlier* must be defined precisely, using a hard threshold θ to decide whether a data element is a member of the inlier or outlier sets. The inlier sets considered here are inherently asymmetric, containing only elements from one of the datasets. However, a symmetric objective function can also be constructed if needed by evaluating the inlier set cardinality in both directions and selecting the minimum. Also, while inlier set cardinality objective functions can be defined over continuous data representations, they are typically used for discrete data such as point-sets and bearing vector sets.

The primary advantage of inlier set cardinality is that it is inherently robust to outliers, while still operating on raw data representations. It also avoids the problems associated with trimming and robust loss functions, which require the user to specify a parameter such as the inlier fraction, which can rarely be known in advance. Moreover, these parameters are often less intuitive to select than a geometrically meaningful inlier threshold, such as the maximum allowable spatial or angular error.

Let $\mathbf{x} \in \mathbb{R}^n$ be an nD point or bearing vector and $\mathcal{X}_k = \{\mathbf{x}_{ki}\}_{i=1}^{N_k}$ be a set of N_k points or bearing vectors. Then the inlier set cardinality for a rotation $\mathbf{R} \in SO(n)$ and

a translation $\mathbf{t} \in \mathbb{R}^n$ with an inlier threshold $\theta \geqslant 0$ is given by

$$\nu(\mathbf{R}, \mathbf{t}, \theta) = |\mathcal{S}_I| \tag{3·54}$$

$$\mathcal{S}_I = \{\mathbf{x}_1 \in \mathcal{X}_1 \mid \exists \mathbf{x}_2 \in \mathcal{X}_2 : d(\mathbf{x}_1, \mathbf{R}\mathbf{x}_2 + \mathbf{t}) \leqslant \theta\} \tag{3·55}$$

where $|\cdot|$ denotes the set cardinality and $d(\cdot, \cdot)$ denotes an arbitrary distance measure between vectors. An equivalent formulation is given by

$$\nu(\mathbf{R}, \mathbf{t}, \theta) = \sum_{\mathbf{x}_1 \in \mathcal{X}_1} \max_{\mathbf{x}_2 \in \mathcal{X}_2} \mathbf{1}(\theta - d(\mathbf{x}_1, \mathbf{R}\mathbf{x}_2 + \mathbf{t})) \tag{3·56}$$

where $\mathbf{1}(x) \triangleq \mathbf{1}_{\mathbb{R}_{\geq 0}}(x)$ is the indicator function that has the value 1 for all elements of the non-negative real numbers and the value 0 otherwise, given by

$$\mathbf{1}(x) = \begin{cases} 1 & \text{if } x \in \mathbb{R}_{\geq 0} \\ 0 & \text{else.} \end{cases} \tag{3·57}$$

This objective function, which is to be maximised, is a highly non-concave function. Equivalently, with more familiar terminology, the additive inverse of the inlier set cardinality is a highly non-convex objective function. Consequently, the function has many local optima, making it difficult to optimise effectively. In addition, the objective function makes the asymmetry of the inlier set cardinality explicit, since switching $\mathcal{X}_1$ and $\mathcal{X}_2$ may lead to a different solution. The inlier set cardinality in the other direction is given by $\nu'(\mathbf{R}, \mathbf{t}, \theta) = |\mathcal{S}'_I|$ for the inlier set

$$\mathcal{S}'_I = \{\mathbf{x}_2 \in \mathcal{X}_2 \mid \exists \mathbf{x}_1 \in \mathcal{X}_1 : d(\mathbf{x}_1, \mathbf{R}\mathbf{x}_2 + \mathbf{t}) \leqslant \theta\} \tag{3·58}$$

The symmetric inlier set cardinality objective function for a rotation $\mathbf{R} \in SO(n)$ and a translation $\mathbf{t} \in \mathbb{R}^n$ with an inlier threshold θ is therefore given by

$$\nu^{\text{sym}}(\mathbf{R}, \mathbf{t}, \theta) = \min\{\nu(\mathbf{R}, \mathbf{t}, \theta), \nu'(\mathbf{R}, \mathbf{t}, \theta)\}. \tag{3·59}$$

The Largest Common Point-set (LCP) between point-sets $\mathcal{P}_1$ and $\mathcal{P}_2$ is a related concept [Akutsu et al., 1998], defined as the subset $\mathcal{P}'_1 \subseteq \mathcal{P}_1$ with the largest possible cardinality such that the distance between $\mathcal{P}'_1$ and $T(\mathcal{P}'_2, \mathbf{R}, \mathbf{t})$ is less than θ, where $\mathcal{P}'_2 \subseteq \mathcal{P}_2$ and T is a transformation function. LCP under the Hausdorff distance [Chew et al., 1997], which is the maximum distance between a point and its nearest neighbour in the other set, is very similar to the inlier set cardinality function. In contrast, LCP under the bottleneck matching metric [Efrat et al., 2001], which seeks a bijection between the subsets, is more restrictive than the inlier set cardinality.

When correspondences are provided, another related objective function can be constructed, sometimes referred to as the consensus set cardinality [Li, 2009]. Let $\mathbf{x} \in \mathbb{R}^n$ be an nD point or bearing vector and $\mathcal{X}_k = \{\mathbf{x}_{ki}\}_{i=1}^N$ be a set of N points or bearing vectors. Then the consensus set cardinality for two ordered equally-sized datasets $\mathcal{X}_1$ and $\mathcal{X}_2$ with putative correspondences $\mathbf{x}_{1i} \leftrightarrow \mathbf{x}_{2i}$, for a rotation $\mathbf{R} \in SO(n)$ and a translation $\mathbf{t} \in \mathbb{R}^n$ with an inlier threshold $\theta \geqslant 0$ is given by

$$\nu_C(\mathbf{R}, \mathbf{t}, \theta) = |\mathcal{I}| \tag{3.60}$$

$$\mathcal{I} = \{i \in [1, N] \mid d(\mathbf{x}_{1i}, \mathbf{R}\mathbf{x}_{2i} + \mathbf{t}) \leqslant \theta\} \tag{3.61}$$

An equivalent formulation is given by

$$\nu_C(\mathbf{R}, \mathbf{t}, \theta) = \sum_{i=1}^{N} \mathbf{1}(\theta - d(\mathbf{x}_{1i}, \mathbf{R}\mathbf{x}_{2i} + \mathbf{t})). \tag{3.62}$$

Note the subtle but crucial difference between the consensus set and inlier set cardinality objective functions. The inlier set cardinality is a much more challenging objective function to optimise, because the transformation and correspondence set must both be solved jointly. The original RANSAC algorithm [Fischler and Bolles, 1981] was designed to maximise the consensus set cardinality objective function, albeit in a non-deterministic and heuristic way. Globally-optimal methods, such as Li [2009], have also been proposed.

Specific formulations of the inlier set cardinality function can also be written for positional or directional sensor data by selecting a distance measure. For nD–nD positional sensor data alignment, the inlier set consists of those points in $\mathcal{P}_1$ that are within θ of any point in $\mathcal{P}_2$ with respect to the Euclidean distance metric. Let $\mathbf{p} \in \mathbb{R}^n$ be an nD point and $\mathcal{P}_k = \{\mathbf{p}_{ki}\}_{i=1}^{N_k}$ be a set of N_k points. Then the inlier set cardinality for a rotation $\mathbf{R} \in SO(n)$ and a translation $\mathbf{t} \in \mathbb{R}^n$ with an inlier threshold $\theta \geqslant 0$ is given by

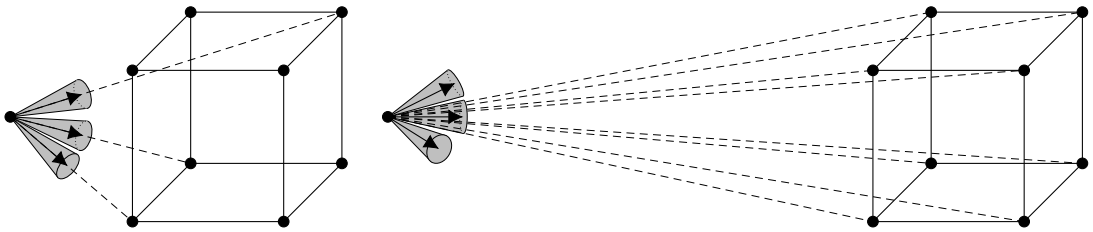
$$\nu(\mathbf{R}, \mathbf{t}, \theta) = |\mathcal{S}_I| \tag{3.63}$$

$$\mathcal{S}_I = \{\mathbf{p}_1 \in \mathcal{P}_1 \mid \exists \mathbf{p}_2 \in \mathcal{P}_2 : \|\mathbf{p}_1 - \mathbf{R}\mathbf{p}_2 - \mathbf{t}\| \leqslant \theta\} \tag{3.64}$$

or equivalently

$$\nu(\mathbf{R}, \mathbf{t}, \theta) = \sum_{\mathbf{p}_1 \in \mathcal{P}_1} \max_{\mathbf{p}_2 \in \mathcal{P}_2} \mathbf{1}(\theta - \|\mathbf{p}_1 - \mathbf{R}\mathbf{p}_2 - \mathbf{t}\|). \tag{3.65}$$

For 2D–3D directional sensor data alignment, the inlier set consists of those bearing vectors in $\mathcal{F}$ that are within θ of any point in $\mathcal{P}$ with respect to the angular distance metric. Given a set of points $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^M$ and bearing vectors $\mathcal{F} = \{\mathbf{f}_i\}_{i=1}^N$ and an inlier threshold θ, the inlier set cardinality for a rotation $\mathbf{R} \in SO(3)$ and a translation



(a) Non-degenerate case (b) Degenerate case

Figure 3.5: An example of the degenerate case for 2D–3D directional sensor data alignment when the cardinality of the set of 3D point inliers is maximised instead of the set of bearing vector inliers. A vector that lies within any of the cones is within the inlier threshold of the nearest bearing vector. (a) The pose that maximises the cardinality of the set of bearing vector inliers (3) is the correct camera pose. (b) The pose that maximises the cardinality of the set of 3D point inliers (8) is incorrect. This is a degenerate case where all 3D points become inliers when the translation of the camera is sufficiently far from the 3D points. In this example, the cube vertices are all within the angular inlier threshold of one bearing vector.

$\mathbf{t} \in \mathbb{R}^3$ is given by

$$\nu(\mathbf{R}, \mathbf{t}, \theta) = |\mathcal{S}_I| \tag{3.66}$$

$$\mathcal{S}_I = \{\mathbf{f} \in \mathcal{F} \mid \exists \mathbf{p} \in \mathcal{P} : \angle(\mathbf{f}, \mathbf{R}(\mathbf{p} - \mathbf{t})) \leqslant \theta\} \tag{3.67}$$

or equivalently

$$\nu(\mathbf{R}, \mathbf{t}, \theta) = \sum_{\mathbf{f} \in \mathcal{F}} \max_{\mathbf{p} \in \mathcal{P}} \mathbf{1}(\theta - \angle(\mathbf{f}, \mathbf{R}(\mathbf{p} - \mathbf{t}))) \tag{3.68}$$

where $\angle(\cdot, \cdot)$ denotes the angular distance between vectors. It is important to observe that it is the cardinality of the set of bearing vector inliers that is measured, not the cardinality of the set of 3D point inliers. Maximising the latter measure results in a set of degenerate poses where all 3D points are inliers with respect to a single bearing vector. These degenerate poses position the camera far from the point-set such that all points fall within the inlier cone of one bearing vector, as shown in Figure 3.5.

3.4.5 Probability Distribution Divergences

There are many statistical divergences and distances that can be used to align sets of sensor data when they are represented as probability distributions, such as Gaussian and von Mises–Fisher mixture models. Statistical divergences are non-negative and attain zero when the distributions are identical, however, unlike distances, they need not be symmetric or satisfy the triangle inequality. Usefully, many of these divergences weight observations probabilistically, not geometrically, providing a principled way to handle outliers. In this section, some divergence measures will be briefly outlined, focusing on those that have been used as alignment objective functions.

A large family of divergences was identified by Jones et al. [2001], and is given by

$$\frac{1}{\phi}\left(\int p_1^{1+\alpha}(x)\,\mathrm{d}x\right)^{\phi} - \frac{1+\alpha}{\alpha\phi}\left(\int p_1^{\alpha}(x)p_2(x)\,\mathrm{d}x\right)^{\phi} + \frac{1}{\alpha\phi}\left(\int p_2^{1+\alpha}(x)\,\mathrm{d}x\right)^{\phi} \tag{3.69}$$

for probability density functions p_1 and p_2. When $\phi = 1$, (3.69) becomes the density power divergence [Basu et al., 1998], a subset of the Bregman divergence functions [Bregman, 1967]. When $\phi \to 0$, (3.69) becomes the Windham divergence [Windham, 1995]. The estimators associated with these two cases have been shown to be exactly unbiased [Jones et al., 2001] and M-estimators [Hampel et al., 1986; Stewart, 1999]. Both cases have similar performance, although the density power divergence has a better asymptotic efficiency and breakdown point [Jones et al., 2001]. Furthermore, when $\alpha = 1$, the density power divergence becomes the L_2 distance between densities and the Windham divergence becomes the correlation between densities [Scott and Szewczyk, 2001]. These are both robust measures and have been used as closed-form alignment objective functions [Jian and Vemuri, 2011; Tsin and Kanade, 2004; Sandhu et al., 2010]. When $\alpha \to 0$, both divergences become the asymmetric Kullback–Leibler (KL) divergence [Kullback and Leibler, 1951], minimised by the Maximum Likelihood Estimator (MLE). Since the MLE is not robust to outliers, algorithms that used objective functions based on the KL divergence [Chui and Rangarajan, 2000a,b; Magnusson et al., 2007; Myronenko and Song, 2010] include an additional Gaussian component to account for outliers. Furthermore, Jian and Vemuri [2011] showed that the ICP algorithm could be interpreted as minimising the approximated KL divergence between mixtures, accounting for its sensitivity to outliers.

Alternative probability distribution divergences include the Rényi [Van Erven and Harremos, 2014], Jensen–Shannon [Lin, 1991] and Jensen–Rényi [Hamza and Krim, 2003] divergences. These divergences generalise the KL divergence and have been used for sensor data alignment [Wang et al., 2008, 2009]. The Jensen–Rényi divergence is also symmetric and has a closed-form expression, which are useful properties for an alignment objective function. The L_2 distance shares these properties and its application to sensor data alignment will be considered in the next two sections.

3.4.6 L_2 Distance between Gaussian Mixtures

As previously discussed, there are many statistical measures that can be used to align volumetric or probabilistic sensor data signals. However, the L_2 distance between probability distributions, a density power divergence [Basu et al., 1998], has many favourable properties for the geometric alignment problem [Jian and Vemuri, 2011]. In particular, the L_2 distance between Gaussian mixtures is a robust objective function

[Scott, 2001] that can be expressed in closed-form and efficiently implemented. These properties will be addressed in more detail later in this section.

The advantages of representing 2D or 3D positional sensor data as Gaussian Mixture Models (GMMs) were discussed in Section 3.3.5. Two key benefits are that Gaussian mixtures can admit arbitrarily accurate estimates of many noisy surface densities [Devroye, 1987] and can be computed efficiently from point-set data. However, GMMs are not suitable for modelling directional data such as bearing vectors and therefore cannot be used for certain alignment problems.

Let $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_{ki}, \boldsymbol{\Sigma}_{ki}, \phi_{ki}\}_{i=1}^{n_k}$ be the parameter set of an arbitrary n_k-component GMM with means $\boldsymbol{\mu}_{ki}$, covariance matrices $\boldsymbol{\Sigma}_{ki}$, and mixture weights $\phi_{ki} \geqslant 0$, where $\sum_{i=1}^{n_k} \phi_{ki} = 1$. Then the L_2 distance between Gaussian mixtures for a rotation $\mathbf{R} \in SO(n)$ and a translation $\mathbf{t} \in \mathbb{R}^n$ is given by

$$\begin{aligned} f(\mathbf{R}, \mathbf{t}) &= \int_{\mathbb{R}^n} \left[p(\mathbf{p}|T(\boldsymbol{\theta}_1, \mathbf{R}, \mathbf{t})) - p(\mathbf{p}|\boldsymbol{\theta}_2)\right]^2 \mathrm{d}\mathbf{p} \\ &= \int_{\mathbb{R}^n} \left[p(\mathbf{p}|T(\boldsymbol{\theta}_1, \mathbf{R}, \mathbf{t}))\right]^2 - 2p(\mathbf{p}|T(\boldsymbol{\theta}_1, \mathbf{R}, \mathbf{t}))p(\mathbf{p}|\boldsymbol{\theta}_2) + \left[p(\mathbf{p}|\boldsymbol{\theta}_2)\right]^2 \mathrm{d}\mathbf{p} \end{aligned} \tag{3.70}$$

where $p(\mathbf{p}|\boldsymbol{\theta})$ is the Gaussian mixture probability density function (3.32) and T is the function defined by

$$\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \phi_i\}_{i=1}^{n} \mapsto \{\mathbf{R}\boldsymbol{\mu}_i + \mathbf{t}, \mathbf{R}\boldsymbol{\Sigma}_i\mathbf{R}^\mathsf{T}, \phi_i\}_{i=1}^{n} \tag{3.71}$$

that maps a Gaussian mixture parameter set to another parameter set representing a rigid transformation of the original GMM. A closed-form objective function can be found using the following observations. Firstly, the first term of (3.70) is invariant under rigid transformations and the last term is independent of the transformation. Therefore the transformation that optimises a function without these terms will also optimise the L_2 distance function and thus the terms can be dropped. Secondly, the middle term is the inner product of two Gaussian mixtures and has a closed form. This can be seen by substituting (3.32) into (3.70), giving

$$\begin{aligned} &\int_{\mathbb{R}^n} p(\mathbf{p}|T(\boldsymbol{\theta}_1, \mathbf{R}, \mathbf{t}))p(\mathbf{p}|\boldsymbol{\theta}_2)\,\mathrm{d}\mathbf{p} \\ &\quad= \int_{\mathbb{R}^n} \sum_{i=1}^{n_1} \phi_{1i}\mathcal{N}(\mathbf{p}|\mathbf{R}\boldsymbol{\mu}_{1i} + \mathbf{t}, \mathbf{R}\boldsymbol{\Sigma}_{1i}\mathbf{R}^\mathsf{T}) \sum_{j=1}^{n_2} \phi_{2j}\mathcal{N}\Big(\mathbf{p}\Big|\boldsymbol{\mu}_{2j}, \boldsymbol{\Sigma}_{2j}\Big)\,\mathrm{d}\mathbf{p} && (3.72) \\ &\quad= \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \phi_{1i}\phi_{2j} \int_{\mathbb{R}^n} \mathcal{N}(\mathbf{p}|\mathbf{R}\boldsymbol{\mu}_{1i} + \mathbf{t}, \mathbf{R}\boldsymbol{\Sigma}_{1i}\mathbf{R}^\mathsf{T})\mathcal{N}\Big(\mathbf{p}\Big|\boldsymbol{\mu}_{2j}, \boldsymbol{\Sigma}_{2j}\Big)\,\mathrm{d}\mathbf{p} && (3.73) \\ &\quad= \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \phi_{1i}\phi_{2j}\mathcal{N}\Big(\mathbf{0}\Big|\mathbf{R}\boldsymbol{\mu}_{1i} + \mathbf{t} - \boldsymbol{\mu}_{2j}, \mathbf{R}\boldsymbol{\Sigma}_{1i}\mathbf{R}^\mathsf{T} + \boldsymbol{\Sigma}_{2j}\Big) && (3.74) \end{aligned}$$

where (3.72) is a consequence of identity (3.34). Finally, by substituting (3.33) and (3.74) into (3.70) and removing the invariant or independent terms, the objective function to optimise the L_2 distance between Gaussian mixtures is given by

$$f(\mathbf{R},\mathbf{t}) = -\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\phi_{1i}\phi_{2j}\frac{\exp\left(-\frac{1}{2}(\mathbf{R}\boldsymbol{\mu}_{1i}+\mathbf{t}-\boldsymbol{\mu}_{2j})^{\intercal}(\mathbf{R}\boldsymbol{\Sigma}_{1i}\mathbf{R}^{\intercal}+\boldsymbol{\Sigma}_{2j})^{-1}(\mathbf{R}\boldsymbol{\mu}_{1i}+\mathbf{t}-\boldsymbol{\mu}_{2j})\right)}{\sqrt{|2\pi(\mathbf{R}\boldsymbol{\Sigma}_{1i}\mathbf{R}^{\intercal}+\boldsymbol{\Sigma}_{2j})|}} \tag{3.75}$$

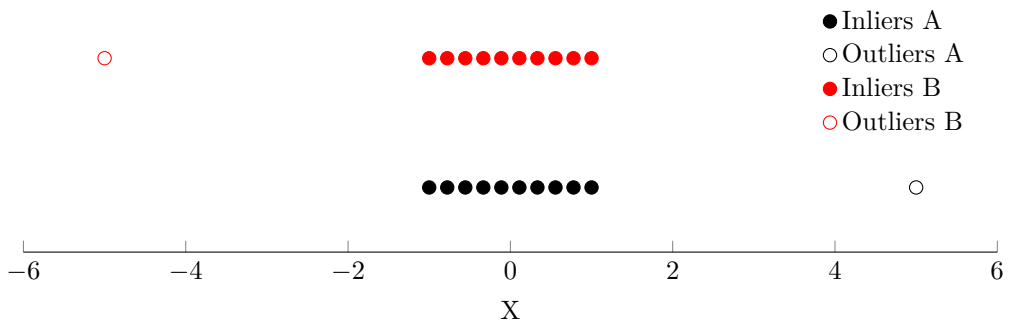
where $|\cdot|$ is the matrix determinant. The partial derivatives or gradient of this function with respect to the transformation parameters can also be found in closed-form.

Other than having a computationally-efficient closed form and thereby avoiding numerical integration, a primary advantage of this objective function is its robustness to outliers. This arises from the "inherently robust" L_2E estimator that minimises the L_2 distance between densities without requiring any tuning factors, unlike many other robust functions [Scott, 2001]. The robustness of the estimator to outliers has been demonstrated both empirically and from its connection with M-estimators [Basu et al., 1998; Jones et al., 2001; Scott, 2001]. While counter-intuitive, it arises from the Gaussian attenuation of outlying values. Basu et al. [1998] remarked that the function downweights observations probabilistically not geometrically, such that an observation with a lower probability of occurrence under the model is given a smaller weight, regardless of how far the observation is from other observations. The L_2 distance between densities was also shown to be a special case ($\alpha = 1$) of the density power divergence [Basu et al., 1998]. The density power divergence for probability density functions p_1 and p_2 is given by

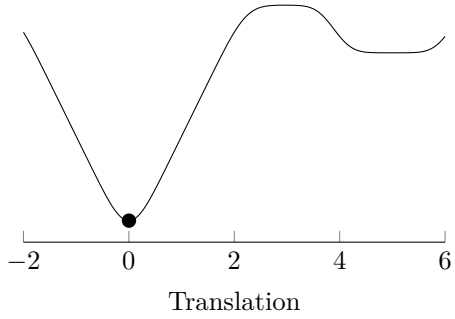
$$\int\left\{p_1^{1+\alpha}(x) - \frac{1+\alpha}{\alpha}p_1^{\alpha}(x)p_2(x) + \frac{1}{\alpha}p_2^{1+\alpha}(x)\right\}\mathrm{d}x. \tag{3.76}$$

Jones et al. [2001] compared the class of density power divergences with the class of Windham divergences [Windham, 1995], both of which have been shown to be M-estimators [Hampel et al., 1986; Stewart, 1999], and concluded that the classes perform similarly but the density power divergence has a better asymptotic efficiency and breakdown point. It is also notable that α, which parametrises the density power divergence, provides a continuous bridge between the Kullback-Leibler (KL) divergence ($\alpha \to 0$) and the L_2 distance ($\alpha = 1$). The KL divergence is minimised by the Maximum Likelihood Estimator (MLE) and has better asymptotic efficiency than the L_2E estimator but is less robust [Basu et al., 1998]. The parameter α can be used to control the trade-off between efficiency and robustness, but only has a closed form for $\alpha = 1$.

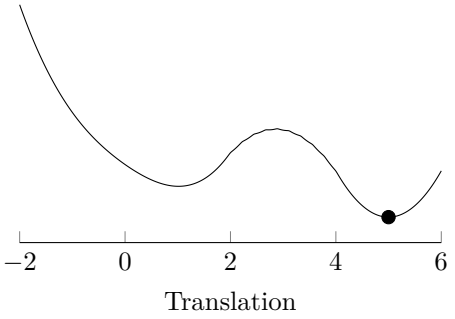
A toy example demonstrating that the L_2E estimator is not biased by systematic outliers, unlike the Maximum Likelihood Estimator (MLE), is shown in Figure 3.6. In



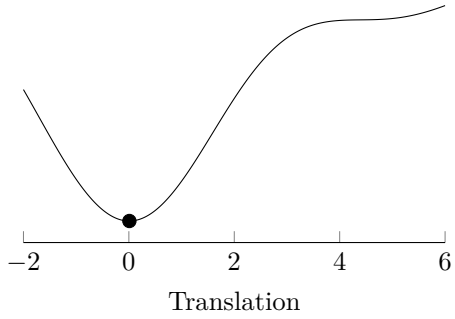
(a) Two 1D point-sets, both with a single outlier



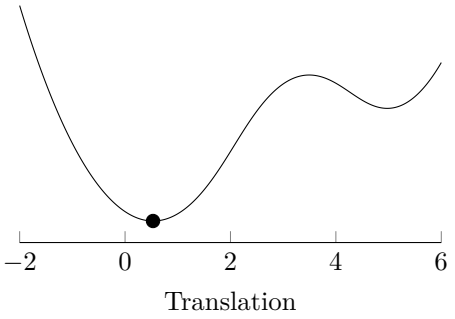
(b) L_2 Distance and L_2E ($\sigma = 0.2$)



(c) Negative Log-Likelihood and MLE ($\sigma = 0.2$)



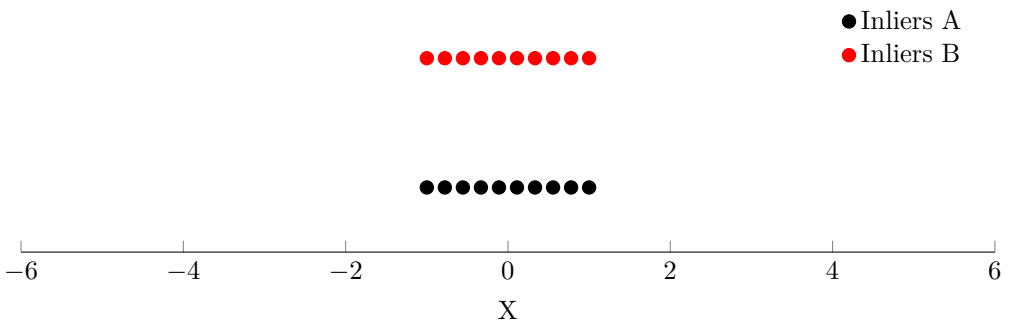
(d) L_2 Distance and L_2E ($\sigma = 1.0$)



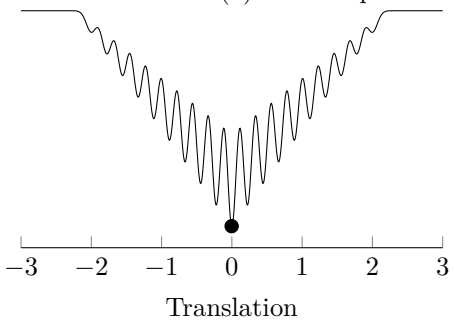
(e) Negative Log-Likelihood and MLE ($\sigma = 1.0$)

Figure 3.6: Toy example demonstrating the robustness of the L_2E estimator. (a) Two 1D point-sets A and B which overlap exactly, except for a single outlier in each. As point-set B translates with respect to point-set A, the L_2 distance between Gaussian mixtures (constructed from the point-sets using kernel density estimation) and the negative log-likelihood is evaluated and plotted for different scales σ. (b) At a scale of $\sigma = 0.2$, the L_2E estimator is globally-optimal and multiple local minima exist. (c) At the same scale (and below), the Maximum Likelihood Estimator (MLE) is severely biased by the outliers and finds the incorrect translation. It also has multiple local minima. (d)–(e) At larger scales (such as $\sigma = 1.0$), the MLE is still biased, but less so. As the scale increases further, both estimates converge towards aligning the centres-of-mass of the point-sets. The estimators (L_2E and MLE) that minimise the L_2 distance and negative log-likelihood are marked as black dots.

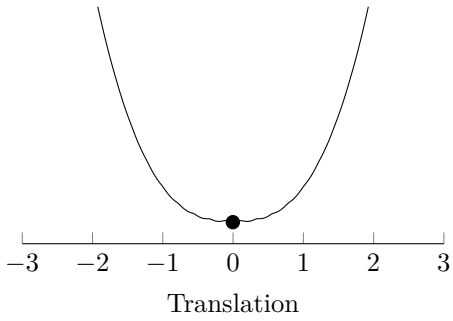
contrast, Figure 3.7 shows that, in the absence of outliers, the MLE is more efficient for the alignment task.



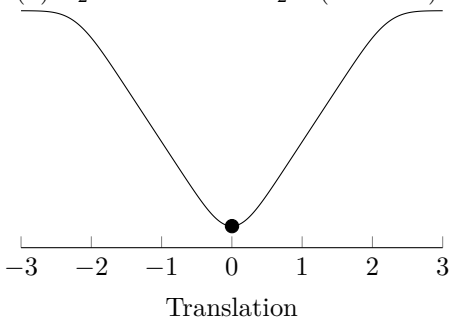
(a) Two 1D point-sets, both without any outliers



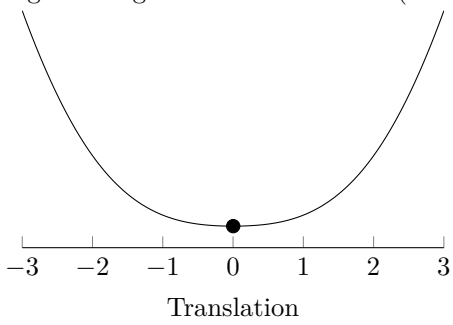
(b) L_2 Distance and L_2E ($\sigma = 0.05$)



(c) Negative Log-Likelihood and MLE (σ=0.05)



(d) L_2 Distance and L_2E Estimator ($\sigma = 0.2$)



(e) Negative Log-Likelihood and MLE ($\sigma = 0.2$)

Figure 3.7: Toy example demonstrating the alignment of two point-sets without outliers. (a) Two 1D point-sets A and B which overlap exactly. As point-set B translates with respect to point-set A, the L_2 distance between Gaussian mixtures (constructed from the point-sets using kernel density estimation) and the negative log-likelihood is evaluated and plotted for different scales σ. (b) At a scale of $\sigma = 0.05$, the L_2E estimator is globally-optimal and the profile of the L_2 distance function contains many local minima. (c) At the same scale, the MLE estimator is also globally-optimal, but the negative log-likelihood profile has fewer, much shallower local minima. (d)–(e) At larger scales ($\sigma = 0.2$), both profiles smooth out and both still find the global optimum. The estimators (L_2E and MLE) that minimise the L_2 distance and negative log-likelihood are marked as black dots.

3.4.7 L_2 Distance between von Mises–Fisher Mixtures

An L_2 distance between densities can also be applied to directional data, with the same properties of robustness and asymptotic efficiency. However, directional data densities have their own set of challenges, with many formulations of the L_2 distance not having a closed form. An exception to this is the L_2 distance between von Mises–Fisher Mixture Models (vMFMMs) on the sphere, used in Straub et al. [2017], which is both robust to outliers and can be calculated in closed-form. Unfortunately, vMFMMs are not as expressive as other mixture models on the sphere, being analogous to GMMs with isotropic covariances. Moreover, the L_2 distance only has a closed-form for vMFMMs in $\mathbb{R}^2$ or $\mathbb{R}^3$ since it requires the evaluation of modified Bessel functions of the first kind, which are not closed-form for higher dimensions. In addition, since they only model directional information, they cannot easily be used to determine translation, even if one or both vMFMMs were generated from point-set data.

The advantages of representing 3D directional sensor data as von Mises–Fisher mixtures were discussed in Section 3.3.6. Two key benefits are that vMFMMs can admit arbitrarily accurate estimates of noisy sphere-projected surface densities and can be computed efficiently from point-set or bearing vector set data. However, they are not suitable for modelling positional data such as point-sets and therefore cannot be used for certain alignment problems.

Let $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_{ki}, \kappa_{ki}, \phi_{ki}\}_{i=1}^{n_k}$ be the parameter set of an arbitrary n_k-component vMFMM with means $\boldsymbol{\mu}_{ki}$, concentrations κ_{ki}, and mixture weights $\phi_{ki} \geqslant 0$, where $\sum_{i=1}^{n_k} \phi_{ki} = 1$. Then the L_2 distance between von Mises–Fisher mixtures in $\mathbb{R}^3$ for a rotation $\mathbf{R} \in SO(3)$ is given by

$$\begin{aligned} f(\mathbf{R}) &= \int_{S^2} \left[p(\mathbf{f}|T(\boldsymbol{\theta}_1, \mathbf{R})) - p(\mathbf{f}|\boldsymbol{\theta}_2)\right]^2 \mathrm{d}\mathbf{f} \\ &= \int_{S^2} \left[p(\mathbf{f}|T(\boldsymbol{\theta}_1, \mathbf{R}))\right]^2 - 2p(\mathbf{f}|T(\boldsymbol{\theta}_1, \mathbf{R}))p(\mathbf{f}|\boldsymbol{\theta}_2) + \left[p(\mathbf{f}|\boldsymbol{\theta}_2)\right]^2 \mathrm{d}\mathbf{f} \end{aligned} \tag{3.77}$$

where $p(\mathbf{f}|\boldsymbol{\theta})$ is the von Mises–Fisher mixture probability density function (3.40) and T is the function defined by

$$\{\boldsymbol{\mu}_i, \kappa_i, \phi_i\}_{i=1}^{n} \mapsto \{\mathbf{R}\boldsymbol{\mu}_i, \kappa_i, \phi_i\}_{i=1}^{n} \tag{3.78}$$

that maps a von Mises–Fisher mixture parameter set to another parameter set representing a rotation of the original vMFMM. A closed-form objective function can be found using the same observations as for the GMMs. The first term of (3.77) is invariant under rotations and the last term is independent of the rotation, and therefore both terms can be dropped. Secondly, the middle term is the inner product of two von

Mises–Fisher mixtures and has a closed form. This can be seen by substituting (3.35) into (3.77) giving

$$\begin{aligned}
&\int_{S^2} p(\mathbf{f}|T(\boldsymbol{\theta}_1, \mathbf{R}))p(\mathbf{f}|\boldsymbol{\theta}_2)\,\mathrm{d}\mathbf{f} \\
&\quad= \int_{S^2} \sum_{i=1}^{n_1} \phi_{1i}\,\mathrm{vMF}(\mathbf{f}|\mathbf{R}\boldsymbol{\mu}_{1i}, \kappa_{1i}) \sum_{j=1}^{n_2} \phi_{2j}\,\mathrm{vMF}\Big(\mathbf{f}\Big|\boldsymbol{\mu}_{2j}, \kappa_{2j}\Big)\,\mathrm{d}\mathbf{f} && (3.79)\\
&\quad= \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \phi_{1i}\phi_{2j} \int_{S^2} \mathrm{vMF}(\mathbf{f}|\mathbf{R}\boldsymbol{\mu}_{1i}, \kappa_{1i})\,\mathrm{vMF}\Big(\mathbf{f}\Big|\boldsymbol{\mu}_{2j}, \kappa_{2j}\Big)\,\mathrm{d}\mathbf{f} && (3.80)\\
&\quad= \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \phi_{1i}\phi_{2j} C_3(\kappa_{1i})C_3(\kappa_{2j}) \int_{S^2} \exp\Big(\Big(\kappa_{1i}\mathbf{R}\boldsymbol{\mu}_{1i} + \kappa_{2j}\boldsymbol{\mu}_{2j}\Big)^{\mathsf{T}}\mathbf{f}\Big)\,\mathrm{d}\mathbf{f} && (3.81)\\
&\quad= \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \phi_{1i}\phi_{2j} C_3(\kappa_{1i})C_3(\kappa_{2j}) \Big[C_3\Big(\Big\|\kappa_{1i}\mathbf{R}\boldsymbol{\mu}_{1i} + \kappa_{2j}\boldsymbol{\mu}_{2j}\Big\|\Big)\Big]^{-1} && (3.82)
\end{aligned}$$

where (3.36) is substituted into (3.80) to get (3.81) and the integral is the inverse of the normalisation constant of a vMF density with $\kappa = \|\kappa_{1i}\mathbf{R}\boldsymbol{\mu}_{1i} + \kappa_{2j}\boldsymbol{\mu}_{2j}\|$ and $\boldsymbol{\mu} = (\kappa_{1i}\mathbf{R}\boldsymbol{\mu}_{1i} + \kappa_{2j}\boldsymbol{\mu}_{2j})/\kappa$, which can be seen from the identity

$$\int_{S^2} C_3(\kappa)\exp(\kappa\boldsymbol{\mu}^{\mathsf{T}}\mathbf{f})\,\mathrm{d}\mathbf{f} = 1. \tag{3.83}$$

Finally, by substituting (3.39) and (3.82) into (3.77) and removing the invariant or independent terms, the objective function to optimise the L_2 distance between von Mises–Fisher mixtures is given by

$$f(\mathbf{R}) = -\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \frac{\phi_{1i}\phi_{2j}\kappa_{1i}\kappa_{2j}}{\sinh\kappa_{1i}\sinh\kappa_{2j}} \frac{\sinh\Big\|\kappa_{1i}\mathbf{R}\boldsymbol{\mu}_{1i} + \kappa_{2j}\boldsymbol{\mu}_{2j}\Big\|}{\Big\|\kappa_{1i}\mathbf{R}\boldsymbol{\mu}_{1i} + \kappa_{2j}\boldsymbol{\mu}_{2j}\Big\|}. \tag{3.84}$$

The argument for the robustness of this objective function proceeds in the same way as for GMMs in Section 3.4.6 and will not be reiterated.

3.5 Local Optimisation for Alignment

Given an objective function, the next step is to optimise it to find the parameter vector that best aligns the sensor data. In this section, the local optimisation problem is formulated and methods for optimising it in a neighbourhood local to the initial parameter vector are briefly outlined. The keyword *local* means that the optimiser will at best find a local optimum and is not searching across the entire parametric domain. For convex objective functions, this is not problematic because the local optimum is the global optimum. However, geometric alignment problems are typically non-convex,

usually exceedingly so. Therefore, a good parameter initialisation is essential for local optimisation to retrieve the correct result.

The local optimisation problem for geometric sensor data alignment can be written as follows. Given an objective function f and an initial rotation $\mathbf{R}^0$ and translation $\mathbf{t}^0$,

$$\begin{aligned} \underset{\mathbf{R},\mathbf{t}}{\text{optimise}} \quad & f(\mathbf{R},\mathbf{t}) \\ \text{subject to} \quad & \mathbf{R} \in SO(n) \\ & \mathbf{t} \in \mathbb{R}^n \end{aligned} \tag{3.85}$$

and return the arguments $\mathbf{R}^*$ and $\mathbf{t}^*$ at the optimum. Note that any optimisation problem can be written as a minimisation problem modulo a negative sign. If available or required, functions to compute the first and second order partial derivatives with respect to the parameters (the gradients and Hessian) can also be supplied to the solver.

Hence, the alignment problem is a constrained nonlinear optimisation problem, with a constraint on the rotation parameters ensuring that they represent a valid rotation. An optional constraint can be placed on the translation parameters if only a subset of $\mathbb{R}^n$ is to be searched. The main class of local optimisation methods used in geometric alignment are iterative methods that converge to a solution after a non-deterministic number of steps. This class can be further subdivided into those methods that just require the function, such as golden-section search and interpolation methods; those methods that also require gradients or approximate gradients from finite differences, such as gradient descent, Gauss-Newton, quasi-Newton and Levenberg–Marquardt methods; and those methods that also require Hessians or approximate Hessians, such as Newton's method and interior point methods. For many of these approaches, line searches or trust regions are used to ensure the method converges. There is often a trade-off between the number of iterations and the computational complexity of each iteration, so the choice of method is problem-dependent.

The constraints on the rotation parameters require some care. They can be enforced by using manifold optimisation or Lagrange multipliers, or by renormalising the rotation representation at each optimisation iteration. Where a quaternion parametrisation is used, a common approach is to allow the parameters to vary freely during each optimisation iteration and then project the updated solution back to the space of valid rotations by normalising the quaternion [Schmidt and Niemann, 2001]. Alternatively, the unit-norm constraint can be enforced by using Lagrange multipliers. Where a rotation matrix parametrisation is used, Gram–Schmidt orthonormalisation can be applied. In contrast, the constraints on the translation parameters can often be expressed as box constraints, which can be solved by optimisers such as BFGS-B.

In whatever way the local optimisation problem is solved, it will still be suscepti-

ble to local optima since the objective function is highly non-convex over the search space. There are many heuristic approaches to alleviate the problem of converging to local optima. One such approach is to adopt a multi-resolution approach by decreasing any scale parameter at each iteration. This annealing approach starts with a smooth objective function, enabling large motions towards a good region of the search space, and progressively reduces the smoothness of the objective function, enabling smaller motions towards more precise optima. Another approach is to repeat the optimisation process many times from different initial parameter settings that systematically, randomly or quasi-randomly cover the domain. While the former approach can help widen the algorithm's basin of convergence, the latter approach explicitly searches over a larger region of the parametric domain. This provides a smooth transition between local and global optimisation algorithms, from searching within a local neighbourhood region to searching across an entire domain.

3.6 Global Optimisation for Alignment

In contrast to local optimisation, global optimisation searches over the entire parametric domain. Optimising over the entire domain does not guarantee that the optimal solution is found, however; for that, a globally-optimal algorithm is required. In this section, the global optimisation problem is formulated and methods for optimising it in across the entire parametric domain are briefly outlined. The fundamental property of global optimisation algorithms is that a good parameter initialisation is not required.

The global optimisation problem for geometric sensor data alignment uses the same formulation as (3.85), except parameter initialisations are not required. As for the local optimisation formulation, functions to compute the first and second order partial derivatives with respect to the parameters can also be supplied to the solver if required.

Non-convexity and non-differentiability are the primary motivators for global optimisation. Due to the non-convexity of the objective function, local optimisation can at best find a local optimum, typically the optimum closest in some sense to the initial parameter vector. While objective functions can be designed to smooth out the function landscape and widen the basin of convergence of the correct solution, this cannot mitigate the problem entirely. The more the objective function is smoothed, the less it represents the underlying geometric alignment problem. Thus, while it might be easier to find the solution, the solution may no longer be relevant to the alignment problem. Non-differentiability presents a different challenge. Most good local optimisation algorithms need to compute exact or approximate gradients in order to converge to a local optimum. As a result, continuous and differentiable functions can be handled well by local optimisers, but discrete and non-differentiable functions cannot.

As observed in Section 3.5, the alignment problem is a constrained nonlinear optimisation problem. There are many approaches to solving this problem globally. It can be helpful to think of geometric alignment algorithms as jointly solving for the transformation and correspondences. Hence, there are two options for a global algorithm: to lead with search over the transformation space or the correspondence space.

The first approach has transformation search lead the correspondence search. As foreshadowed in Section 3.5, a naïve approach is to run multiple instances of a local optimisation algorithm using different initial parameters. These initialisations can cover the parametric domain systematically or randomly, however it can be useful to use an optimally self-avoiding quasi-random distribution [David et al., 2002]. More sophisticated approaches in this class apply methods such as particle filtering [Sandhu et al., 2010], genetic algorithms [Silva et al., 2005] or Kalman filtering [Moreno-Noguer et al., 2008] to intelligently select the parameter initialisations.

While any local optimisation algorithm can be made global in this way, specifically global algorithms tend to approach the problem in a different way. In particular, many methods search over the parametric domain implicitly by instead searching explicitly over the correspondences. Hence in these approaches correspondence search leads the transformation search. RANdom SAmple Consensus (RANSAC), introduced by Fischler and Bolles [1981], is a robust but non-deterministic global method for solving the consensus set maximisation problem (3.62). As such, it requires a set of putative correspondences and is therefore not solving the geometric alignment problem. Nonetheless, it is a fundamental algorithm and the underlying principle has been extended to correspondence-free alignment. RANSAC maximises the consensus set by stochastically generating a minimal set, computing the transformation from this set, computing the cardinality of the consensus set given this transformation, updating the parameters if a better cardinality was found, and repeating. The minimal sets differ depending on the data and the problem. For example, three 3D–3D point correspondences are a minimal set for point-set registration and three 2D–3D point correspondences are a minimal set for camera pose estimation. The RANSAC algorithm can be modified to work without correspondences by introducing another stochastic step, where a minimal set is selected at random from dataset $\mathcal{X}_2$ after one has been selected at random from dataset $\mathcal{X}_1$ [Grimson, 1990]. This approach scales very poorly with the number of data elements and is not practical for most problems.

The most common way to pre-compute a correspondence set for RANSAC is using feature correspondences [Rusu et al., 2009]. Within a single modality, this can be an effective strategy, particularly if the features are robust and reproducible. However, for feature extraction across multiple modalities, such as 2D–3D alignment, this is a non-trivial unsolved problem. Even in a single modality, factors such as variable

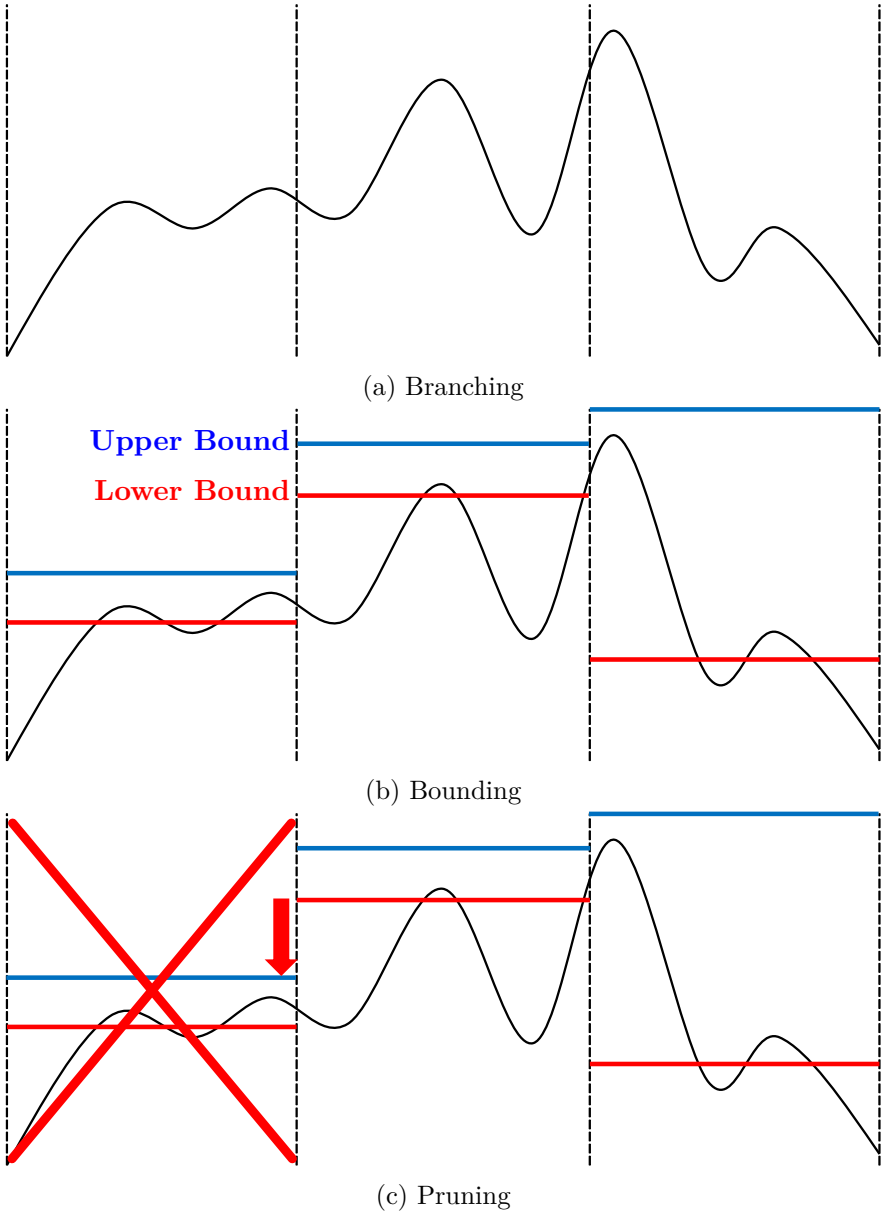
sampling densities, repetitive features and occlusions make the correspondence problem challenging. However, one class of RANSAC-based approaches does not require feature extraction: congruent set methods. These methods extract all approximately-congruent near-minimal sets directly from the raw sensor data and use RANSAC to find the alignment using these sets. The 4-Points Congruent Sets method (4PCS) [Aiger et al., 2008], which provides a way to rapidly extract coplanar 4-point sets, its extension Super4PCS [Mellado et al., 2014], which exploits a clever data structure to achieve linear-time performance, and the 2-Points+Normal Sets (2PNS) method [Raposo and Barreto, 2017], which uses normals to reduce the number of points needed, are examples of these methods.

The primary disadvantage of all of these global optimisation methods is that they do not necessarily converge to the global optimum. While they are not restricted to finding the local optimum in the neighbourhood of a parameter initialisation, they may instead find a local optimum somewhere else in the parametric domain. To guarantee that the global optimum has been found, globally-optimal methods, such as branch-and-bound, must be used.

3.7 Branch-and-Bound for Globally-Optimal Alignment

To solve highly non-convex and NP-hard optimisation problems, such as geometric sensor data alignment, the global optimisation technique of Branch-and-Bound (BB) [Land and Doig, 1960; Lawler and Wood, 1966] may be applied, outlined in Figure 3.8. It provides a framework for optimisation with some guarantees that the solution found is the global optimum within the domain. Depending on the objective and bounding functions, BB may guarantee full global optimality or a weaker ϵ-suboptimality. The latter ensures that the solution is within ϵ of the true global optimum, for a user-defined value ϵ. The trade-off is typically between optimality and runtime, with smaller values of ϵ requiring longer runtimes.

To apply the BB paradigm, a suitable means of parametrising and branching (or partitioning) the function domain must be found, as well as an efficient way to calculate upper and lower bounds of the function on each branch. An important requirement is that the upper and lower bounds must converge as the size of the branches tend to zero. BB algorithms that are ϵ-suboptimal have bounding functions that converge asymptotically as the branch size decreases. That is, the limit as the branch size δ tends to zero of the difference between the upper and lower function bounds $\bar{f}$ and $\underline{f}$ is given by $\lim_{\delta \to 0}(\bar{f}(\delta) - \underline{f}(\delta)) = 0$. In contrast, those that are fully optimal have bounding functions that converge to zero before the limit: $\bar{f}(\delta) - \underline{f}(\delta) = 0$ for $\delta > 0$. A BB algorithm systematically subdivides the search space using a branching strategy and



(a) Branching

(b) Bounding

(c) Pruning

Figure 3.8: Overview of the branch-and-bound algorithm for a 1D maximisation problem. A queue is initialised with a node containing the entire function domain, then the following steps are iterated: (a) remove a node from the queue and subdivide its domain into branch nodes; (b) evaluate upper and lower bounds of the function maximum within each branch; (c) if the bounds indicate that a branch cannot contain the global maximum, discarded it. A node is pruned if its upper bound is less than the greatest lower bound found so far.

then prunes the search space using the bounding functions. If halted at any time, the global optimum (or an ϵ-suboptimum) is guaranteed to be attained in the remaining branches and not attained in the discarded branches.

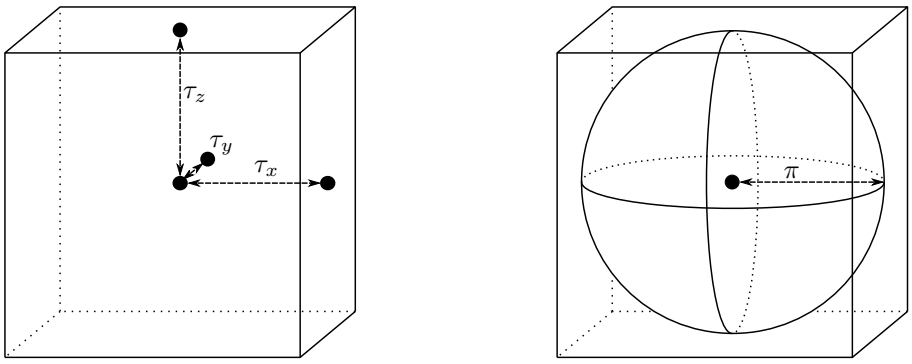
While the bounds need to be computationally efficient to calculate, the time and

memory efficiency of the algorithm also depends on how tight the bounds are. In this context, *tight* refers to the how well the bounding function approximates the actual maximum and minimum function values for any given branch. This affects the time and memory efficiency of the algorithm because tighter bounds reduce the search space quicker by allowing suboptimal branches to be pruned. These two requirements are generally in opposition and must be optimised together. For this reason, it is always important to check whether more sophisticated, tighter bounding functions increase the time or memory efficiency of the algorithm: theoretical novelty is not a sufficient criterion. One confounding factor, however, is that the time and memory efficiency of a bounding function may be dependent on the data itself. Therefore, it is recommended that a new bounding function be tested with a range of typical datasets.

3.7.1 Parametrising the Domain

To find a globally-optimal solution to the geometric sensor data alignment problem, the objective function must be optimised over the domain of 3D motions, that is, the group $SE(3) = SO(3) \times \mathbb{R}^3$. However, the space of these transformations is unbounded. Therefore, to apply the BB paradigm, the space of translations is restricted to be within the bounded set Ω_t. Since Ω_t can be arbitrarily but nonetheless finitely large, it is often reasonable that the optimal translation is contained in this set. If additional domain specific knowledge is available, a smaller set may be justifiable.

The parametrisation of the domain is a design choice that may depend on the specific problem or bounding strategy. Several options were presented in Section 3.1. For the alignment problem, translation space $\mathbb{R}^2$ or $\mathbb{R}^3$ is typically parametrised with 2- or 3-vectors in a Cartesian coordinate system within a bounded domain chosen as the cuboid Ω_t, as shown in Figure 3.9(a). In 2D, rotation space is parametrised by the scalar rotation angle θ. In 3D, both angle-axis 3-vectors and unit quaternions are useful parametrisations of rotation space $SO(3)$. For angle-axis vectors, as discussed in Section 3.1.2, the space of all 3D rotations can be represented as a solid ball of radius π in $\mathbb{R}^3$. For ease of manipulation and branching, the 3D cube that circumscribes the π-ball can be used as the rotation domain Ω_r [Li and Hartley, 2007], as shown in Figure 3.9(b). Quaternions may also be used, although the tessellation and subdivision of a hemisphere of the space of unit quaternions S^3 is non-trivial. One tetrahedron-based approach is presented in Straub et al. [2017], where the initial exactly uniform tessellation of rotation space is found by normalising the vertices of a 4D 600-cell and selecting the resulting tetrahedra that cover a single hemisphere. The subdivision scheme is also non-trivial, since a subdivision pattern must be chosen for each tetrahedron to ensure the rotation range shrinks, and does not preserve exact uniformity.



(a) Translation Domain Ω_t (b) Rotation Domain Ω_r

Figure 3.9: Parametrisation of translation and rotation domains in 3D. (a) The translation space $\mathbb{R}^3$ can be parametrised by 3-vectors bounded by a cuboid with half-widths $[\tau_x, \tau_y, \tau_z]$. (b) The rotation space $SO(3)$ can be parametrised by angle-axis 3-vectors in a radius-π ball.

3.7.2 Branching the Domain

In its most general form, branch-and-bound partitions the space of parameters hierarchically and may therefore be structured as a tree. For any objective function $f : \Omega \to \mathbb{R}$, where Ω is the parametric domain, the BB algorithm subdivides the domain into k regions whose union is the original domain and are preferably but not necessarily disjoint. In this scheme, k is the branching factor and each subdivided region of the search space is a node of the tree.

Commonly used structures for branching different dimensional spaces include the octree family (quadtrees, octrees, hyperoctrees), kd-trees, triangular or tetrahedral tessellations, and golden-section interval selection. In the specific case of an angle-axis parametrisation, the domain of the π-ball-circumscribing cube can be branched into sub-cubes using an octree data structure. In the case of a quaternion parametrisation, the domain of tetrahedrons providing a cover of a hemisphere of S^3 can be branched into sub-tetrahedra using a tetrahedron tessellation scheme.

In some situations it is justifiable to reduce the dimensionality of the search space of BB by nesting a BB problem within another BB problem. While the entire search space is still explored in this scheme, it is often computationally preferable to solve a smaller problem multiple times than a large problem once. For alignment problems, translation and rotation search can be nested in this way.

3.7.3 Bounding the Branches

The success of a BB algorithm is predicated on the quality of its bounds, where quality is assessed with respect to computational cost and tightness. For any BB optimisation problem, the objective function f needs to be bounded when evaluated within some

subset Ψ of the parametric domain Ω, that is $\Psi \subseteq \Omega$. This allows the algorithm to perform a feasibility test on the subset of the domain. Let $\underline{f}_\Psi$ and $\bar{f}_\Psi$ be lower and upper bounds of the function for any subset Ψ of Ω. Then for a minimisation problem

$$\underline{f}_\Psi \leqslant \min_{\boldsymbol{\theta} \in \Psi} f(\boldsymbol{\theta}) \leqslant \bar{f}_\Psi \tag{3.86}$$

and for a maximisation problem

$$\underline{f}_\Psi \leqslant \max_{\boldsymbol{\theta} \in \Psi} f(\boldsymbol{\theta}) \leqslant \bar{f}_\Psi \tag{3.87}$$

where $\boldsymbol{\theta}$ is a parameter vector in the set Ψ. For the minimisation problem, an entire branch (subset) can be pruned (discarded) whenever its lower bound is greater than the lowest upper bound found so far. The reverse process holds for a maximisation problem. In this way, branches that cannot contain the global optimum are found to be infeasible and are discarded.

To prove that a BB algorithm converges to the global optimum as the branch size tends to zero, it is only necessary to show that the upper and lower bounds are valid, that is, provably correct, and converge as the size of the branches tend to zero. The bounds are not required to converge asymptotically or continuously and may converge well before the branch size diminishes appreciably, especially for discrete functions.

A BB algorithm for a minimisation (or maximisation) problem terminates when the difference between the lowest lower bound (or highest upper bound) across all remaining branches and the lowest upper bound (or highest lower bound) found so far is less than a user-specified threshold ϵ. For problems where $\epsilon = 0$, typically for discrete optimisation problems, full optimality is achieved. For problems where $\epsilon > 0$, typically for continuous optimisation problems, ϵ-suboptimality is achieved. In the latter case, the solution is within ϵ of the global optimum. Note that the ϵ gap applies to the function value, not the parameter values. The optimal parameter values may be entirely different than those that generated the ϵ-suboptimal solution. For this reason, it is important to ensure that the value of ϵ is as small as practical so that incorrect local optima are not returned by the algorithm. Unfortunately, it can be difficult to know in advance what a suitably small value would be for a given problem. As with other parameter selection problems, the best approach is often to use cross-validation with data that has a known ground-truth.

Another approach to mitigate the problem of ϵ-suboptimality is to instead terminate the algorithm when some condition about the remaining branches is met. That is, using parameter space criteria to decided when to terminate. For example, terminating when only a single cluster remains in the parameter space and the radius of the cluster-enclosing hypersphere is below a given threshold ϵ_r. The single cluster criterion

ensures that all 'distant' local minima have been excluded and all remaining feasible regions are proximal in parameter space. The hypersphere radius criterion ensures that the parameter vector associated with the best solution is at most ϵ_r from the optimum parameter vector by the Euclidean distance metric. The hypersphere radius criterion subsumes the single cluster criterion, however the latter is easier to evaluate and therefore is useful as a first condition. For the geometric alignment problem, the termination strategy can be split between translation and rotation domains with a different threshold for each. For example, BB can be set to terminate when the maximum possible deviation in translation and rotation is 0.01m and 0.1° respectively.

In BB, one of the bounds can be trivial to obtain. For a minimisation problem, an upper bound can be found by evaluating the function at any parameter vector in the subset Ψ. For a maximisation problem, a lower bound can be found in the same way. Typically, the parameter vector chosen is the one which is quickest to evaluate. However, there are many cases where a more sophisticated bound is useful. One strategy is to apply a local optimisation algorithm initialised at any parameter vector in the domain subset. However, consideration must be given to the computational cost of the bounding functions. Therefore, it may be sensible to only apply the more sophisticated bound when the cheap bound is better than any that have been previously discovered.

The other (non-trivial) bound tends to be problem-dependent and more sophisticated techniques can be applied. Again, consideration must be given to the computational cost, since sophisticated and tight bounds may be too slow to evaluate. The BB algorithm is consistently in tension between using sophisticated techniques and speed.

3.7.4 Search Strategies

There are several search strategies that can be used with branch-and-bound. The specific choice is problem-dependent, so in this section an overview of some alternatives is given. A search strategy is a rule for choosing which branch to explore or subdivide next. At any given stage, there may be a significant number of remaining branches, all of which will have some values associated with them. These will include the lower and upper function bounds computed for each branch and may include information such as the branch's size or level in the tree.

Depth-First Search

Two common tree search strategies are depth-first and breadth-first search. Depth-first search continually expands each tree node, following a single expansion path, until a branch is discarded or a maximum depth level is reached. It then returns to the previous level of the tree and expands the next node of that branch. This strategy uses

less memory than the other ones because it reaches the deeper levels where pruning is more likely before expanding shallower nodes. However, it may also fully expand unpromising nodes if it is explored before a reasonable optima has been found that facilitates pruning. That is, a depth-first strategy may traverse an entire path to the leaves of an initial branch which is soon shown to be unpromising. Alternatively, the structure of the problem and data may be such that a 'push-your-luck' strategy may be appropriate, finding the correct solution much earlier than a breadth-first approach.

Breadth-First Search

A breadth-first strategy searches all nodes on the same level before progressing to nodes on the next level. For most branching strategies, this search strategy is equivalent to exploring nodes that represent a larger subset of the parameter space before exploring smaller subsets. While the strategy systematically explores the parametric domain in a coarse-to-fine way, it is memory intensive since branches are retained more often than they are pruned at the upper level of the tree.

Best-First Search

Another search strategy is greedy or best-first search. This strategy expands the most promising nodes in the tree first. For a minimisation problem, this would be the node with the lowest lower bound. For a maximisation problem, it would be the node with the greatest upper bound. The utility of this approach depends on the specific objective function. For an objective function that does not change rapidly as the parameters change, this strategy is appropriate. For a discrete objective function where the function value is likely to change significantly in the neighbourhood of the parameter vector of the parent branch, it would not be an appropriate strategy.

Heuristic, Combination and Parallel Search

Other search strategies may invoke an easily-calculable heuristic which depends on the particular problem. For example, if previous experience suggests that the solution may lie in a particular region of parameter space, nodes closer to that region could be expanded first. This can be an effective way to incorporate prior knowledge about the solution into the algorithm without voiding the optimality guarantee.

In many cases, two or more strategies can be applied. For example, if the primary search strategy is breadth-first, the exploration order of the nodes within a level of the tree still needs to be determined. One of the other strategies, such as best-first, could be used to choose this ordering.

Parallel processing offers additional choices for search strategies. While breadth-first search lends itself to parallel implementations, other possibilities arise when multiple experiments can be run simultaneously. These include using a random search strategy for every instance of the algorithm and running many instances, or using each strategy in a different instance. The latter is the most robust approach to domain shifts, where specific knowledge about the datasets and therefore the appropriate search strategy can no longer be relied upon.

Search Strategies for Geometric Alignment

Many of these strategies have been applied to geometric alignment problems. Hartley and Kahl [2009] reported that both depth-first and breadth-first search worked effectively for 3D rotation search. Yang et al. [2016] found best-first search to be the most effective for 3D–3D alignment. Breadth-first search can also be suitable in many cases because searching a wide swathe of the transformation domain will often lead to a good function value being found early in the search [Campbell and Petersson, 2016]. Quickly finding a good best-so-far upper bound for minimisation or lower bound for maximisation is essential for BB since it facilitates the pruning of higher-level branches. In turn, this greatly reduces the amount of redundant calculations undertaken. It has been previously observed [Yang et al., 2016] that finding the global optimum of a geometric alignment problem requires a small fraction of the time required to guarantee that it is the global optimum. Since finding the optimum is not normally the limiting factor, the search strategy should aim to reduce the amount of redundant node expansions.

3.7.5 Branch-and-Bound Algorithm

To summarise the main details of the previous sections, the branch-and-bound algorithm for a minimisation problem using best-first search is presented in generic form in Algorithm 3.1. On line 1, the variable f^* that keeps track of the best function value found so far is initialised to infinity. In practise, a non-infinite value can be found by evaluating the objective function at any parameter vector in the domain. Next, the first branch is initialised to the entire parametric domain with associated upper and lower bounds (line 2). Then the branch is pushed into the empty priority queue (line 3).

The main loop now begins by accessing and removing the top element of the priority queue (line 5). Best-first search can be implemented using a priority queue with priority inversely proportional to the lower bound (for minimisation). The queue itself handles the tracking of the lowest lower bound, since this will always be the lower bound of the first (top) element in the queue. For other search strategies, this critical value will need to be updated by another method. For discrete objective functions, this can be achieved

Algorithm 3.1 Prototypical best-first branch-and-bound minimisation algorithm

Input: parametric domain Ω and tolerance ϵ
Output: ϵ-suboptimal function value f^* and corresponding parameters $\boldsymbol{\theta}^* \in \Omega$

1: Initialise the best-so-far function value: $f^* \leftarrow \infty$
2: Initialise the first branch (root): $\Psi \leftarrow \{\Omega, \underline{f} = -\infty, \bar{f} = \infty\}$
3: Add branch to priority queue Q: $Q \leftarrow \Psi$
4: **loop**
5: Remove branch Ψ with lowest lower-bound $\underline{f}$ from Q
6: **if** $f^* - \underline{f} \leqslant \epsilon$ **then** terminate
7: **for all** sub-branches Ψ_i **do**
8: Evaluate $\bar{f}_{\Psi_i}$
9: **if** $\bar{f}_{\Psi_i} < f^*$ **then** $(f^*, \boldsymbol{\theta}^*) \leftarrow g(\bar{f}_{\Psi_i}, \boldsymbol{\theta}_{\Psi_i})$
10: Evaluate $\underline{f}_{\Psi_i}$
11: **if** $\underline{f}_{\Psi_i} \leqslant f^*$ **then** add branch to queue: $Q \leftarrow \Psi_i$

using a histogram of lower bounds which counts the number of remaining branches in each category. Priority queues are useful data structures for BB because they have a time complexity of $\mathcal{O}(1)$ for finding the highest priority element and $\mathcal{O}(\log n)$ for insert and remove operations, in contrast to $\mathcal{O}(n)$ and $\mathcal{O}(1)$ for an unsorted list.

The algorithm terminates when the gap between the best function value and the lowest lower bound is less than the tolerance ϵ (line 6), which may be zero depending on the problem. On the next line (7), the branching step occurs, where Ψ is subdivided according to a refinement scheme, such as octree subdivision. Next, the upper bound of the function is evaluated for the current sub-branch Ψ_i (line 8). If this value is less than the best-so-far function value, f^* and the associated parameter vector are updated (line 9). Local optimisation can be incorporated at this step, abstracted here as a function g. The local optimisation algorithm is initialised with the parameter vector $\boldsymbol{\theta}_{\Psi_i}$ and may find a better function value f^* and parameter vector $\boldsymbol{\theta}^*$. Certain local algorithms provide the stronger guarantee that they will find a lower or equal function value. The final steps are shown in lines 10 and 11, where the lower bound of the function is evaluated for the current sub-branch Ψ_i and the sub-branch is pruned if $\underline{f}_{\Psi_i} > f^*$. It is clear from this step that the closer the lower bound is to the true function minimum on the sub-branch, the earlier the sub-branch will be removed.

A simplified example of the branching and bounding step is presented in Figure 3.10. It shows the branching of the partition Ψ into sub-branches Ψ_i for which the upper and lower function bounds are evaluated. Sub-branches Ψ_1 and Ψ_k are pruned because their lower bounds are greater than the best-so-far function value f^*. However, sub-branch Ψ_2 is further branched because its lower bound is less than the best-so-far function value. In addition, its upper bound is also less than the best-so-far function value so this value is updated with that upper bound.

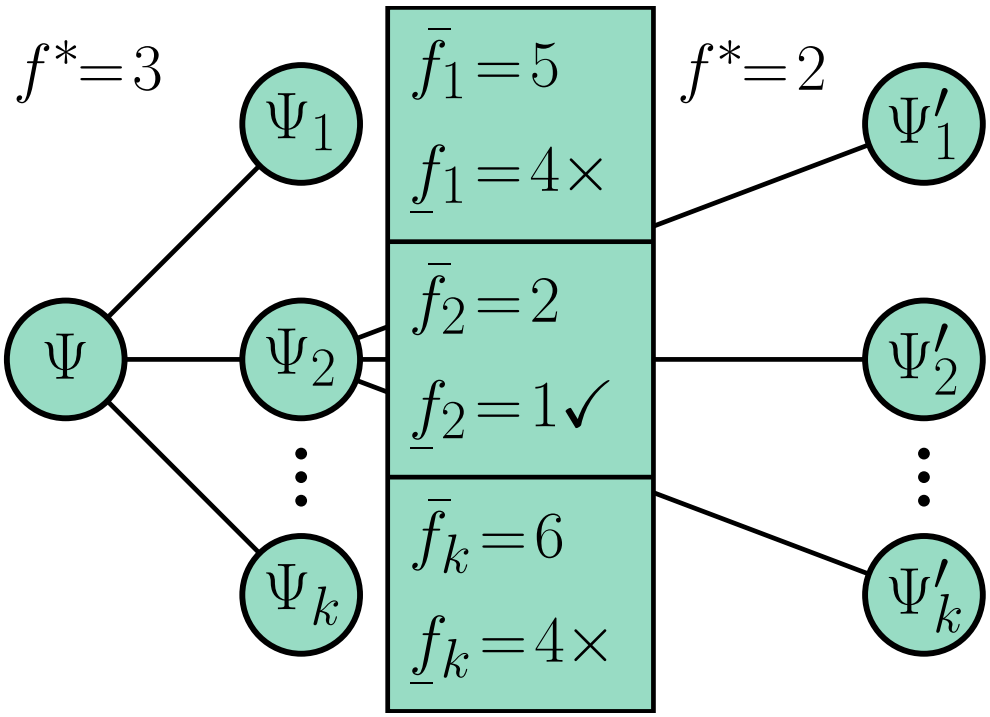


Figure 3.10: A simplified example of branching and bounding. Consider a best-so-far function value f^* of 3 and an initial partition Ψ of the parametric domain. Ψ is branched into k partitions Ψ_i, for each of which the upper and lower function bounds are evaluated. For Ψ_1 and Ψ_k, the lower bounds are greater than the best-so-far function value ($\underline{f}_1 > 3$ and $\underline{f}_k > 3$) so those branches are pruned and not explored further. For Ψ_2, the lower bound is less than the best-so-far function value ($\underline{f}_2 \leqslant 3$) so it is subdivided as shown. In addition, the upper bound is also less than the best-so-far function value ($\bar{f}_2 \leqslant 3$) so the value is updated: $f^* \leftarrow \bar{f}_2$.

3.8 Summary

This chapter formulated the geometric sensor data alignment problem and presented the technical background material necessary to understand the main algorithms used to solve this problem. Elements basic to the geometric alignment problem were presented first, including the parametrisations of rigid motions, distance measures and sensor data representations. Following this, different objective functions for geometric alignment were introduced, emphasising the progression from non-robust to robust functions with respect to how they operate in the presence of noise and random or structured outliers. Finally, optimisation was discussed, emphasising the progression from local to global optimisation techniques, and then from stochastic methods to guaranteed optimal branch-and-bound methods. These dual progressions, motivated at each step, are critical to the argument of this thesis.

The fundamental building blocks for geometric alignment that have been presented in this chapter will be used throughout the remaining work. The focus of Chapter 4 is robust 2D–2D and 3D–3D geometric alignment, which will use many of the elements from this toolkit. The remaining elements, global optimisation and branch-and-bound, will be incorporated into Chapters 5 and 6, where the focus is on robust and globally-optimal 3D–3D and 2D–3D geometric alignment.

Chapter 4

Robust nD–nD Alignment

The focus of this chapter is the geometric alignment of two sets of 2D or 3D positional sensor data, such as laser scans, where the data may be corrupted by noise and random or structured outliers. This can be used to solve the problem of estimating the 3 or 6 degrees-of-freedom pose of a 2/3D sensor with respect to a previously-acquired 2/3D point-set or the relative pose of two 2/3D sensors. Algorithms for solving the 2D–2D and 3D–3D registration problems have matured over time, with a twofold progression towards outlier robustness and global search. That is, non-robust local optimisation registration algorithms, such as the prototypical Iterative Closest Point (ICP) algorithm, have been improved by applying robust objective functions to reduce susceptibility to outliers and widen the region of convergence, such as Gaussian Mixture Alignment (GMA), and by applying global search to reduce susceptibility to local minima, such as Globally-Optimal ICP (Go-ICP). While much progress has been made in both these directions, outliers remain problematic to handle, particularly structured outliers caused by partial overlap and occlusion. Since typical instances of the geometric alignment problem have a large proportion of outliers and many local minima, a useful solver needs to be very robust to outliers and have a wide region of convergence.

In this chapter, a novel local optimisation algorithm for geometric alignment is proposed, which manifests strong robustness to outliers and a wide region of convergence. The algorithm, named Support Vector Registration (SVR), minimises the robust GMA objective function between Support Vector–parametrised Gaussian Mixtures (SVGMs). This novel data representation is generated by training a one-class support vector machine with a Gaussian radial basis function kernel and subsequently approximating the output function with a Gaussian mixture model. An SVGM has a sparse parametrisation that is adaptive to local surface complexity and, being a discriminative model of the point-set, is more invariant to viewpoint than a generative model since it does not model sampling artefacts, such as distance-dependent point density. The resulting SVR algorithm that minimises the L_2 distance between SVGMs is efficient, robust to outliers and sampling artefacts, and has a large region of convergence, as demonstrated

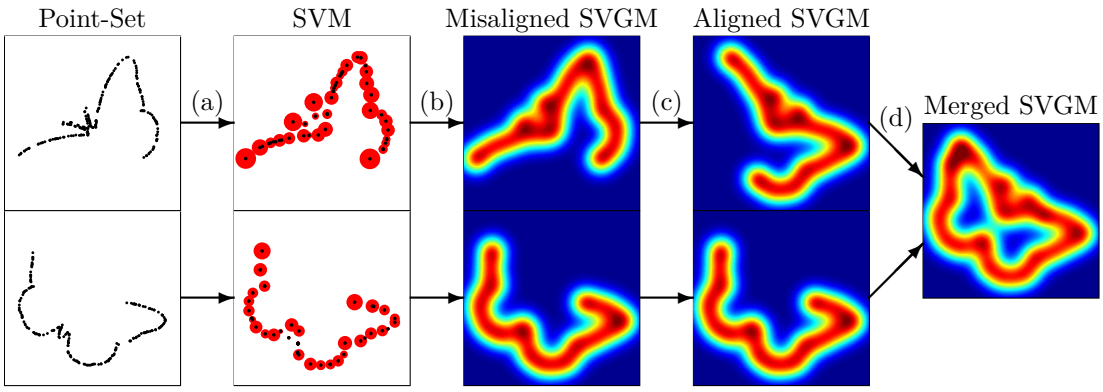


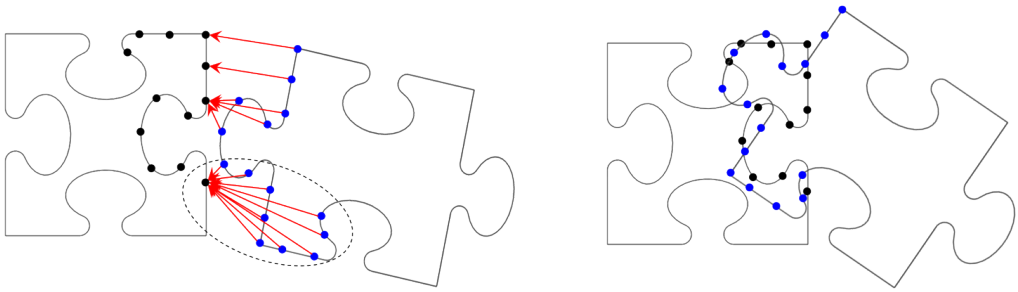
Figure 4.1: Robust point-set registration and merging framework. An nD point-set is represented as an SVGM by (a) training a one-class SVM and then (b) mapping it to a GMM. The mixtures are aligned using (c) the SVR algorithm, which minimises the L_2 distance between two SVGMs. Finally, the mixtures are parsimoniously fused using (d) the GMMerge algorithm. The SVMs are visualised as support vector points scaled by mixture weight and the SVGMs are coloured by probability value. Best viewed in colour.

on a range of 2D and 3D datasets. Finally, a novel algorithm is proposed to parsimoniously and equitably merge aligned mixture models. Hence, the work constitutes a framework for rigid 2D–2D and 3D–3D registration and merging, able to be used for reconstruction and mapping.

4.1 Introduction

Estimating the 3 or 6 degrees-of-freedom alignment of a set of 2/3D positional sensor data with respect to another set of 2/3D positional sensor data is the core task for solving the 2D–2D or 3D–3D rigid registration problem. A general-purpose registration algorithm that operates on positional sensor data, such as a point-set, may not assume that other information is available, such as colour, semantic labels or mesh structure. Gaussian Mixture Alignment (GMA), the problem of finding the transformation that best aligns one Gaussian mixture with another, has a natural application to point-set registration, visualised in Figure 4.1, which endeavours to solve the same problem as GMA for discrete point-sets in $\mathbb{R}^n$. Indeed, the Iterative Closest Point (ICP) algorithm [Besl and McKay, 1992; Zhang, 1994] and several other local registration algorithms [Chui and Rangarajan, 2000a,b; Tsin and Kanade, 2004; Myronenko and Song, 2010] can be interpreted as special cases of GMA [Jian and Vemuri, 2011].

Applications of 2D–2D and 3D–3D rigid registration include merging multiple partial scans into a complete model [Blais and Levine, 1995; Huber and Hebert, 2003]; using registration results as fitness scores for object recognition [Johnson and Hebert,



(a) Missing correspondences can lead to incorrect data association

(b) Optimisation gets trapped in local minima

Figure 4.2: Local optimisation techniques such as Iterative Closest Point (ICP) can be susceptible to missing correspondences and local minima. Missing correspondences can lead to incorrect data association and, without a good initialisation, optimisation can get trapped in local minima, producing erroneous alignment results.

1999; Belongie et al., 2002]; registering a view into a global coordinate system for sensor localisation [Nüchter et al., 2007; Pomerleau et al., 2013]; fusing cross-modality data from different sensors [Makela et al., 2002; Zhao et al., 2005]; and finding relative poses between sensors [Yang et al., 2013a; Geiger et al., 2012].

The dominant solution for 2D–2D and 3D–3D rigid registration is the ICP algorithm [Besl and McKay, 1992; Zhang, 1994] and variants, due to its conceptual simplicity, ease of use and good performance. However, ICP is limited by its assumption that closest point pairs should correspond, which fails when the point-sets are not coarsely aligned or the moving 'model' point-set is not a proper subset of the static 'scene' point-set. The latter occurs frequently, since differing sensor viewpoints and dynamic objects lead to occlusion and partial-overlap. This closest-point assumption means that ICP is susceptible to missing correspondences, which lead to incorrect data association, and local minima, in which the optimisation gets trapped, producing erroneous estimates without a reliable means of detecting failure, as shown in Figure 4.2.

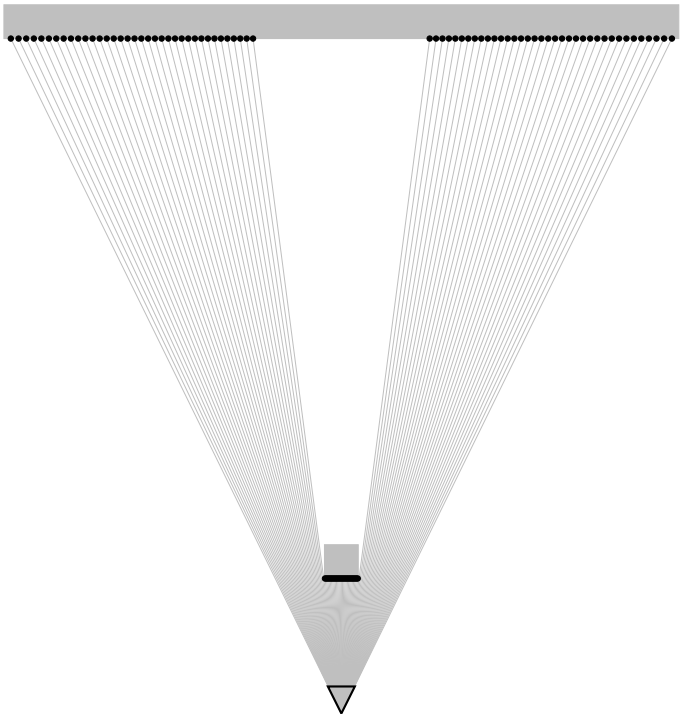
Gaussian mixture alignment [Chui and Rangarajan, 2000a; Tsin and Kanade, 2004; Jian and Vemuri, 2011] mitigates these problems by eschewing explicit correspondences and using a robust objective function. By aligning point-sets without establishing explicit point correspondences, GMA is less sensitive to missing correspondences from partial overlap or occlusion and is less susceptible to local minima, having a wider basin of convergence. Robust objective functions can also be applied, such as the L_2 distance between mixtures [Jian and Vemuri, 2011].

However, the transformation that aligns the Gaussian mixtures only corresponds to the transformation that aligns the underlying surfaces if the mixtures represent those surfaces adequately. Existing methods use generative models that optimise represen-

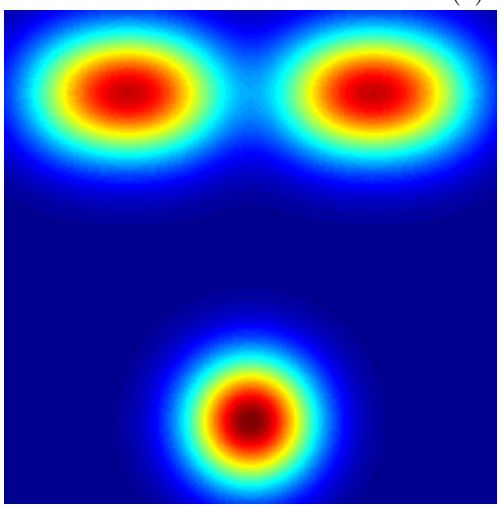
tation, modelling the scene as sampled by the sensor instead of the scene itself. As a result, sampling artefacts, such as occlusions and variable point densities that depend on the distance of the surface from the sensor, are also modelled and thus the model is not especially invariant to viewpoint. While this is unavoidable to some extent, it can be reduced by using a discriminative model that optimises classification, deciding whether the sampled points are well-classified by a surface. This implicit surface is closer to the underlying surface than the probability density of a generative model, since it regularises the sampled points, creating a smoother surface while still adapting to local surface complexity. Moreover, it does not depend strongly on the point density since a dense cluster of points can be classified equally well by a surface as a few points in the same location. It also helps to alleviate the problem of occlusions, which create holes in the sampled surface behind the occluding objects. With a generative model, the lower density edges of the holes are given little weight, whereas a greater weight will be assigned by a discriminative model, helping to in-fill the region. The differences between generative and discriminative models are demonstrated in Figure 4.3.

In this chapter, a Gaussian mixture alignment approach is proposed to solve the nD–nD rigid registration problem using a discriminative sensor data model. The approach has a wider region of convergence than existing local optimisation methods and is more robust to sampling artefacts and structured outliers from occlusions and partial overlap. The central idea is that the robustness of registration is dependent on the data representation used. The Support Vector–parametrised Gaussian Mixture (SVGM), a continuous data representation, is introduced for this purpose. An SVGM is generated from a discrete point-set by training a discriminative model using a one-class Support Vector Machine (SVM) and mapping it to a Gaussian Mixture Model (GMM), as shown in Figure 4.1. Since an SVM is parametrised by a sparse, intelligently-selected subset of data points, an SVGM is compact, adaptive to local surface complexity, and robust to sampling artefacts including varying point densities, noise, and occlusions [Van Nguyen and Porikli, 2013], crucial qualities for efficient and robust registration.

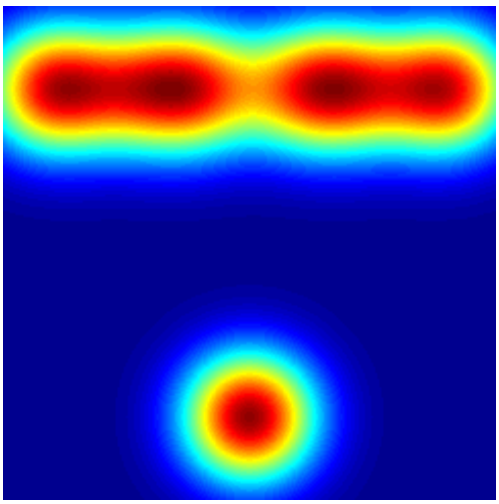
The Support Vector Registration (SVR) algorithm minimises the L_2 distance between SVGMs, which is inherently robust to outliers as discussed in Section 3.4.6. Unlike the benchmark GMA algorithm in Jian and Vemuri [2011], SVR uses an adaptive, sparse and discriminative representation with non-uniform, data-driven mixture weights, enabling faster performance and improving the robustness to sampling artefacts and structured outliers. Finally, a novel Gaussian Mixture Merging (GMMerge) algorithm is proposed that parsimoniously and equitably merges aligned mixtures. Merging Gaussian mixtures is useful for applications where each point-set may contain unique information, such as reconstruction and mapping. The full framework for robust point-set registration and merging using SVGMs is visualised in Figure 4.1.



(a) Point-set



(b) Generative model



(c) Discriminative model

Figure 4.3: Generative and discriminative models for sensor data representation. (a) A 2D point-set with a wall (light grey rectangle) and an occluding structure (light grey square) closer to the sensor. For a constant angular resolution, there is a higher point density on the occluding structure because the density is dependent on the distance from the sensor. (b) A Gaussian mixture constructed using a generative model (kernel density estimation). Undue weight is given to regions where the point-set is dense, such as the occluding structure and the centre of the two sampled regions on the wall, even though this may be an artefact of the sampling method. (c) A Gaussian mixture constructed using a discriminative model (support vector machine) with the same scale σ. The probability density does not depend strongly on point density. As a result, the probability density is not biased by regions of high point density and can in-fill small occluded regions.

The chapter is organised as follows: the problem is contextualised by summarising the relevant literature in Section 4.2; a robust sensor data representation, the SVGM, is introduced in Section 4.3; an algorithm is proposed for robust 2D–2D and 3D–3D registration by minimising the L_2 distance between two SVGMs in Section 4.4; an algorithm is proposed for parsimoniously merging SVGMs in Section 4.5; and the framework's performance is evaluated and discussed in Sections 4.6 and 4.7.

4.2 Related Work

The large quantity of work published on ICP, its variants and other local registration techniques precludes a comprehensive list. The reader is directed to the surveys on ICP variants [Rusinkiewicz and Levoy, 2001; Pomerleau et al., 2013] and recent 3D point-set and mesh registration techniques [Tam et al., 2013] for additional background. Of relevance to this work are extensions that improve robustness to outliers, including trimming [Chetverikov et al., 2005] and outlier rejection [Zhang, 1994; Granger and Pennec, 2002], and extensions that enlarge ICP's region of convergence, including LM-ICP [Fitzgibbon, 2003], which applies the Levenberg-Marquardt algorithm [Moré, 1978] and a distance transform to optimise the ICP error without establishing explicit point correspondences.

The family of Gaussian mixture alignment approaches, to which this work belongs, also seeks to improve robustness to outliers and poor pose initialisations. Notable GMA-related algorithms for rigid and non-rigid registration include Robust Point Matching [Chui and Rangarajan, 2003] that uses soft assignment and deterministic annealing, Coherent Point Drift [Myronenko and Song, 2010] and Kernel Correlation [Tsin and Kanade, 2004] that minimises a distance measure between mixtures. The Gaussian Mixture Model Registration (GMMReg) algorithm [Jian and Vemuri, 2011] defines an equally-weighted Gaussian at every point in the set with identical and isotropic covariances and minimises the robust L_2 distance between densities. The Normal Distributions Transform (NDT) algorithm [Magnusson et al., 2007; Stoyanov et al., 2012] defines Gaussians for every cell in a grid and estimates full data-driven covariances.

Unlike these local optimisation algorithms, approaches that use global optimisation are not dependent on the initial transformation. There are many heuristic or stochastic methods for global alignment that are not guaranteed to converge to the optimal alignment, such as particle filtering [Sandhu et al., 2010], genetic algorithms [Silva et al., 2005] and feature-based alignment [Rusu et al., 2009]. A recent example is Super 4PCS [Mellado et al., 2014], a random sampling method that uses four-point congruent sets [Aiger et al., 2008] and exploits a clever data structure to achieve linear-time performance. In contrast, globally-optimal algorithms [Li and Hartley, 2007; Yang

Figure 4.4: Two partially-overlapping point-sets from the DRAGON-STAND dataset of the Stanford Computer Graphics Laboratory. The point-sets were captured from two different locations and do not overlap completely, making them challenging to register accurately.

et al., 2016] are guaranteed to find the optimal transformation for a particular objective function. While global methods are less susceptible to local optima, they are typically much slower than local optimisation approaches, and may be less robust to outliers or make restrictive assumptions about the point-sets or transformations.

4.3 Support Vector–Parametrised Gaussian Mixtures

The central idea of this work is that the robustness of 2D–2D and 3D–3D registration is dependent on the sensor data representation used. Robustness to structured outliers is of primary concern, because sensor data rarely overlaps completely, such as when an object or scene is sampled from different sensor locations, shown in Figure 4.4. Robustness to sampling artefacts is also very important, because variable sampling densities, noise and occlusions can greatly change the geometry of an object or scene, limiting its invariance to viewpoint. Another consideration is the class of optimisation problem a particular representation admits. Framing registration as a continuous optimisation problem involving continuous density functions can make the registration problem more tractable than the equivalent discrete problem [Jian and Vemuri, 2011].

Consequently, an adaptive, sparse and discriminative sensor data representation is developed, named the Support Vector–parametrised Gaussian Mixture (SVGM). In order to construct an SVGM from a point-set, a discriminative Support Vector Machine (SVM) is trained and then transformed to a Gaussian Mixture Model (GMM). Since an SVM selects a sparse subset of the data points that best classifies the dataset, the representation is compact, adaptive to local surface complexity, and robust to sampling artefacts [Van Nguyen and Porikli, 2013], attributes that persist through to the GMM.

4.3.1 One-Class Support Vector Machine

The output function of an SVM can be used to approximate the surface described by noisy and incomplete point-set data, providing a continuous implicit surface representation [Steinke et al., 2005]. Van Nguyen and Porikli [2013] demonstrated that this representation was robust to noise, fragmentation, missing data and other artefacts for 2D shapes, with the same behaviour expected in 3D. An SVM classifies data by constructing a hyperplane that separates data of two different classes, maximising the margin between the classes while allowing for some mislabelling [Cortes and Vapnik, 1995]. Since point-set data contains only positive samples, a one-class SVM [Schölkopf et al., 2001] can be used to find the hyperplane that maximally separates the data points from the origin or viewpoint in feature space. The training data is mapped to a higher-dimensional feature space, where it may be linearly separable from the origin, with a non-linear kernel function.

The output function $f(\mathbf{p})$ of a one-class SVM is given by

$$f(\mathbf{p}) = \sum_{i=1}^{N} \alpha_i K(\mathbf{p}, \mathbf{p}_i) - \rho \tag{4.1}$$

where $\mathbf{p}_i$ are the point vectors, α_i are the weights, ρ is the bias, N is the number of training samples and K is the kernel function

$$K(\mathbf{p}, \mathbf{p}_i) = \Phi(\mathbf{p}) \cdot \Phi(\mathbf{p}_i) \tag{4.2}$$

that evaluates the inner product of data vectors mapped to a feature space by Φ. In order to map the SVM to a GMM, a Gaussian Radial Basis Function (RBF) kernel [Aizerman et al., 1964] was selected, given by

$$K(\mathbf{p}, \mathbf{p}_i) = \exp\Big(-\gamma \|\mathbf{p} - \mathbf{p}_i\|_2^2\Big) \tag{4.3}$$

where γ is the Gaussian kernel width. The learning problem is formulated as the quadratic program [Schölkopf et al., 2001]

$$\begin{aligned} \min_{\mathbf{w}, \boldsymbol{\xi}, \rho} \quad & \frac{1}{2}\mathbf{w}^\intercal \mathbf{w} - \rho + \frac{1}{\nu N} \sum_{i=1}^{N} \xi_i \\ \text{subject to} \quad & \mathbf{w}^\intercal \Phi(\mathbf{p}_i) \geqslant \rho - \xi_i \\ & \xi_i \geqslant 0 \\ & \text{for } i = 1, \ldots, N \end{aligned} \tag{4.4}$$

where ξ_i are the slack variables and $\mathbf{w}$ is given by

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i \Phi(\mathbf{p}_i). \tag{4.5}$$

This optimisation problem attempts to correctly classify as much of the data as possible, while keeping the model simple and the margin maximal. The formulation has a parameter $\nu \in (0, 1]$ that controls the trade-off between training error and model complexity. It is a lower bound on the fraction of support vectors and an upper bound on the misclassification rate [Schölkopf et al., 2001]. The data points with non-zero weights α_i are the support vectors, with index set $\mathcal{SV} = \{i \in [1, N] \,|\, \alpha_i > 0\}$.

The kernel width γ can be estimated automatically for each point-set by noting that it is inversely proportional to the square of the scale σ. For an $N \times D$ point-set in matrix form $\mathbf{P}$ with mean $\bar{\mathbf{p}}$, the estimated scale $\hat{\sigma}$ is proportional to the $2D^{\text{th}}$ root of the generalised variance [Wilks, 1932], that is

$$\hat{\sigma} \propto \left| \frac{1}{N-1} (\mathbf{P} - \mathbf{1}\bar{\mathbf{p}}^\intercal)^\intercal (\mathbf{P} - \mathbf{1}\bar{\mathbf{p}}^\intercal) \right|^{\frac{1}{2D}}. \tag{4.6}$$

If a training set is available, better performance can be achieved by finding γ using cross-validation on the registration accuracy, searching in the neighbourhood of $1/2\hat{\sigma}^2$. Note that classifier accuracy cannot be used for cross-validation since one-class SVM can trivially classify all points correctly.

The computational complexity of this data representation is polynomial in the number of points N. In this work, the LIBSVM [Chang and Lin, 2011] implementation of one-class SVM is used, which applies a modified version [Fan et al., 2005] of the sequential minimal optimisation algorithm [Platt, 1999] with a time complexity of $\mathcal{O}(N)$ per iteration. The number of iterations was empirically found to be sublinear with respect to N, for the datasets used in this work. Approximate variants can also be used for further reductions in learning time [Joachims, 1999; Tsang et al., 2005].

An example one-class SVM output function with 3401 support vectors, trained using the Stanford Dragon model, is visualised in Figure 4.5. The output function was sampled at every point in a dense 3D grid and inlier points for which $f(\mathbf{p}) \geqslant 0$ were plotted, shaded according to their output value.

4.3.2 Gaussian Mixture Model Transformation

In order to make use of the trained SVM for point-set registration, it must first be approximated as a GMM. The transformation identified by Deselaers et al. [2010] is used to represent the SVM in the framework of a GMM, without altering the decision

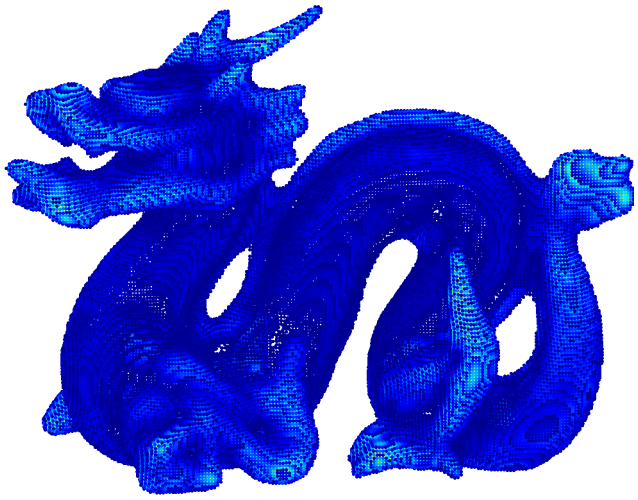


Figure 4.5: One-class SVM inliers, coloured by output function value. A one-class SVM was trained using the Dragon model from the Stanford Computer Graphics Laboratory and the output function was sampled at every point in a dense 3D grid. Points for which the output function was positive (inliers) were plotted, shaded by value (darker colours are more probable).

boundary. A similar principle was used by Schölkopf et al. [1997] to train a Gaussian RBF network with the SVM algorithm.

A GMM converted from an SVM will necessarily optimise classification performance instead of data representation, since SVMs are discriminative models, unlike standard generative methods used to construct GMMs. This allows it to discard redundant data and reduces its susceptibility to sampling artefacts such as varying point densities, which are prevalent in real datasets.

The decision function of an SVM with a Gaussian RBF kernel can be written as

$$r(\mathbf{p}) = \underset{k\in\{-1,1\}}{\arg\max}\left\{\sum_{i\in\mathcal{SV}} k\alpha_i \exp\left(-\gamma\|\mathbf{p}_i - \mathbf{p}\|_2^2\right) - k\rho\right\} \tag{4.7}$$

where k is the class, positive for inliers and negative otherwise. The GMM decision function can be written as

$$r(\mathbf{p}) = \underset{k\in\{-1,1\}}{\arg\max}\left\{\sum_{i=1}^{n_k} p(k)p(i|k)\frac{1}{\left(2\pi\sigma_k^2\right)^{D/2}}\exp\left(-\frac{\|\mathbf{p}-\boldsymbol{\mu}_{ki}\|_2^2}{2\sigma_k^2}\right)\right\} \tag{4.8}$$

where n_k is the number of clusters for class k, $p(k)$ is the prior probability of class k, $p(i|k)$ is the cluster weight of the i^{th} cluster of class k, D is the dataset dimension, and $\boldsymbol{\mu}_{ki}$ and σ_k^2 are the mean and variance of the i^{th} Gaussian component of class k.

Noting the similarity of (4.7) and (4.8), the following mapping can be applied:

$$\boldsymbol{\mu}_{ki} = \begin{cases} \mathbf{p}_i & \text{if } k = +1 \\ \mathbf{0} & \text{else} \end{cases} \tag{4.9}$$

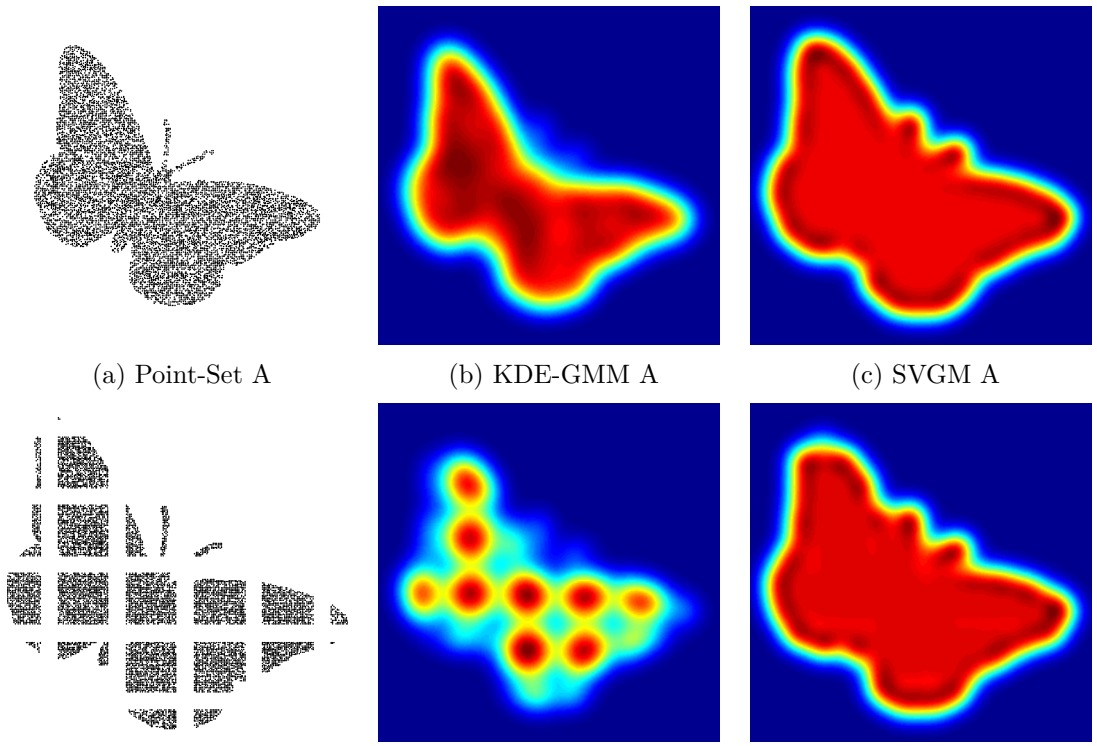
$$\sigma_k^2 = \begin{cases} 1/2\gamma & \text{if } k = +1 \\ \infty & \text{else} \end{cases} \tag{4.10}$$

$$\phi_i = p(k)p(i|k) = \begin{cases} \alpha_i (2\pi\sigma_k^2)^{D/2} & \text{if } k = +1 \\ \rho (2\pi\sigma_k^2)^{D/2} & \text{else} \end{cases} \tag{4.11}$$

for $i \in \mathcal{SV}$, where ϕ_i is the mixture weight, that is, the prior probability of the i^{th} component. The bias term ρ is approximated by an additional density given to the negative class with arbitrary mean, high variance and a cluster weight proportional to ρ. This term is omitted from the registration framework because it does not affect the optimisation. The resulting GMM, named a Support Vector–parametrised Gaussian Mixture (SVGM), has the parameter set $\boldsymbol{\theta} = \{\boldsymbol{\mu}_i,\ \sigma^2,\ \phi_i\}$ for all $i \in \mathcal{SV}$.

The use of GMMs as sensor data representations was discussed in detail in Section 3.3.5. While in this work the GMMs are generated from an SVM, there are many other ways to construct a GMM from point-set data. Kernel Density Estimation (KDE) with identically-weighted Gaussian densities has frequently been used for this purpose, including fixed-bandwidth KDE with isotropic covariances [Jian and Vemuri, 2011; Detry et al., 2009], variable-bandwidth KDE with non-identical covariances [Comaniciu, 2003] and non-isotropic covariance KDE [Xiong et al., 2013a]. While using full covariances may increase the representational power of the model, it can be inefficient and is not necessarily advantageous [Wand and Jones, 1995, p. 107]. The primary disadvantage of these methods is that the number of Gaussian components is typically equal to the point-set size, which can be very large for real-world datasets. In contrast, this work intelligently selects a sparse subset of the data points to locate the Gaussian densities and weights them non-identically. Moreover, an SVM is a discriminative model, which is more robust than generative models to sampling artefacts such as occlusions, missing data and variable sampling densities, as demonstrated in Figure 4.6.

Expectation Maximisation (EM) [Dempster et al., 1977] can also be used to construct a GMM, with fewer components than the KDE method. EM finds the maximum likelihood estimates of the GMM parameters. Unlike an SVM, the number of densities is generally specified a priori for the EM algorithm. Strategies to handle this requirement, such as overestimating the number of components, can be slow and sensitive to initialisation [Scott and Szewczyk, 2001, p. 326].



(a) Point-Set A (b) KDE-GMM A (c) SVGM A

(d) Point-Set B (e) KDE-GMM B (f) SVGM B

Figure 4.6: The effect of significant occlusion on two point-set representations, using the same parameters for both. The SVGM representation is, qualitatively, almost identical when occluded (f) and unoccluded (c), whereas the fixed-bandwidth KDE representation is much less robust to occlusion (e).

4.4 Support Vector Registration

Once the point-sets are in mixture model form, the registration problem can be posed as the problem of minimising a discrepancy measure between mixtures, as shown in Figure 4.7. If the point-sets are well-represented by the Gaussian mixtures, the transformation that aligns the GMMs will correspond to the transformation that aligns the point-sets. As discussed in Section 3.4.6, the L_2 distance between Gaussian mixtures [Jian and Vemuri, 2011] has favourable properties for the geometric alignment problem. It can be expressed in closed-form and efficiently implemented since it avoids numerical approximations of the integral. More critically, it has an estimator that is inherently robust to outliers [Scott, 2001], unlike the maximum likelihood estimator that minimises the Kullback-Leibler divergence. See Section 3.4.6 for a detailed discussion on the robustness of the L_2E estimator that minimises the L_2 distance between probability densities.

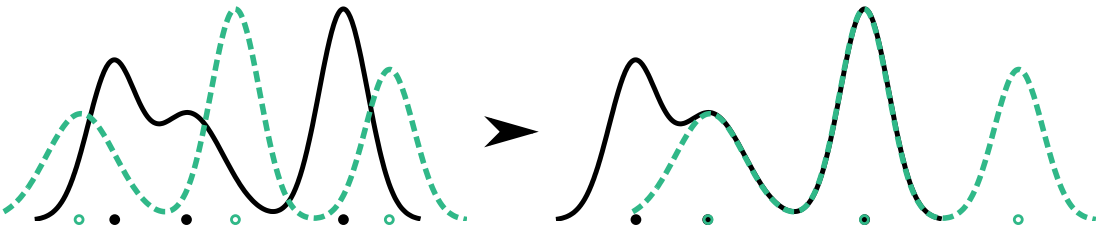


Figure 4.7: Two misaligned 1D Gaussian mixtures (left), generated from partially-overlapping point-sets, are aligned by minimising the distance between mixtures (right).

The objective function for the L_2 distance between Gaussian mixtures (3.75) was derived in Section 3.4.6 for the general case. In this chapter, the Gaussian covariances are constrained to be isotropic and identical, due to the use of an SVM in the learning procedure. This is a standard choice for many GMA approaches that balances the expressiveness of the mixture model against evaluation speed of the objective function. Let $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_{ki}, \sigma_k^2, \phi_{ki}\}_{i \in \mathcal{SV}_k}$ be the parameter set of an n_k-component SVGM with index set $\mathcal{SV}_k$, means $\boldsymbol{\mu}_{ki}$, variances σ_k^2, and mixture weights $\phi_{ki} \geqslant 0$, where $\sum_{i \in \mathcal{SV}_k} \phi_{ki} = 1$. Then the L_2 distance between Gaussian mixtures, up to a constant factor $(2\pi(\sigma_1^2 + \sigma_2^2))^{-n/2}$ and addition by a constant, for a rotation $\mathbf{R} \in SO(n)$ and a translation $\mathbf{t} \in \mathbb{R}^n$ ($n = 2$ or 3) is given by the objective function

$$f(\mathbf{R}, \mathbf{t}) = -\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \phi_{1i} \phi_{2j} \exp\left(-\frac{\left\| \mathbf{R}\boldsymbol{\mu}_{1i} + \mathbf{t} - \boldsymbol{\mu}_{2j} \right\|_2^2}{2(\sigma_1^2 + \sigma_2^2)} \right). \tag{4.12}$$

This can be expressed in the form of a discrete Gauss transform with a computational complexity of $\mathcal{O}(n_1 n_2)$ or a fast Gauss transform [Greengard and Strain, 1991] with a complexity of $\mathcal{O}(n_1 + n_2)$.

The gradient vector is derived in the same way as in Jian and Vemuri [2011]. Let $\mathbf{M}_0 = \left[\boldsymbol{\mu}_{1,1}, \ldots, \boldsymbol{\mu}_{1,n_1}\right]^\mathsf{T}$ be the $n_1 \times n$ matrix of the mean vectors from the GMM parametrised by $\boldsymbol{\theta}_1$ and $\mathbf{M} = T(\mathbf{M}_0, \boldsymbol{\lambda})$ be the matrix after applying a transformation parametrised by $\boldsymbol{\lambda}$. Using the chain rule, the gradient is $\frac{\partial f}{\partial \boldsymbol{\lambda}} = \frac{\partial f}{\partial \mathbf{M}} \frac{\partial \mathbf{M}}{\partial \boldsymbol{\lambda}}$. Let $\mathbf{G} = \frac{\partial f}{\partial \mathbf{M}}$ be an $n_1 \times n$ matrix, which can be found while evaluating the objective function by

$$\mathbf{G}_i = -\frac{1}{\sigma_1^2 + \sigma_2^2} \sum_{j=1}^{m} \Big(\mathbf{R}\boldsymbol{\mu}_{1i} + \mathbf{t} - \boldsymbol{\mu}_{2j} \Big) f_{ij}(\mathbf{R}, \mathbf{t}) \tag{4.13}$$

where $\mathbf{G}_i$ is the i^{th} row of $\mathbf{G}$ and f_{ij} is a summand of f. For a rigid motion, $\mathbf{M} = \mathbf{M}_0\mathbf{R}^\mathsf{T} + \mathbf{1}_{n_1}\mathbf{t}^\mathsf{T}$ where $\mathbf{1}_d$ is a d-dimensional column vector of ones. The gradients with

respect to each motion parameter are given by

$$\frac{\partial f}{\partial \mathbf{t}} = \mathbf{G}^{\mathsf{T}}\mathbf{1}_{n_1} \tag{4.14}$$

$$\frac{\partial f}{\partial r_i} = \mathbf{1}_n^{\mathsf{T}}\Big((\mathbf{G}^{\mathsf{T}}\mathbf{M}_0) \circ \frac{\partial \mathbf{R}}{\partial r_i}\Big)\mathbf{1}_n \tag{4.15}$$

where $\circ$ is the element-wise Hadamard product and r_i are the elements parametrising $\mathbf{R}$: a rotation angle α for 2D rotations and a unit quaternion for 3D rotations.

The objective function is smooth, differentiable and convex in the neighbourhood of the optimal motion parameters and therefore gradient-based numerical optimisation methods can be used, such as nonlinear conjugate gradient or quasi-Newton methods. For this implementation, an interior-reflective Newton method was selected [Coleman and Li, 1996], being time and memory efficient and scaling well with the number of Gaussian components. For the quaternion parametrisation of 3D rotations, the unit-norm constraint was enforced by projecting the quaternion back to the space of valid rotations after each update by normalisation. An alternate formulation using Lagrange multipliers was also implemented, however it converged slightly less frequently than the normalisation method. See Section 3.5 for a more detailed discussion about this constraint and the alternatives that can be used to enforce it.

Although the objective function is locally convex, it is rarely convex over the entire transformation domain. As a result, this approach is susceptible to local optima, as with all local optimisation methods. This is particularly problematic for alignment problems with large motions and 3D data with symmetries or near-symmetries. There are many heuristic approaches that can alleviate this problem. Since a scale parameter is an input to the algorithm, a multi-resolution approach can be adopted. Like simulated annealing, the scale parameter γ is increased at each iteration, with the algorithm initialised by the transformation found at the previous scale. This coarse-to-fine strategy is appropriate because the objective function is smoother for smaller values of γ and approaches the ICP objective function as γ increases. Another strategy is to use random-start local search, initialising the algorithm at randomly sampled points in the transformation domain. However, this can be deployed for any local optimisation algorithm and so was not used to ensure a fair comparison.

The Support Vector Registration (SVR) algorithm is outlined in Algorithm 4.1. The initial rotation and translation are typically the identity rotation and translation unless prior information is available from odometry, GPS or some other source. The training parameters ν and γ can be estimated, using (4.6) for γ, or by cross-validation on a training set. For most applications, the γ value is identical for both point-sets, although this is not mandatory.

Algorithm 4.1 Support Vector Registration (SVR): a robust algorithm for point-set registration using support vector–parametrised Gaussian mixtures.

Input: two point-sets $\mathcal{P}_k = \{\mathbf{p}_{ki}\}_{i=1}^{n_k}$ with $k = 1, 2$; initial rotation $\mathbf{R}_0$; initial translation $\mathbf{t}_0$; initial scale parameters γ_1, γ_2; one-class SVM parameter ν

Output: rotation $\mathbf{R}$ and translation $\mathbf{t}$ such that $\mathcal{P}_1$ transformed by $(\mathbf{R}, \mathbf{t})$ is well-aligned with $\mathcal{P}_2$

1: Initialise rotation and translation: $\mathbf{R} \leftarrow \mathbf{R}_0$, $\mathbf{t} \leftarrow \mathbf{t}_0$
2: **repeat**
3: Train an SVM from each point-set:
$\boldsymbol{\theta}_k^{\text{SVM}} = \{\mathbf{p}_{ki},\ \gamma_k,\ \alpha_{ki}\}_{i \in \mathcal{SV}_k} \leftarrow \text{trainSVM}(\mathcal{P}_k, \gamma_k, \nu)$
4: Map the SVMs to GMMs using (4.9), (4.10) and (4.11):
$\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_{ki},\ \sigma_k^2,\ \phi_{ki}\}_{i \in \mathcal{SV}_k} \leftarrow \text{mapToGMM}\left(\boldsymbol{\theta}_k^{\text{SVM}}\right)$
5: Optimise the objective function $f(\mathbf{R}, \mathbf{t})$ (4.12) using the gradients (4.14), (4.15), and update the transformation parameters: $(\mathbf{R}, \mathbf{t}) \leftarrow \arg\min_{\mathbf{R},\mathbf{t}} f(\mathbf{R}, \mathbf{t})$
6: Anneal the scale parameter: $\gamma \leftarrow \gamma\delta$
7: **until** change in function value or transformation parameters is below a threshold

4.5 Merging Gaussian Mixtures

For the Support Vector–parametrised Gaussian Mixture (SVGM) representation to be useful for applications where each set of sensor data may contain unique information, such as reconstruction and mapping, an efficient method of merging two aligned mixtures is desirable. A naïve approach is to use a weighted sum of the Gaussian mixtures [Deselaers et al., 2010]. However, this generates a mixture with an unnecessarily high number of components with substantial redundancy. Importantly, the probability of regions not observed in both point-sets would decrease, meaning that regions that are often occluded would disappear from the model as more mixtures were merged. While the time-consuming process of sampling from the combined mixture and re-estimating it using expectation maximisation would eliminate redundancy, it would not alleviate the missing data problem. The same disadvantage afflicts faster sample-free variational-Bayes approaches [Bruneau et al., 2010]. Finally, re-estimating an SVGM from samples of the combined mixture or point-sets would circumvent these problems, since the discriminative framework of the SVM is insensitive to higher-density overlapping regions, but this is not time efficient.

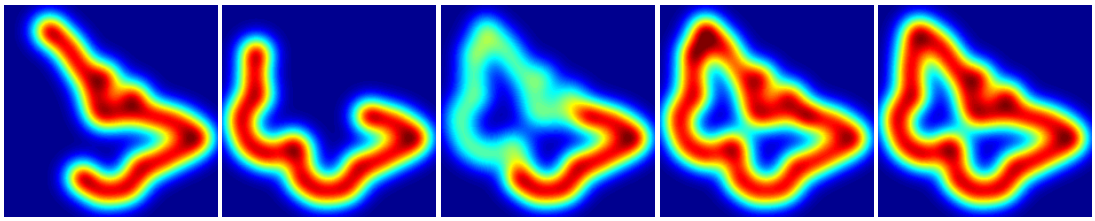
Algorithm 4.2 outlines GMMerge, an efficient algorithm for parsimoniously approximating the merged mixture without weighting the intersection regions disproportionately. Each density of the mixture with parameter set $\boldsymbol{\theta}_1$ is re-weighted using a sparsity-inducing piecewise linear function. The parameter $t \in [0, \infty)$ controls how many densities are added. For $t = 0$, the merged mixture with parameter set $\boldsymbol{\theta}_{12}$ contains only $\boldsymbol{\theta}_2$. As $t \to \infty$, $\boldsymbol{\theta}_{12}$ additionally contains every non-redundant density

Algorithm 4.2 GMMerge: an algorithm for parsimonious Gaussian mixture merging

Input: two aligned mixture models, with parameter sets $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_{ki}, \sigma_k^2, \phi_{ki}\}_{i=1}^{n_k}$, mean vectors $\boldsymbol{\mu}_{ki}$, variances σ_k^2 and mixture weights ϕ_{ki}, and a merging parameter t

Output: merged model $\boldsymbol{\theta}_{12}$

1: Initialise merged model: $\boldsymbol{\theta}_{12} \leftarrow \boldsymbol{\theta}_2$
2: **for** $i = 1, \ldots, n_1$ **do**
3: For the i^{th} density of $\boldsymbol{\theta}_1$, calculate: $\Delta = p(\boldsymbol{\mu}_{1i}|\boldsymbol{\theta}_{1i}) - p(\boldsymbol{\mu}_{1i}|\boldsymbol{\theta}_2)$
4: Update weight using a sparsity-inducing function: $\phi_{1i} \leftarrow \phi_{1i} \max(0, \min(1, t\Delta))$
5: **if** $\phi_{1i} > 0$ **then**
6: Add to merged mixture: $\boldsymbol{\theta}_{12} \leftarrow \boldsymbol{\theta}_{1i} \cdot \boldsymbol{\theta}_{12}$
7: Renormalise the merged mixture $\boldsymbol{\theta}_{12}$



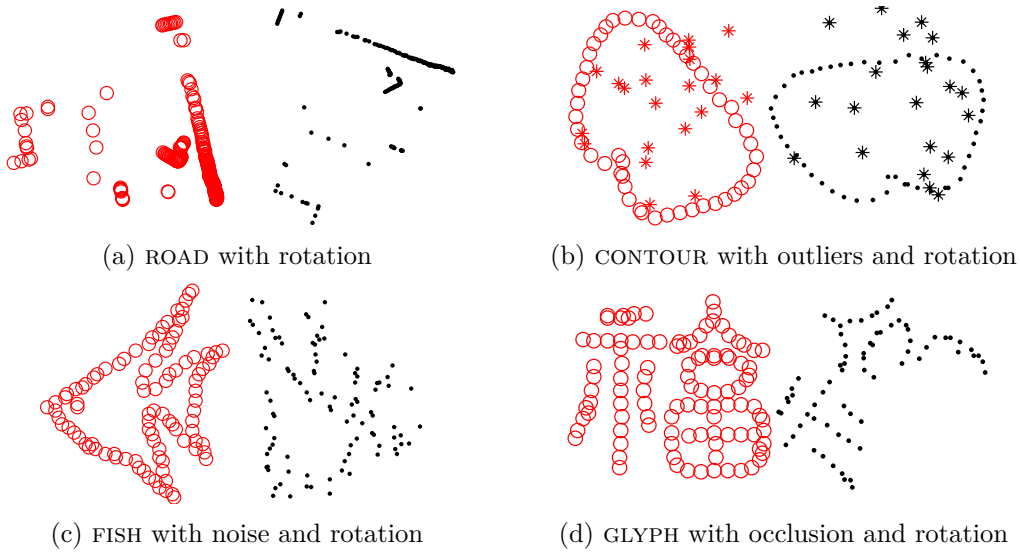
(a) GMM $\boldsymbol{\theta}_1$ (b) GMM $\boldsymbol{\theta}_2$ (c) Naïve merge (d) GMMerge (e) Ground truth

Figure 4.8: Merging aligned Gaussian mixtures (a) and (b) with a naïve weighted sum (c) and GMMerge (d). The mixture produced by GMMerge is almost identical to the ground truth (e), while the naïve approach over-emphasises overlapping regions.

from $\boldsymbol{\theta}_1$. Figure 4.8 shows the SVGM representations of two 2D point-sets, the naïvely merged mixture and the GMMerge mixture.

4.6 Results

The Support Vector Registration (SVR) algorithm was tested using many different point-sets, including synthetic and real datasets in 2D and 3D, at a range of motion scales and outlier, noise and occlusion fractions. In all experiments, the initial rotation and translation parameters $(\mathbf{R}_0, \mathbf{t}_0)$ were the identity rotation and translation, ν was 0.01 and γ was selected by cross-validation, except where otherwise noted. For this, a small subset ($< 5\%$) of the dataset was withheld and the value of γ that achieved the best registration accuracy on this training set was chosen. Parameter values were tested in the neighbourhood of the estimate $\hat{\gamma} = 1/2\hat{\sigma}^2$ (4.6). Only a small subset of the dataset was required because the registration accuracy is not very sensitive to γ, as will be demonstrated. For all benchmark methods, parameters were chosen using a grid search optimising registration accuracy.



(a) ROAD with rotation

(b) CONTOUR with outliers and rotation

(c) FISH with noise and rotation

(d) GLYPH with occlusion and rotation

Figure 4.9: Datasets for 2D registration. A rotation and a range of perturbations are applied to each point-set, including random outliers, Gaussian noise, and occlusion.

4.6.1 2D Registration Experiments

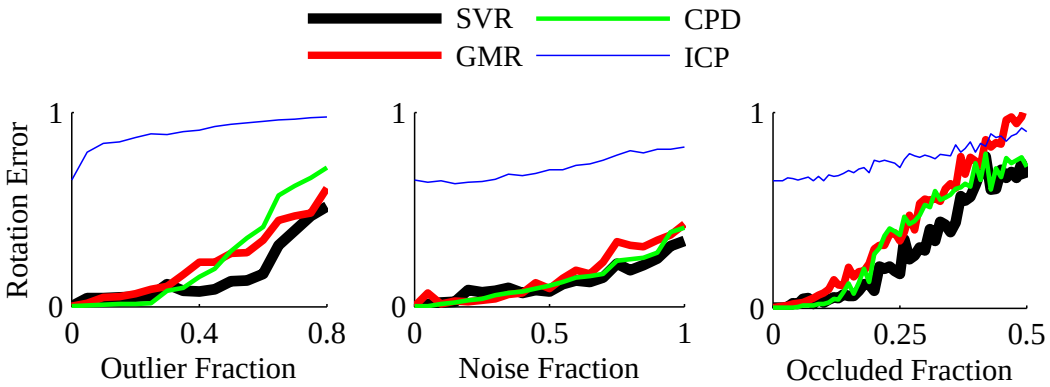
To test the efficacy of SVR for 2D registration, the four point-sets in Figure 4.9 were used: ROAD [Tsin and Kanade, 2004], CONTOUR, FISH and GLYPH [Chui and Rangarajan, 2003]. The point-sets are available at the websites specified in the bibliography. Three benchmark algorithms were chosen: Gaussian Mixture Model Registration (abbreviated to GMR) [Jian and Vemuri, 2011], Coherent Point Drift (CPD) [Myronenko and Song, 2010] and Iterative Closest Point (ICP) [Besl and McKay, 1992]. Annealing was applied for both SVR ($\delta = 10$) and GMR. Note that the advantages of SVR manifest themselves more clearly on denser point-sets than the sparse sets tested here.

The range of motions that attained a correct registration result was tested by rotating the model point-set by $\alpha \in [-3.14, 3.14]$ radians with a step size of 0.01. Table 4.1 reports the range of contiguous initial rotations for which the algorithm converged, chosen as a rotation error $\leqslant 1°$. The results show that SVR has a wider basin of convergence than the other methods, even for sparse point-sets. Better results for GMR were reported in Jian and Vemuri [2011], but were not able to be reproduced in these experiments. This may be attributable to an annealing process that was not specified in the paper.

To test the algorithm's robustness to outliers, additional points were drawn randomly from the uniform distribution and then were concatenated with the model and scene point-sets separately. To avoid bias, the outliers were sampled from the minimum covering circle of the point-set. The motion was fixed to a rotation of 1 radian (57°) and the experiment was repeated 50 times with different outliers each time. The mean

Table 4.1: Rotational convergence range (in radians) for 2D registration. The tested algorithm converged to the correct alignment (rotation error $\leqslant 1°$) when initialised to any rotation within these ranges. The results indicate that SVR has the widest basin of convergence.

Point-Set	SVR	GMR	CPD	ICP
ROAD	**-3.1–3.1**	-3.0–3.0	-1.6–1.6	-0.8–0.8
CONTOUR	**-1.6–1.6**	-1.5–1.5	-1.5–1.5	-0.1–0.1
FISH	**-1.6–1.6**	-1.5–1.5	-1.2–1.3	-0.4–0.5
GLYPH	**-1.6–1.6**	**-1.6–1.6**	-1.6–1.5	-0.4–0.4



(a) Effect of outliers (b) Effect of noise (c) Effect of occlusions

Figure 4.10: Effect of outliers, noise and occlusions for 2D registration. The mean rotation error (in radians) of 50 repetitions is reported for each point-set pair. The results show that SVR is relatively robust to a large range of perturbations commonly found in real data.

rotation error for a range of outlier fractions is shown in Figure 4.10(a) and indicates that the proposed method is more robust to outliers than the others for large outlier fractions. At this rotation, ICP performs poorly even without outliers.

To test for robustness to noise, a noise model was applied to the model point-set by adding Gaussian noise to each point sampled from the distribution $\mathcal{N}(\mathbf{0}, (\lambda\hat{\sigma})^2)$, where λ is the noise fraction and $\hat{\sigma}$ is the estimated generalised standard deviation across the entire point-set (4.6). A fixed rotation of 1 radian was used and the experiment was repeated 50 times, resampling each time. The average rotation error for a range of noise fractions is shown in Figure 4.10(b) and indicates that SVR is comparable to the other GMA methods.

To test for robustness to occlusions, a random seed point was selected and a fraction of the second point-set was removed using a k-nearest neighbour algorithm. A fixed rotation of 1 radian was used and the experiment was repeated 50 times with different seed points. The mean rotation error for a range of occlusion fractions is shown in Figure 4.10(c) and indicates that the algorithm is more robust to occlusion than the other algorithms.

Table 4.2: Number of correctly aligned point-set pairs (out of 30) for a range of relative poses. Mean computation time in seconds is also reported.

	Local				Global	
Pose	**SVR**	**GMR**	**CPD**	**ICP**	**GOI**	**S4P**
±24°	**30**	29	26	28	**30**	29
±48°	**29**	20	18	19	27	24
±72°	16	13	14	13	**18**	17
±96°	4	2	3	1	10	**13**
Runtime	0.2	19.2	5.7	**0.04**	1407	399

4.6.2 3D Registration Experiments

The advantages of SVR are particularly apparent with dense 3D point-sets. For evaluation, the DRAGON-STAND [Curless and Levoy, 2014], AASS-LOOP [Magnusson, 2011] and HANNOVER2 [Wulf, 2011] datasets were used, available at the websites specified in the bibliography. Seven benchmark algorithms were evaluated: GMMReg (abbreviated to GMR) [Jian and Vemuri, 2011], CPD [Myronenko and Song, 2010], ICP [Besl and McKay, 1992], NDT Point-to-Distribution (NDP) [Magnusson et al., 2007] and NDT Distribution-to-Distribution (NDD) [Stoyanov et al., 2012], Globally-Optimal ICP (GOI) [Yang et al., 2013b] and SUPER 4PCS (S4P) [Mellado et al., 2014]. Annealing was used only where indicated.

To evaluate the performance of the algorithm with respect to motion scale, the experiment in Jian and Vemuri [2011] using the DRAGON-STAND dataset was replicated. This dataset contains 15 self-occluding scans of the dragon model acquired by rotating the model in 24° increments on a turntable. All 30 point-set pairs with a relative rotation of ±24° were registered and this was repeated for ±48°, ±72° and ±96°. The criterion for convergence was $\hat{\mathbf{q}} \cdot \mathbf{q} > 0.99$, as specified in Jian and Vemuri [2011], where $\hat{\mathbf{q}}$ and $\mathbf{q}$ are the estimated and ground truth quaternions respectively. In addition to testing the performance with respect to a range of rotations, this experiment also evaluates the algorithm's robustness to increasing levels of occlusion, correlated with the rotation angle. The number of correctly converged registrations is reported in Table 4.2, showing that SVR has a significantly larger basin of convergence than the other local methods and is competitive with the slower global methods. While γ was selected by cross-validation, using the estimate $\hat{\sigma}$ yielded a very similar result. A representative sensitivity analysis for γ and ν is shown in Figure 4.11 for the DRAGON-STAND dataset. It indicates that rotation error is quite insensitive to perturbations in γ and is very insensitive to ν, justifying the choice of fixing this parameter.

To evaluate the robustness of SVR to occlusion, the same procedure was followed as for 2D using the DRAGON-STAND dataset. The mean rotation error (in radians) and the fraction of correctly converged point-set pairs with respect to the fraction of occluded

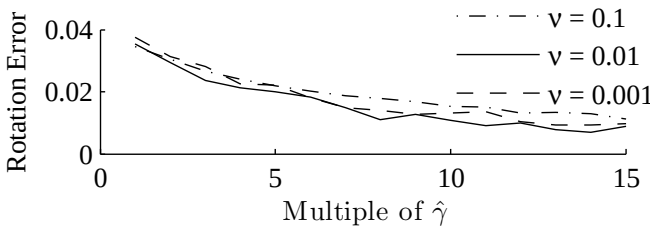
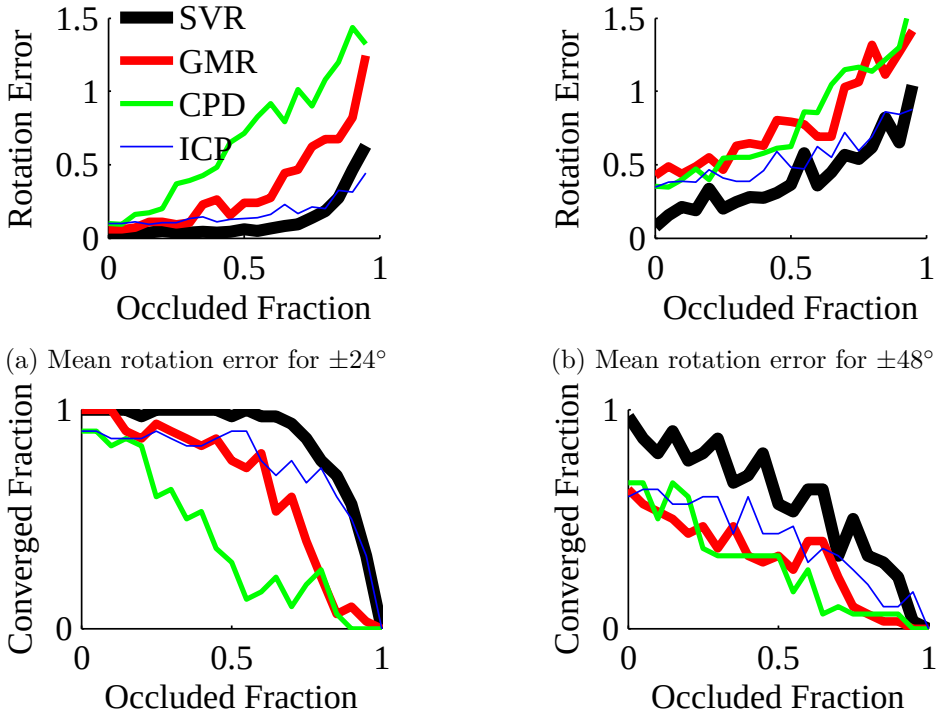


Figure 4.11: Sensitivity analysis for γ and ν. The median rotation error (in radians) for all DRAGON-STAND point-sets with pose differences of ±24° are plotted with respect to multiples of $\hat{\gamma} = 1/2\hat{\sigma}^2$.



(a) Mean rotation error for ±24°

(b) Mean rotation error for ±48°

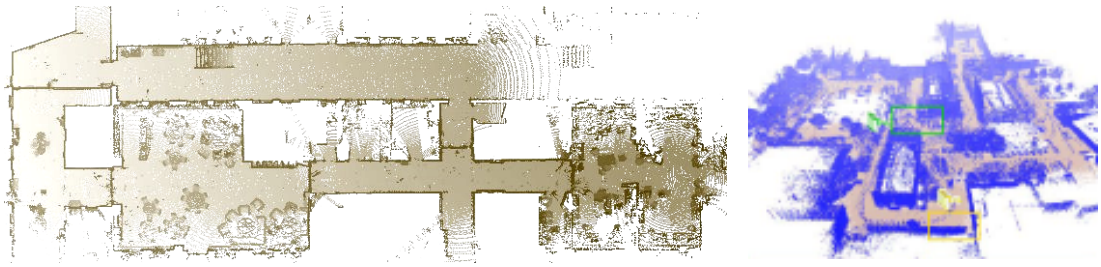
(c) Convergence rate for ±24°

(d) Convergence rate for ±48°

Figure 4.12: Mean rotation error (in radians) and convergence rate of all DRAGON-STAND point-sets with ±24° and ±48° pose differences, with respect to the fraction of occluded points.

points is shown in Figure 4.12, for relative poses of ±24° and ±48°. The results show that SVR is significantly more robust to occlusion than the other methods.

The final experiments evaluated the performance of SVR on two large real-world 3D datasets, shown in Figure 4.13: AASS-LOOP (60 indoor point-sets with ∼13 500 points on average) and HANNOVER2 (923 outdoor point-sets with ∼10 000 points on average), after downsampling using a 0.1 m grid. Both were captured using a laser scanner and ground truth was provided. These are challenging datasets because sequential point-sets overlap incompletely and occluded regions are frequently present. The results for



(a) AASS-LOOP (b) HANNOVER2

Figure 4.13: Aerial views of two large-scale 3D datasets.

Table 4.3: Registration results for AASS-LOOP. While mean translation error (in metres) and rotation error (in radians) are commonly reported, the success rate (translation error $\leqslant 0.5$ m and rotation error $\leqslant 0.2$ radians) is a more useful metric for comparison. The mean computation time (in seconds) is also reported. SVR$^+$ is SVR with annealing.

Metric	SVR	SVR$^+$	GMR	ICP	NDP	NDD	S4P
Translation Error	0.95	**0.67**	1.61	0.99	1.10	0.85	0.71
Rotation Error	0.08	0.06	0.12	0.04	**0.02**	0.06	0.32
Success Rate	81.4	**86.4**	18.6	55.2	50.0	63.8	78.0
Runtime	3.43	29.7	599	10.8	9.12	**1.02**	60.7

Table 4.4: Registration results for HANNOVER2. The mean translation error (in metres), rotation error (in radians), success rate (%) and mean runtime (in seconds) are reported. SVR$^+$ uses annealing.

Metric	SVR	SVR$^+$	GMR	ICP	NDP	NDD	S4P
Translation Error	0.10	**0.09**	1.32	0.43	0.79	0.40	0.40
Rotation Error	**0.01**	**0.01**	0.05	0.05	0.05	0.05	0.03
Success Rate	**99.8**	**99.8**	8.88	74.4	54.2	76.4	75.0
Runtime	14.0	32.6	179	5.68	4.03	**0.51**	39.7

registering adjacent point-sets are shown in Table 4.3 for AASS-LOOP and Table 4.4 for HANNOVER2. The ICP and annealed NDT results are reported directly from Stoyanov et al. [2012] and their criteria for a successful registration is used: translation error $\leqslant 0.5$m and rotation error $\leqslant 0.2$ radians. SVR outperforms the other methods by a significant margin, even more so when annealing ($\delta = 2$) is applied (SVR$^+$).

The mean computation speeds of the experiments, regardless of convergence, are reported in Tables 4.2, 4.3 and 4.4. All experiments were run on a PC with a 3.4 GHz Quad Core CPU and 8 GB of RAM. The SVR code is written in unoptimised MATLAB, except for a cost function in C++, and uses the LIBSVM [Chang and Lin, 2011] library. The benchmarking code was provided by the respective authors, except for ICP, for which a standard MATLAB implementation with kd-tree nearest-neighbour queries was used. For the DRAGON-STAND runtime comparison, all point-sets were randomly downsampled to 2 000 points, because GMR, CPD, GOI and S4P were prohibitively

slow for larger point-sets. In contrast, the entire point-sets were used for the AASS-LOOP and HANNOVER2 runtime comparisons, since the ICP, NDP and NDD methods work better with more points. Hence, these experiments give a good indication of how the algorithms' runtimes scale with the number of points.

4.7 Discussion

The experimental evaluation shows that SVR has a larger region of convergence than the other local optimisation methods and is more robust to sampling artefacts such as occlusions and variable point densities. This is an expected consequence of the SVGM representation, since it is demonstrably robust to these complicating factors, as depicted in Figures 4.3 and 4.6. In addition, the computation time results show that the algorithm scales well with point-set size, unlike GMR and CPD, largely due to the data compression property of the one-class SVM. There is a trade-off, controlled by the parameter γ, between registration accuracy and computation time.

It is important to emphasise the role of the discriminative model in improving robustness to sampling artefacts. A discriminative model is better able to represent the underlying surfaces of a scene than a generative model, because it optimises classification, deciding whether the sampled points are well-classified by a surface. This implicit surface regularises the sampled points while adapting to local surface complexity and does not depend strongly on the point density, since a dense cluster of points can be classified equally well as a sparse set of points in the same location. As a result, the probability density is not biased by regions of high point density and can in-fill small occluded regions. In contrast, generative models optimise representation and therefore model the scene as it was sampled by the sensor. As a result, sampling artefacts such as occlusions and variable point densities are also modelled. This reduces the model's invariance to viewpoint significantly. Moreover, the model is less effective at in-filling occluded regions, since it assigns a lower probability density to the sparser areas adjacent to an occlusion.

This framework for registration and merging can be extended to accurate reconstruction applications. To do so, the one-class SVM may be replaced with a two-class SVM in order to better model the fine details of a scene. As demonstrated by Steinke et al. [2005], an SVM regression algorithm may be used to approximate the signed distance function in the vicinity of the surface. To generate training samples from the negative class (free space), surface points were displaced along their approximated normal vectors by a fixed distance d and then those points that were closer than $0.9d$ to their nearest surface point were discarded [Carr et al., 2001]. The SVGMs constructed using this approach may be fused using GMMerge. However, capturing fine detail is

unnecessary, counter-productive and inefficient for the purpose of registration.

While SVR is a local algorithm, it can still outperform global algorithms on a number of measures, particularly speed, for certain tasks. In Section 4.6.2, SVR was compared to the guaranteed-optimal method Globally-Optimal ICP (GOI) [Yang et al., 2013b] and the faster but non-optimal method Super 4PCS (S4P) [Mellado et al., 2014]. The motion scale results of GOI were comparable to the SVR results, while the average runtime was four orders of magnitude longer. Note that a globally-optimal alignment with respect to the ICP error function is not necessarily the correct alignment for point-sets with missing data or partial overlap. S4P had a more favourable runtime–accuracy trade-off but was nonetheless outperformed by SVR.

A limitation of the SVR algorithm is its time complexity of $\mathcal{O}(n_1 n_2)$ for n_k being the number of Gaussian components in the k^{th} mixture. Therefore the runtime scales quadratically with the number of components. As a result, scenes cannot be modelled to a high resolution without increasing the runtime significantly, increasing the ambiguity of the alignment problem. However, the number of Gaussian components scales with the complexity and size of the scene, not the number of points, helping to mitigate this problem for point-sets with high point-density.

Several strategies can be introduced to reduce the runtime of the algorithm. Firstly, the learning time required to train the one-class SVM can be reduced by using approximate variants of the core algorithm [Joachims, 1999; Tsang et al., 2005]. Secondly, the discrete Gauss transform, which evaluates the sum of Gaussian kernels and underpins the algorithm, has a time complexity of $\mathcal{O}(n_1 n_2)$ and dominates the complexity behaviour of the algorithm as a whole. The time complexity can be reduced to $\mathcal{O}(n_1 + n_2)$ by using an approximation such as the (improved) fast Gauss transform [Greengard and Strain, 1991; Yang et al., 2003]. Alternatively, a data structure can be designed by analogy to the distance transform, storing the set of K least-attenuated Gaussians at each point in $\mathbb{R}^3$ and reducing the time complexity to $\mathcal{O}(K n_1)$. In this formulation, the second mixture could have an arbitrary number of components without affecting the runtime. Finally, the number of iterations could be reduced by applying more sophisticated optimisation techniques that utilise analytically-expressed Hessian matrices.

4.8 Summary

This chapter developed a theoretical framework for robust 2D–2D and 3D–3D registration and merging by solving the Gaussian mixture alignment problem under the L_2 distance for a novel sensor data representation. The Support Vector–parametrised Gaussian Mixture (SVGM) data representation was constructed from a discriminative SVM model and is therefore robust to sampling artefacts, including occlusions and

variable point densities, and is parametrised by a sparse subset of the data points, compressing the data and adapting to local surface complexity. Robustness without over-parametrisation are crucial attributes for efficient and robust registration. The central algorithm, Support Vector Registration (SVR), outperformed state-of-the-art approaches in 2D and 3D rigid registration, exhibiting a larger basin of convergence on challenging datasets, including two large-scale field datasets. In particular, the algorithm was shown to be computationally efficient and robust to structured outliers induced by partial-overlap and occlusion. The GMMerge algorithm complements the registration algorithm by providing a parsimonious and equitable method for merging aligned mixtures, which can subsequently be used as an input to SVR.

A key finding from this work was that a useful local optimisation algorithm for sensor data alignment needs to be very robust to structured outliers and have a wide basin of convergence. A second key finding was that these attributes are strongly dependent on the sensor data representation used, not just the robustness of the objective function. This is a consequence of the many sampling artefacts inherent in sensor data that are not viewpoint invariant. A data representation that models the sampling artefacts instead of the underlying surfaces stymies the alignment of data captured from different viewpoints.

There are several areas that warrant further investigation. Firstly, there is significant scope for optimising the algorithm using approximations such as the improved fast Gauss Transform [Yang et al., 2003] or faster optimisation algorithms that require an analytic Hessian. Secondly, non-rigid registration is a natural extension to this work and should benefit from the robustness of SVR to missing data. It may also be useful to train the SVM with full data-driven covariance matrices [Abe, 2005] and use the full covariances for registration, as in Stoyanov et al. [2012]. Finally, extending the algorithm to global optimality using branch-and-bound would remove the susceptibility of SVR to local optima.

As just prefigured, the following chapter will extend the investigation of robust data representations and objective functions to the globally-optimal 3D–3D geometric alignment problem. There will be some elements in common with this chapter, including the data representation and L_2 objective function, however much of the material is characteristic to the problem. This predominantly stems from the requirement that tight bounds on the objective function be derived so that a branch-and-bound framework can be applied. In addition, a sensible branching strategy and implementation of the bounding functions is needed, so that the algorithm has a feasible runtime.

Chapter 5

Robust and Globally-Optimal 3D–3D Alignment

The focus of this chapter is the geometric alignment of two sets of 3D positional sensor data, such as laser scans, where the data may be corrupted by noise and random or structured outliers. This can be used to solve the problem of estimating the six degrees-of-freedom pose of a 3D sensor with respect to a previously-acquired 3D point-set or the relative pose of two 3D sensors. Algorithms for solving the 3D–3D registration problem have matured over time, progressing from non-robust local optimisation approaches that are susceptible to local minima and outliers, such as the prototypical Iterative Closest Point (ICP) algorithm, to robust local optimisation approaches that widen the basin of convergence and use objective functions that are robust to outliers, such as Gaussian Mixture Alignment (GMA), to a globally-optimal approach that inherits the non-robust ICP objective function, Globally-Optimal ICP (Go-ICP). However, none of these approaches are robust to outliers and immune to local minima. Since typical instances of the 3D–3D registration problem have a large proportion of outliers and many local minima, a useful solver needs to be both robust and global. Globally-optimal approaches have the additional advantage of reliability, providing a guarantee that the solution is a global optimum.

In this chapter, a novel globally-optimal approach is proposed that inherits the robust GMA objective function. The algorithm, named Globally-Optimal Gaussian Mixture Alignment (GOGMA), is the first to find the optimal solution to the 3D rigid Gaussian mixture alignment problem. It improves on the family of GMA approaches by using global optimisation and so not requiring a good pose initialisation. It improves on Go-ICP by using a robust objective function and so being less sensitive to outliers. The approach employs branch-and-bound to search the 6D space of 3D rigid motions $SE(3)$, guaranteeing global optimality without requiring a pose prior. The geometry of $SE(3)$ is used to find novel upper and lower bounds for the objective function and local optimisation is integrated into the scheme to accelerate convergence

without voiding the optimality guarantee. Evaluation on a range of datasets empirically supports the optimality proof and shows that the method performs much more robustly on challenging datasets than existing approaches.

5.1 Introduction

Estimating the 6 degrees-of-freedom alignment of two sets of 3D positional sensor data is the core task for solving the 3D–3D rigid registration problem. A related task is Gaussian Mixture Alignment (GMA), which is the problem of finding the transformation that best aligns one Gaussian mixture with another. It has a natural application to point-set registration, which endeavours to solve the same problem as GMA for discrete point-sets in $\mathbb{R}^n$. Indeed, the Iterative Closest Point (ICP) algorithm [Besl and McKay, 1992; Zhang, 1994] and several other local registration algorithms [Chui and Rangarajan, 2000a,b; Tsin and Kanade, 2004; Myronenko and Song, 2010] can be interpreted as special cases of GMA [Jian and Vemuri, 2011].

Applications of 3D–3D rigid registration include merging multiple partial scans into a complete model [Blais and Levine, 1995; Huber and Hebert, 2003]; using registration results as fitness scores for object recognition [Johnson and Hebert, 1999; Belongie et al., 2002]; registering a view into a global coordinate system for sensor localisation [Nüchter et al., 2007; Pomerleau et al., 2013]; fusing cross-modality data from different sensors [Makela et al., 2002; Zhao et al., 2005]; and finding relative poses between sensors [Yang et al., 2013a; Geiger et al., 2012].

The dominant solution for 3D–3D rigid registration is the ICP algorithm [Besl and McKay, 1992; Zhang, 1994] and variants, due to its conceptual simplicity, ease of use and good performance. However, ICP is limited by its assumption that closest point pairs should correspond, which fails when the point-sets are not coarsely aligned or the moving 'model' point-set is not a proper subset of the static 'scene' point-set. The latter occurs frequently, since differing sensor viewpoints and dynamic objects lead to occlusion and partial-overlap. This closest-point assumption means that ICP is susceptible to missing correspondences, which lead to incorrect data association, and local minima, in which the optimisation gets trapped, producing erroneous estimates without a reliable means of detecting failure, as shown in Figure 4.2.

Gaussian mixture alignment [Chui and Rangarajan, 2000a; Tsin and Kanade, 2004; Jian and Vemuri, 2011; Campbell and Petersson, 2015] mitigates these problems by eschewing explicit correspondences and using a robust objective function. By aligning point-sets without establishing explicit point correspondences, GMA is less sensitive to missing correspondences from partial overlap or occlusion and is less susceptible to local minima, having a wider basin of convergence. Robust objective functions can also

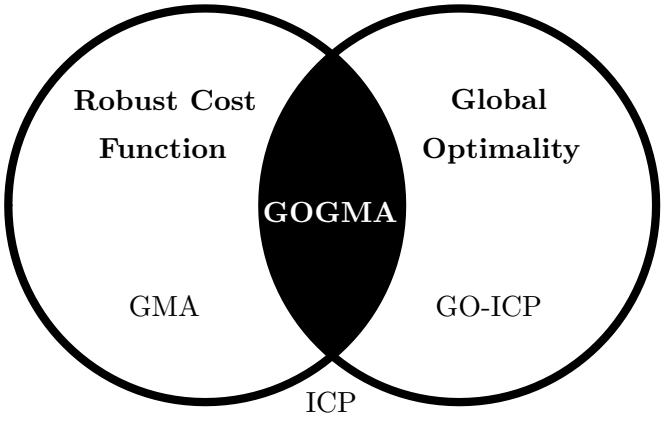


Figure 5.1: Desirable features for a registration algorithm. GOGMA lies in the intersection, being both robust and globally-optimal. Iterative Closest Point (ICP) is non-robust and locally-optimal, such that missing correspondences lead to incorrect data association and optimisation gets trapped in local minima. Gaussian Mixture Alignment (GMA) is robust but locally-optimal, eschewing explicit correspondences and using a robust cost function, but still requiring a good initialisation to converge. Globally-Optimal ICP (GO-ICP) is non-robust but globally-optimal, inheriting the ICP cost function and susceptible to occlusion and partial-overlap.

be applied, such as the L_2 distance between mixtures [Jian and Vemuri, 2011; Campbell and Petersson, 2015]. However, GMA still requires a good initialisation and cannot guarantee global optimality. Moreover, the transformation that aligns the Gaussian mixtures only corresponds to the transformation that aligns the points if the point-sets are well-represented by their Gaussian mixtures.

Existing globally-optimal registration algorithms use branch-and-bound to avoid local minima, but assume translation or correspondences are known. The exception is Go-ICP [Yang et al., 2013b, 2016], which was the first globally-optimal algorithm for the 3D–3D rigid registration problem defined by ICP. Specifically, it used a branch-and-bound approach to find the global minimum of the ICP error metric, the L_2 norm of closest-point residuals. Despite solving the problem of local minima, Go-ICP inherits the non-robust ICP cost function that is susceptible to occlusion and partial overlap. Yang et al. [2016] proposed a trimming strategy to handle outlier correspondences. However, this increased the runtime significantly and required the user to set an outlier fraction parameter that is rarely known in advance.

In this chapter, the first globally-optimal solution is proposed to the 3D Gaussian mixture alignment problem under Euclidean (rigid) transformations. It inherits the robust L_2 density distance objective function of L_2 GMA while avoiding the problem of local minima, as shown in Figure 5.1. The method, named GOGMA, employs the branch-and-bound algorithm to guarantee global optimality regardless of initialisation, using a parametrisation of $SE(3)$ space that facilitates branching. The pivotal contri-

bution is the derivation of the objective function bounds using the geometry of $SE(3)$. In addition, local GMA optimisation is applied whenever the algorithm finds a better transformation, to accelerate convergence without voiding the optimality guarantee.

The chapter is organised as follows: the problem is contextualised by summarising the relevant literature in Section 5.2; a robust GMA objective function for 3D–3D registration is introduced in Section 5.3; a parametrisation of the domain of 3D motions, a branching strategy and a derivation of the bounds are developed in Section 5.4; an algorithm is proposed for globally-optimal Gaussian mixture alignment in Section 5.5; and its performance is evaluated and discussed in Sections 5.6 and 5.7.

5.2 Related Work

The large quantity of work published on ICP, its variants and other local registration techniques precludes a comprehensive list. The reader is directed to the surveys on ICP variants [Rusinkiewicz and Levoy, 2001; Pomerleau et al., 2013] and recent 3D point-set and mesh registration techniques [Tam et al., 2013] for additional background. To improve the robustness of ICP to occlusion and partial overlap, approaches have included trimming [Chetverikov et al., 2005] and outlier rejection [Zhang, 1994]. To enlarge its basin of convergence, approaches have included LM-ICP [Fitzgibbon, 2003], which used the Levenberg–Marquardt algorithm [Moré, 1978] and a distance transform to optimise the ICP error without establishing explicit point correspondences.

The family of Gaussian mixture alignment approaches also sought to improve robustness to poor initialisations, noise and outliers. Notable GMA-related algorithms for rigid and non-rigid registration include Robust Point Matching [Chui and Rangarajan, 2003] that used soft assignment and deterministic annealing, Coherent Point Drift [Myronenko and Song, 2010] and Kernel Correlation [Tsin and Kanade, 2004] that minimised a distance measure between mixtures. The Gaussian Mixture Model Registration algorithm [Jian and Vemuri, 2011] defined an equally-weighted Gaussian at every point in the set with identical and isotropic covariances and minimised the robust L_2 distance between densities. The Normal Distributions Transform algorithm [Magnusson et al., 2007; Stoyanov et al., 2012] defined Gaussians for every cell in a grid and estimated full data-driven covariances. The Support Vector Registration algorithm [Campbell and Petersson, 2015] used an SVM to construct a Gaussian mixture with non-uniform weights that adapts to the structure of the point-set and is robust to occlusion, partial overlap and varying point densities. While more robust than ICP, these methods all employ local optimisation, which is dependent on the initial pose.

There are many heuristic or stochastic methods for global alignment that are not guaranteed to converge. One class utilises stochastic optimisation techniques, such as

particle filtering [Sandhu et al., 2010], particle swarm optimisation [Wachowiak et al., 2004], genetic algorithms [Silva et al., 2005; Robertson and Fisher, 2002] and simulated annealing [Blais and Levine, 1995; Papazov and Burschka, 2011], which often need good initialisations to converge. Another class is feature-based alignment, which exploits the transformation invariance of a local descriptor to build sparse feature correspondences, such as fast point feature histograms [Rusu et al., 2009]. The transformation can be found from the correspondences using random sampling [Rusu et al., 2009], greedy algorithms [Johnson and Hebert, 1999], Hough transforms [Woodford et al., 2014] or branch-and-bound [Gelfand et al., 2005; Bazin et al., 2012]. Super 4PCS [Mellado et al., 2014] is a recent example of a method that uses random sampling without features. It is a four-points congruent sets method [Aiger et al., 2008] that exploits a clever data structure to achieve linear-time performance.

In contrast, globally-optimal techniques avoid local minima by searching the entire transformation space, often using the branch-and-bound paradigm. Existing 3D methods [Li and Hartley, 2007; Olsson et al., 2009; Parra Bustos et al., 2014; Yang et al., 2013b, 2016; Straub et al., 2017] are often very slow or make restrictive assumptions about the point-sets, correspondences or transformations. For example, Li and Hartley [2007] minimised a Lipschitzised L_2 error function using branch-and-bound, but assumed that the point-sets were the same size and the transformation was pure rotation. Olsson et al. [2009] found optimal solutions to point-to-point/line/plane registration using branch-and-bound and bilinear relaxation of rotation quaternions, but assumed correspondences were known. Parra Bustos et al. [2014] achieved efficient run-times using stereographic projection techniques for optimal 3D alignment, but assumed that translation was known. The first globally-optimal algorithm for full 6-DoF 3D–3D rigid alignment without correspondences was proposed by Yang et al. [2016]. The algorithm (Go-ICP) found the optimal solution to the closest point L_2 error between point-sets and was accelerated by using local ICP as a sub-routine. However, it was sensitive to occlusion and partial overlap, due to its non-robust cost function. The proposed trimming strategy went some way to alleviating this, but increased the runtime, required an estimate of the overlap percentage and may lead to ambiguity in the solution. Moreover, the implementation used a distance transform to make the problem tractable. This approximation meant that ϵ-suboptimality could not be guaranteed unless the resolution of the distance transform was sufficiently high. Finally, Straub et al. [2017] proposed a 6-DoF alignment algorithm that decoupled rotation and translation search by first rotationally aligning translation-invariant surface normal distributions, and then aligning Gaussian mixtures to estimate the translation given rotation. Tight bounds on the robust L_2 distance objective function were derived for a rectangular tessellation of translation space $\mathbb{R}^3$ and a near-uniform tetrahedral tessellation of rota-

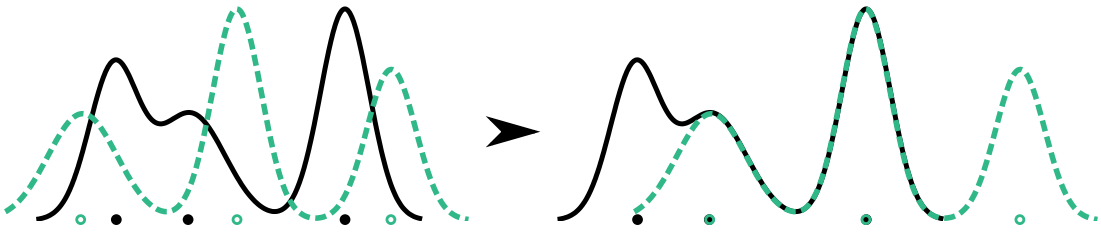


Figure 5.2: Two misaligned 1D Gaussian mixtures (left), generated from partially-overlapping point-sets, are registered with Gaussian mixture alignment (right).

tion space $SO(3)$, which is more efficient to optimise over than angle-axis tessellations. Decoupling rotation and translation search improved the optimisation efficiency significantly, since the complexity scales exponentially in the search space dimension. However, it meant that the solutions for rotation and translation were not jointly optimal, creating alignment failure modes. Moreover, the method required surface normals, which limited the applicability of the algorithm to smoother, densely-sampled surfaces.

5.3 Gaussian Mixture Alignment

The alignment of Gaussian Mixture Models (GMMs) to solve the point-set registration task, as shown in Figure 5.2, is a well-studied problem [Chui and Rangarajan, 2000a; Tsin and Kanade, 2004; Magnusson et al., 2007; Jian and Vemuri, 2011; Campbell and Petersson, 2015]. The use of GMMs as sensor data representations was discussed in detail in Section 3.3.5. They can be generated from point-set data using Kernel Density Estimation (KDE) [Jian and Vemuri, 2011; Detry et al., 2009; Comaniciu, 2003], Expectation Maximisation (EM) [Dempster et al., 1977; Deselaers et al., 2010], Dirichlet Process (DP) estimation [Antoniak, 1974; Straub et al., 2017] or mixture-mapped Support Vector Machines (SVMs) [Campbell and Petersson, 2015].

Once the point-sets are in GMM form, the registration problem can be posed as minimising a discrepancy measure between GMMs. If the point-sets are well-represented by the Gaussian mixtures, the transformation that aligns the GMMs will correspond to the transformation that aligns the point-sets. As discussed in Section 3.4.6, the L_2 distance between Gaussian mixtures [Jian and Vemuri, 2011] has favourable properties for the geometric alignment problem. It can be expressed in closed-form, efficiently implemented and has an estimator that is inherently robust to outliers [Scott, 2001]. See Section 3.4.6 for a detailed discussion on the robustness of the L_2E estimator that minimises the L_2 distance between probability densities.

The objective function for the L_2 distance between Gaussian mixtures (3.75) was derived in Section 3.4.6 for the general case. In this chapter, the Gaussian covariances

are constrained to be isotropic, a standard choice for most GMA approaches. While GMMs with full covariances are more expressive, the bounding functions for full covariance GMMs are much less tractable. Let $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_{ki}, \sigma_{ki}^2, \phi_{ki}\}_{i=1}^{n_k}$ be the parameter set of an n_k-component GMM with means $\boldsymbol{\mu}_{ki}$, variances σ_{ki}^2, and mixture weights $\phi_{ki} \geqslant 0$, where $\sum_{i=1}^{n_k} \phi_{ki} = 1$. Then the L_2 distance between Gaussian mixtures, up to a constant factor $(2\pi)^{-n/2}$ and addition by a constant, for a rotation $\mathbf{R} \in SO(n)$ and a translation $\mathbf{t} \in \mathbb{R}^n$ is given by the objective function

$$f(\mathbf{R}, \mathbf{t}) = -\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \frac{\phi_{1i}\phi_{2j}}{\left(\sigma_{1i}^2 + \sigma_{2j}^2\right)^{\frac{n}{2}}} \exp\left(-\frac{\left(e_{ij}(\mathbf{R}, \mathbf{t})\right)^2}{2\left(\sigma_{1i}^2 + \sigma_{2j}^2\right)}\right) \tag{5.1}$$

where $e_{ij}(\mathbf{R}, \mathbf{t})$ is the pairwise residual error given by

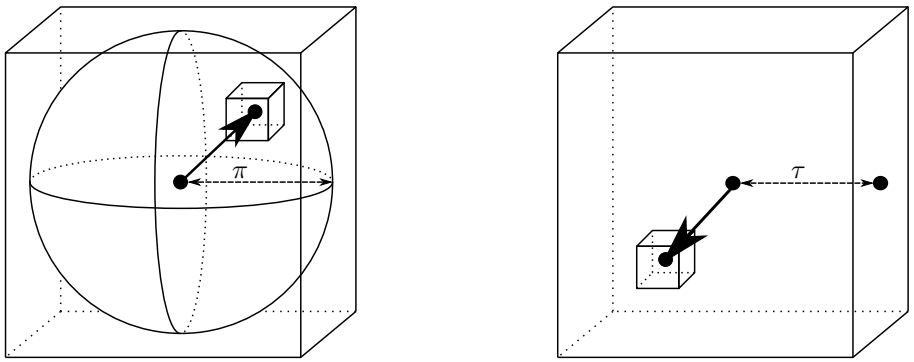
$$e_{ij}(\mathbf{R}, \mathbf{t}) = \left\| \mathbf{R}\boldsymbol{\mu}_{1i} + \mathbf{t} - \boldsymbol{\mu}_{2j} \right\|_2. \tag{5.2}$$

While this objective function can also be used for 2D–2D registration, only 3D–3D registration is considered in this chapter, that is, $n = 3$.

The objective function (5.1) is minimised using a branch-and-bound approach with local optimisation to accelerate convergence. For this, the quasi-Newton L-BFGS-B algorithm [Byrd et al., 1995] was selected, using the closed-form partial derivatives from Campbell and Petersson [2015]. A quaternion parametrisation (see Section 3.1.2) was used for rotation. To enforce the unit-norm constraint, the 4 parameters were allowed to vary within the box constraints $[-1, 1]$ before being projected back to the space of valid rotations by normalising the quaternion [Schmidt and Niemann, 2001].

5.4 Branch-and-Bound

To minimise the highly non-convex Gaussian mixture alignment objective function (5.1), the global optimisation technique of Branch-and-Bound (BB) [Land and Doig, 1960] may be applied. To do so, a suitable means of parametrising and branching (partitioning) the function domain must be found, as well as an efficient way to calculate upper and lower bounds of the function for each branch, which converge as the branch size tends to zero. While the bounds need to be computationally efficient to calculate, the time and memory efficiency of the algorithm also depends on how tight the bounds are, since tighter bounds reduce the search space quicker by allowing sub-optimal branches to be pruned. These two factors are generally in opposition and must be optimised together. Many more details on the branch-and-bound algorithm and its application to the geometric alignment problem can be found in Section 3.7.



(a) Rotation Domain Ω_r (b) Translation Domain Ω_t

Figure 5.3: Parametrisation of $SE(3)$. (a) The rotation space $SO(3)$ is parametrised by angle-axis 3-vectors in a solid radius-π ball. (b) The translation space $\mathbb{R}^3$ is parametrised by 3-vectors bounded by a cube with half side-length τ. The joint domain is branched into sub-hypercubes using a hyperoctree data structure. One sub-hypercube is shown in the figure, depicted as two sub-cubes in the rotation and translation dimensions.

5.4.1 Parametrising and Branching the Domain

To find a globally-optimal solution, the L_2 distance between Gaussian mixtures must be minimised over the domain of 3D motions, that is, the group $SE(3) = SO(3) \times \mathbb{R}^3$. However, the space of these transformations is unbounded. Therefore, to apply the BB paradigm, the space of translations is restricted to be within the bounded set Ω_t, a cube with half side-length τ. Together, these domains form a 6D hypercube, shown as separate 3D cubes in Figure 5.3.

Rotation space $SO(3)$ is minimally parametrised with angle-axis 3-vectors $\mathbf{r}$ with rotation angle $\|\mathbf{r}\|$ and rotation axis $\mathbf{r}/\|\mathbf{r}\|$. The notation $\mathbf{R_r} \in SO(3)$ is used to denote the rotation matrix obtained from the matrix exponential map of the skew-symmetric matrix $[\mathbf{r}]_\times$ induced by $\mathbf{r}$. The Rodrigues' rotation formula (3.10) can be used to efficiently calculate this mapping. See Section 3.1.2 for more details. Using this parametrisation, the space of all 3D rotations can be represented as a solid ball of radius π in $\mathbb{R}^3$. The mapping is one-to-one on the interior of the π-ball and two-to-one on the surface. For ease of manipulation, the 3D cube circumscribing the π-ball is used as the rotation domain Ω_r, as in Li and Hartley [2007]. Translation space $\mathbb{R}^3$ is parametrised with 3-vectors in a bounded domain chosen as the cube Ω_t with half side-length τ. If the GMMs were generated from point-sets scaled to fit within $[-0.5, 0.5]^3$, choosing $\tau = 1$ would ensure that the domain covered every feasible translation. In practice, a smaller value of τ can generally be used, such as 0.5, since the more the point-set bounding boxes overlap, the smaller τ can be without loss of optimality.

In this implementation of BB, the domain is branched into sub-hypercubes using a

hyperoctree data structure. The sub-hypercubes are defined as

$$\mathcal{C} = \mathcal{C}_r(\mathbf{r}_0, \delta_r) \times \mathcal{C}_t(\mathbf{t}_0, \delta_t) \tag{5.3}$$

$$\mathcal{C}_x(\mathbf{x}_0, \delta) = \{\mathbf{x} \in \mathbb{R}^3 \mid \mathbf{e}_i^\mathsf{T}(\mathbf{x} - \mathbf{x}_0) \in [-\delta, \delta], i = [1, 3]\} \tag{5.4}$$

where δ is the half side-length of the cube and $\mathbf{e}_i$ is the i^{th} standard basis vector. To simplify the notation, let $\mathcal{C}_r = \mathcal{C}(\mathbf{r}_0, \delta_r)$ and $\mathcal{C}_t = \mathcal{C}(\mathbf{t}_0, \delta_t)$ for the rotation and translation sub-cubes respectively.

5.4.2 Bounding the Branches

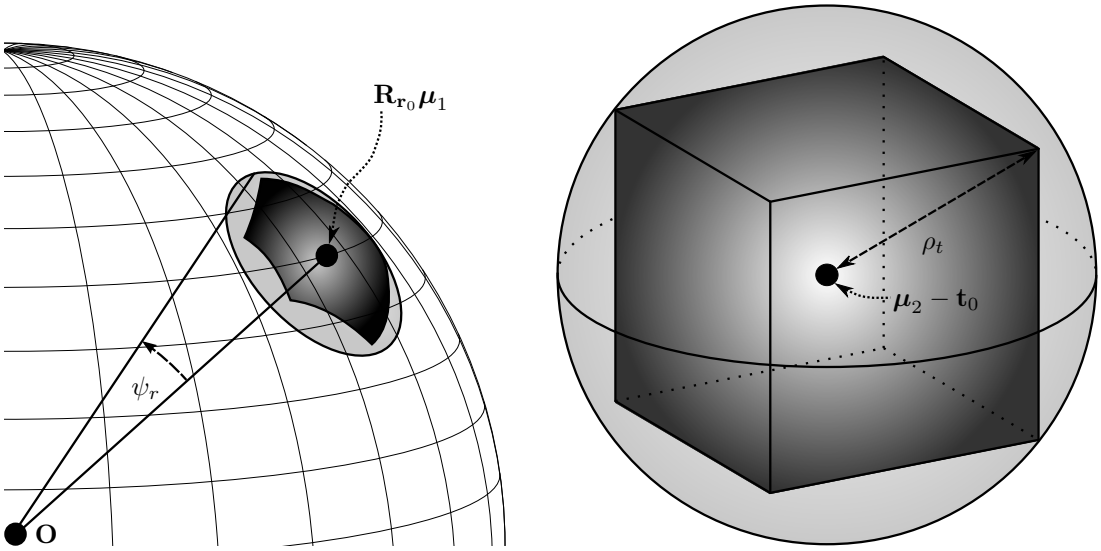
The success of a branch-and-bound algorithm is predicated on the quality of its bounds. For Gaussian mixture alignment, the GMA objective function (5.1) needs to be bounded within a transformation domain $\mathcal{C}_r \times \mathcal{C}_t$. Some preparatory material is now presented.

Uncertainty Bounds

If a branch contained a single rotation or translation, then the new mean vector of a Gaussian transformed by that branch would be known with certainty. However, each branch contains a set of (infinitely) many different rotations or translations. Transforming a Gaussian mean vector by a contiguous set of rotations or translations induces a transformation region, shown for rotation and translation separately in Figure 5.4. The transformed mean vector may lie anywhere in the transformation region, which is the Minkowski sum of an umbrella-shaped spherical patch and a cube for rotation and translation dimensions respectively.

To bound the objective function on a branch, the pairwise residual errors given by $e_{ij}(\mathbf{R}, \mathbf{t}) = \|\mathbf{R}\boldsymbol{\mu}_{1i} + \mathbf{t} - \boldsymbol{\mu}_{2j}\|$ need to be bounded. This residual is the Euclidean distance between a transformed mean vector from one Gaussian mixture and a mean vector from the other Gaussian mixture. To find a bound on this residual, the rotation and translation uncertainty need to be bounded. An upper bound on the rotation uncertainty can be given by the uncertainty angle ψ_r, shown in Figure 5.4(a), and an upper bound on the translation uncertainty can be given by the uncertainty distance ρ_t, shown in Figure 5.4(b).

The rotation uncertainty angle is the angle by which a vector rotated by $\mathbf{R}_{\mathbf{r}_0}$ may differ from that vector rotated by $\mathbf{R}_\mathbf{r}$ for $\mathbf{r} \in \mathcal{C}_r$. To bound the uncertainty angle due to rotation, Lemmas 1 and 2 from Hartley and Kahl [2009] are used. For reference, the relevant parts are merged into Lemma 5.1. The lemma indicates that the angle between two rotated vectors is less than or equal to the Euclidean distance between their rotations' angle-axis representations in $\mathbb{R}^3$.



(a) Rotation Transformation Region (b) Translation Transformation Region

Figure 5.4: Transformation region induced by hypercube $\mathcal{C} = \mathcal{C}_r \times \mathcal{C}_t$, shown for rotation and translation separately. (a) Rotation transformation region for $\mathcal{C}_r$ with centre $\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_1$. The optimal rotation of $\boldsymbol{\mu}_1$ may be anywhere within the heavily-shaded umbrella-shaped transformation region, which is entirely contained by the lightly-shaded spherical cap uncertainty region defined by the mean vector $\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_1$ and the aperture angle $\psi_r(\mathcal{C}_r)$. The angle $\psi_r(\mathcal{C}_r)$ is an upper bound on the rotation uncertainty. (b) Translation transformation region for $\mathcal{C}_t$ with centre $\boldsymbol{\mu}_2 - \mathbf{t}_0$. The optimal translation of $\boldsymbol{\mu}_2$ may be anywhere within the cubic transformation region, which is entirely contained by the translation uncertainty region, a circumscribed sphere with radius $\rho_t(\mathcal{C}_t)$. The distance $\rho_t(\mathcal{C}_t)$ is an upper bound on the translation uncertainty.

Lemma 5.1. *For an arbitrary vector* $\mathbf{p}$ *and two rotations, represented as* $\mathbf{R}_{\mathbf{r}_1}$ *and* $\mathbf{R}_{\mathbf{r}_2}$ *in matrix form and* $\mathbf{r}_1$ *and* $\mathbf{r}_2$ *in angle-axis form,*

$$\angle(\mathbf{R}_{\mathbf{r}_1}\mathbf{p},\, \mathbf{R}_{\mathbf{r}_2}\mathbf{p}) \leqslant \|\mathbf{r}_1 - \mathbf{r}_2\|. \tag{5.5}$$

From this, the maximum angle between a mean vector $\boldsymbol{\mu}$ rotated by $\mathbf{r}_0$ and $\boldsymbol{\mu}$ rotated by $\mathbf{r} \in \mathcal{C}_r$, for a cube of rotation angle-axis vectors $\mathcal{C}_r$, can be found. This upper bound on the rotation uncertainty angle, also from Hartley and Kahl [2009], is reproduced here.

Lemma 5.2. *(Rotation uncertainty angle) Given a 3D point vector* $\boldsymbol{\mu}$ *and a rotation cube* $\mathcal{C}_r$ *of half side-length* δ_r *centred at* $\mathbf{r}_0$*, then* $\forall \mathbf{r} \in \mathcal{C}_r$*,*

$$\angle(\mathbf{R}_{\mathbf{r}}\boldsymbol{\mu}, \mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}) \leqslant \min\{\sqrt{3}\delta_r, \pi\} \triangleq \psi_r(\mathcal{C}_r). \tag{5.6}$$

Proof. Inequality (5.6) can be derived as follows:

$$\angle(\mathbf{R}_\mathbf{r}\boldsymbol{\mu}, \mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}) \leqslant \min\{\|\mathbf{r} - \mathbf{r}_0\|, \pi\} \tag{5.7}$$
$$\leqslant \min\{\sqrt{3}\delta_r, \pi\} \tag{5.8}$$

where (5.7) follows from Lemma 5.1 and the maximum possible angle between point vectors and (5.8) follows from $\max \|\mathbf{r} - \mathbf{r}_0\| = \sqrt{3}\delta_r$, the half space diagonal of the rotation cube, for $\mathbf{r} \in \mathcal{C}_r$. □

The translation uncertainty distance is the distance by which a vector translated by $\mathbf{t}_0$ may differ from that vector translated by $\mathbf{t}$ for $\mathbf{t} \in \mathcal{C}_t$. To bound the uncertainty distance due to translation, the translation cube is enclosed within a circumsphere of radius ρ_t, shown in Figure 5.4(b). From this, the maximum distance between a mean vector $\boldsymbol{\mu}$ translated by $\mathbf{t}_0$ and $\boldsymbol{\mu}$ translated by $\mathbf{t} \in \mathcal{C}_t$, for a cube of translation vectors $\mathcal{C}_t$, can be found. This upper bound on the translation uncertainty distance, also used in Yang et al. [2016], is given in Lemma 5.3.

Lemma 5.3. *(Translation uncertainty distance) Given a 3D point vector $\boldsymbol{\mu}$ and a translation cube $\mathcal{C}_t$ of half side-length δ_t centred at $\mathbf{t}_0$, then $\forall \mathbf{t} \in \mathcal{C}_t$,*

$$\|(\boldsymbol{\mu} - \mathbf{t}) - (\boldsymbol{\mu} - \mathbf{t}_0)\| \leqslant \sqrt{3}\delta_t \triangleq \rho_t(\mathcal{C}_t). \tag{5.9}$$

Proof. Inequality (5.9) can be derived as follows:

$$\|(\boldsymbol{\mu} - \mathbf{t}) - (\boldsymbol{\mu} - \mathbf{t}_0)\| = \|\mathbf{t} - \mathbf{t}_0\| \tag{5.10}$$
$$\leqslant \max_{\mathbf{t} \in \mathcal{C}_t} \|\mathbf{t} - \mathbf{t}_0\| \tag{5.11}$$
$$= \sqrt{3}\delta_t \tag{5.12}$$

where (5.12) is the half space diagonal of the translation cube $\mathcal{C}_t$. □

Objective Function Bounds

As a first step towards bounding the GMA objective function (5.1), the preceding lemmas are used to bound the minimum pairwise residual error $e_{ij}(\mathbf{R}_\mathbf{r}, \mathbf{t})$ within a transformation domain $\mathcal{C}_r \times \mathcal{C}_t$. The pairwise residual error is the L_2 distance between the Gaussian means $\mathbf{R}_\mathbf{r}\boldsymbol{\mu}_{1i}$ and $\boldsymbol{\mu}_{2j} - \mathbf{t}$. An upper bound on the minimum error can be found by evaluating the function at any transformation in the branch. In this case, the transformation at the centre of the rotation and translation cubes is convenient and quick to evaluate.

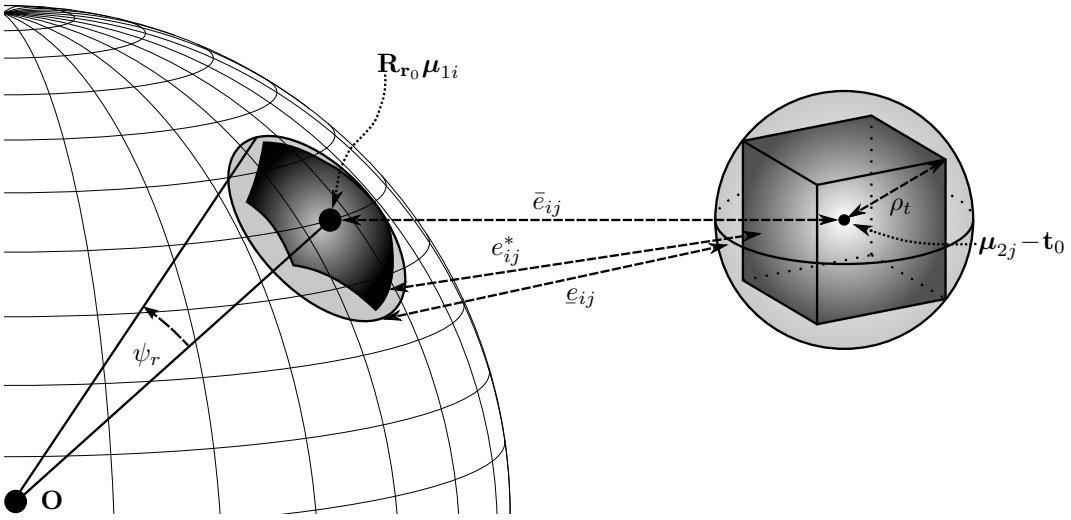


Figure 5.5: Bounding the minimum pairwise residual error. The minimum pairwise residual error $e^*_{ij} = \min_{\mathbf{r}\in\mathcal{C}_r,\mathbf{t}\in\mathcal{C}_t} e_{ij}(\mathbf{R}_\mathbf{r}, \mathbf{t})$ is the minimum distance between the umbrella-shaped rotation and cubic translation transformation regions. It is bounded above by the distance $\bar{e}_{ij}$ between the centres of the spherical cap and the sphere and is bounded below by the minimum distance $\underline{e}_{ij}$ between the spherical cap and the sphere. That is, $\underline{e}_{ij} \leqslant e^*_{ij} \leqslant \bar{e}_{ij}$.

Theorem 5.1. *(Upper bound of the minimum pairwise residual error) For the transformation domain $\mathcal{C}_r \times \mathcal{C}_t$ centred at $(\mathbf{r}_0, \mathbf{t}_0)$, the upper bound of the minimum pairwise residual error can be chosen as*

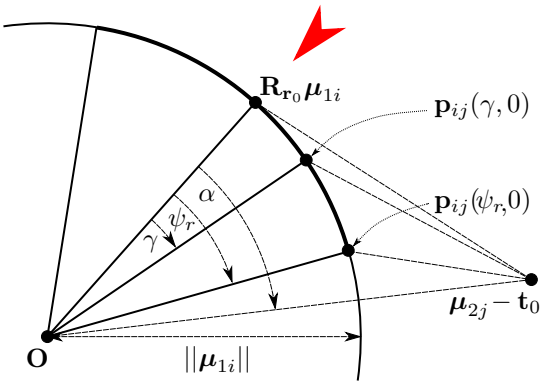
$$\bar{e}_{ij} \triangleq e_{ij}(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}_0). \tag{5.13}$$

Proof. The validity of the upper bound follows from

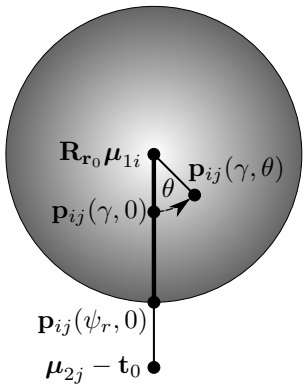
$$\min_{\substack{\mathbf{r}\in\mathcal{C}_r \\ \mathbf{t}\in\mathcal{C}_t}} e_{ij}(\mathbf{R}_\mathbf{r}, \mathbf{t}) \leqslant e_{ij}(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}_0). \tag{5.14}$$

That is, the minimum within the domain is less than or equal to the function value at a specific point within the domain. □

A lower bound on the minimum pairwise residual error within a transformation domain $\mathcal{C}_r \times \mathcal{C}_t$ can be found using the bounds on the rotation and translation uncertainties, the angle ψ_r and the distance ρ_t respectively. The geometric intuition is that the minimum distance between the umbrella-shaped rotation transformation region and the cubic translation transformation region is greater than the minimum distance between the spherical cap rotation uncertainty region and spherical translation uncertainty region, as shown in Figure 5.5.



(a) Viewpoint A: The point $\mathbf{p}_{ij}(\gamma, 0)$ lies on the spherical cap and is defined as the rotation of the vector $\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i}$ about the origin towards $(\boldsymbol{\mu}_{2j} - \mathbf{t}_0)$ by an angle $\gamma \in [0, \psi_r]$. All points in the diagram are coplanar.



(b) Viewpoint B: the point $\mathbf{p}_{ij}(\gamma, \theta)$ lies on the spherical cap and is defined as the rotation of the vector $\mathbf{p}_{ij}(\gamma, 0)$ about the axis $\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i}$ by an angle $\theta \in [0, 2\pi)$. All points in the diagram other than $\mathbf{p}_{ij}(\gamma, \theta)$ are coplanar.

Figure 5.6: Defining the set of points on a spherical cap as a function of two angles γ and θ. As the angles vary within their bounds, the function $\mathbf{p}_{ij}(\gamma, \theta)$ generates a set of points that lie on the spherical cap. For $\theta = 0$, the points $\mathbf{p}_{ij}$ are coplanar and lie on an arc of a great circle of the sphere. (a) The spherical cap (the bold curve) is viewed in the direction orthogonal to the plane defined by points $\{\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i}, \boldsymbol{\mu}_{2j} - \mathbf{t}_0, \mathbf{O}\}$. (b) The spherical cap (the grey shaded circle) is viewed in the direction of the large red arrow in (a).

A critical component of the lower bound is finding the minimum distance between a point in $\mathbb{R}^3$ and a spherical cap. The set of points on a spherical cap can be expressed as a function $\mathbf{p}_{ij}(\gamma, \theta)$ of two angles $\gamma \in [0, \psi_r]$ and $\theta \in [0, 2\pi)$ by first rotating the point vector $\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i}$ about the origin towards $(\boldsymbol{\mu}_{2j} - \mathbf{t}_0)$ by an angle γ and then rotating this intermediate vector about the axis $\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i}$ by an angle θ, as shown in Figure 5.6. If the origin $\mathbf{O}$, the centre of the spherical cap $\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i}$ and $(\boldsymbol{\mu}_{2j} - \mathbf{t}_0)$ are collinear, the first rotation can be toward any point. While this is not the most natural definition of a spherical cap, it creates a pairwise coordinate system that is useful for the proof. The minimum distance between a point and a spherical cap is given in Lemma 5.4.

Lemma 5.4. *(Spherical cap distance) For the spherical cap defined by the point function* $\mathbf{p}_{ij}(\gamma, \theta)$ *for* $\gamma \in [0, \psi_r]$ *and* $\theta \in [0, 2\pi)$*, the minimum distance from a point* $(\boldsymbol{\mu}_{2j} - \mathbf{t}_0)$ *to the spherical cap is given by*

$$\min_{\gamma,\theta} \|\mathbf{p}_{ij}(\gamma, \theta) - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\| = \begin{cases} \big|\|\boldsymbol{\mu}_{1i}\| - \|\boldsymbol{\mu}_{2j} - \mathbf{t}_0\|\big| & \text{for } \alpha \leqslant \psi_r(\mathcal{C}_r) \\ \|\mathbf{p}_{ij}(\psi_r(\mathcal{C}_r), 0) - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\| & \text{for } \alpha > \psi_r(\mathcal{C}_r) \end{cases} \tag{5.15}$$

where the angle $\psi_r(\mathcal{C}_r)$ was given in Lemma 5.2, the angle α is given by

$$\alpha = \angle\Big(\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i}, \boldsymbol{\mu}_{2j} - \mathbf{t}_0\Big) = \arccos \frac{(\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i}) \cdot (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)}{\|\boldsymbol{\mu}_{1i}\|\|\boldsymbol{\mu}_{2j} - \mathbf{t}_0\|} \tag{5.16}$$

and the distance $\|\mathbf{p}_{ij}(\psi_r, 0) - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\|$ is given by

$$\|\mathbf{p}_{ij}(\psi_r, 0) - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\| = \sqrt{\|\boldsymbol{\mu}_{1i}\|^2 + \|\boldsymbol{\mu}_{2j} - \mathbf{t}_0\|^2 - 2\cos(\alpha - \psi_r)\|\boldsymbol{\mu}_{1i}\|\|\boldsymbol{\mu}_{2j} - \mathbf{t}_0\|}. \tag{5.17}$$

Proof. An arbitrary point $\mathbf{p}_{ij}(\gamma, \theta)$ on the spherical cap can be expressed as the rotation of the point $\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i}$ about the origin towards $(\boldsymbol{\mu}_{2j} - \mathbf{t}_0)$ by an angle γ, followed by a rotation of this intermediate vector (denoted $\mathbf{p}_{ij}(\gamma, 0)$) about the axis $\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i}$ by an angle θ. To simplify the derivation, let $\boldsymbol{\mu}^0_{1i} = \mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i}$ and $\boldsymbol{\mu}^0_{2j} = \boldsymbol{\mu}_{2j} - \mathbf{t}_0$. The first axis of rotation, perpendicular to the plane formed by $\boldsymbol{\mu}^0_{1i}$ and $\boldsymbol{\mu}^0_{2j}$, is given by

$$\hat{\mathbf{u}} = \frac{\boldsymbol{\mu}^0_{1i} \times \boldsymbol{\mu}^0_{2j}}{\|\boldsymbol{\mu}_{1i}\|\|\boldsymbol{\mu}^0_{2j}\| \sin\alpha}. \tag{5.18}$$

Therefore, by the Rodrigues' rotation formula (3.10),

$$\begin{aligned}
\mathbf{p}_{ij}(\gamma, 0) &= \cos\gamma\, \boldsymbol{\mu}^0_{1i} + \sin\gamma\, \hat{\mathbf{u}} \times \boldsymbol{\mu}^0_{1i} + (1 - \cos\gamma)(\hat{\mathbf{u}} \cdot \boldsymbol{\mu}^0_{1i})\hat{\mathbf{u}} && (5.19)\\
&= \cos\gamma\, \boldsymbol{\mu}^0_{1i} + \sin\gamma\, \hat{\mathbf{u}} \times \boldsymbol{\mu}^0_{1i} && (5.20)\\
&= \cos\gamma\, \boldsymbol{\mu}^0_{1i} + \sin\gamma \frac{(\boldsymbol{\mu}^0_{1i} \cdot \boldsymbol{\mu}^0_{1i})\boldsymbol{\mu}^0_{2j} - (\boldsymbol{\mu}^0_{1i} \cdot \boldsymbol{\mu}^0_{2j})\boldsymbol{\mu}^0_{1i}}{\|\boldsymbol{\mu}_{1i}\|\|\boldsymbol{\mu}^0_{2j}\| \sin\alpha} && (5.21)\\
&= \|\boldsymbol{\mu}_{1i}\|\left(\frac{\sin(\alpha - \gamma)}{\sin\alpha}\frac{\boldsymbol{\mu}^0_{1i}}{\|\boldsymbol{\mu}_{1i}\|} + \frac{\sin\gamma}{\sin\alpha}\frac{\boldsymbol{\mu}^0_{2j}}{\|\boldsymbol{\mu}^0_{2j}\|}\right) && (5.22)
\end{aligned}$$

where (5.20) follows, after substituting in (5.18), from the result that the scalar triple product is zero if any two vectors involved are equal, (5.21) follows from a vector triple product identity and (5.22) follows by expanding, simplifying and using $\boldsymbol{\mu}^0_{1i} \cdot \boldsymbol{\mu}^0_{2j} = \|\boldsymbol{\mu}_{1i}\|\|\boldsymbol{\mu}^0_{2j}\| \cos\alpha$. Rotating about the second axis of rotation $\boldsymbol{\mu}^0_{1i}$ by an angle θ using the Rodrigues' rotation formula gives

$$\begin{aligned}
\mathbf{p}_{ij}(\gamma, \theta) &= \cos\theta\, \mathbf{p}_{ij}(\gamma, 0) + \sin\theta\, \frac{\boldsymbol{\mu}^0_{1i} \times \mathbf{p}_{ij}(\gamma, 0)}{\|\boldsymbol{\mu}_{1i}\|} + (1 - \cos\theta)\frac{\boldsymbol{\mu}^0_{1i} \cdot \mathbf{p}_{ij}(\gamma, 0)}{\|\boldsymbol{\mu}_{1i}\|}\frac{\boldsymbol{\mu}^0_{1i}}{\|\boldsymbol{\mu}_{1i}\|} && (5.23)\\
&= \|\boldsymbol{\mu}_{1i}\|\left(\left(\cos\gamma - \frac{\cos\alpha \sin\gamma \cos\theta}{\sin\alpha}\right)\frac{\boldsymbol{\mu}^0_{1i}}{\|\boldsymbol{\mu}_{1i}\|} + \frac{\sin\gamma \sin\theta}{\sin\alpha}\frac{\boldsymbol{\mu}^0_{1i} \times \boldsymbol{\mu}^0_{2j}}{\|\boldsymbol{\mu}_{1i}\|\|\boldsymbol{\mu}^0_{2j}\|}\right.\\
&\qquad \left. + \frac{\sin\gamma \cos\theta}{\sin\alpha}\frac{\boldsymbol{\mu}^0_{2j}}{\|\boldsymbol{\mu}^0_{2j}\|}\right) && (5.24)
\end{aligned}$$

where (5.24) follows from substituting in (5.22), expanding and simplifying. Now, the

squared distance between point $\boldsymbol{\mu}_{2j}^0$ and an arbitrary point $\mathbf{p}_{ij}(\gamma,\theta)$ on the spherical cap is given by

$$\begin{aligned}
\|\mathbf{p}_{ij}(\gamma,\theta)-\boldsymbol{\mu}_{2j}^0\|^2 &= (\mathbf{p}_{ij}(\gamma,\theta)-\boldsymbol{\mu}_{2j}^0)\cdot(\mathbf{p}_{ij}(\gamma,\theta)-\boldsymbol{\mu}_{2j}^0) && (5.25)\\
&= \mathbf{p}_{ij}(\gamma,\theta)\cdot\mathbf{p}_{ij}(\gamma,\theta)+\boldsymbol{\mu}_{2j}^0\cdot\boldsymbol{\mu}_{2j}^0-2\mathbf{p}_{ij}(\gamma,\theta)\cdot\boldsymbol{\mu}_{2j}^0 && (5.26)\\
&= \|\boldsymbol{\mu}_{1i}\|^2+\|\boldsymbol{\mu}_{2j}^0\|^2-2\left(\cos\gamma-\frac{\cos\alpha\sin\gamma\cos\theta}{\sin\alpha}\right)\boldsymbol{\mu}_{1i}^0\cdot\boldsymbol{\mu}_{2j}^0\\
&\qquad -2\frac{\sin\gamma\cos\theta}{\sin\alpha}\frac{\|\boldsymbol{\mu}_{1i}\|}{\|\boldsymbol{\mu}_{2j}^0\|}\boldsymbol{\mu}_{2j}^0\cdot\boldsymbol{\mu}_{2j}^0 && (5.27)\\
&= \|\boldsymbol{\mu}_{1i}\|^2+\|\boldsymbol{\mu}_{2j}^0\|^2-2\left(\cos\alpha\cos\gamma-\frac{\cos^2\alpha\sin\gamma\cos\theta}{\sin\alpha}\right)\|\boldsymbol{\mu}_{1i}\|\|\boldsymbol{\mu}_{2j}^0\|\\
&\qquad -2\frac{\sin\gamma\cos\theta}{\sin\alpha}\|\boldsymbol{\mu}_{1i}\|\|\boldsymbol{\mu}_{2j}^0\| && (5.28)\\
&= \|\boldsymbol{\mu}_{1i}\|^2+\|\boldsymbol{\mu}_{2j}^0\|^2-2(\cos\alpha\cos\gamma+\sin\alpha\sin\gamma\cos\theta)\|\boldsymbol{\mu}_{1i}\|\|\boldsymbol{\mu}_{2j}^0\| && (5.29)
\end{aligned}$$

where (5.27) follows from substituting in (5.24) and noting that the scalar triple product is zero if any two vectors involved are equal and (5.29) follows from the identity $\cos^2\alpha = 1-\sin^2\alpha$. The squared distance is minimised when $\theta = 0$ and is given by

$$\min_{\theta}\|\mathbf{p}_{ij}(\gamma,\theta)-\boldsymbol{\mu}_{2j}^0\|^2 = \|\boldsymbol{\mu}_{1i}\|^2+\|\boldsymbol{\mu}_{2j}^0\|^2-2\cos(\alpha-\gamma)\|\boldsymbol{\mu}_{1i}\|\|\boldsymbol{\mu}_{2j}^0\|. \tag{5.30}$$

When $\alpha \leqslant \psi_r$ (Case 1), equation (5.30) is minimised when $\gamma = \alpha$, giving

$$\min_{\gamma,\theta}\|\mathbf{p}_{ij}(\gamma,\theta)-\boldsymbol{\mu}_{2j}^0\|^2 = \left(\|\boldsymbol{\mu}_{1i}\|-\|\boldsymbol{\mu}_{2j}^0\|\right)^2 \tag{5.31}$$

Therefore, for $\alpha \leqslant \psi_r$

$$\min_{\gamma,\theta}\|\mathbf{p}_{ij}(\gamma,\theta)-(\boldsymbol{\mu}_{2j}-\mathbf{t}_0)\| = \big|\|\boldsymbol{\mu}_{1i}\|-\|\boldsymbol{\mu}_{2j}-\mathbf{t}_0\|\big|. \tag{5.32}$$

When $\alpha > \psi_r$ (Case 2), equation (5.30) is minimised when $\gamma = \psi_r$, giving

$$\begin{aligned}
\min_{\gamma,\theta}\|\mathbf{p}_{ij}(\gamma,\theta)-\boldsymbol{\mu}_{2j}^0\|^2 &= \|\boldsymbol{\mu}_{1i}\|^2+\|\boldsymbol{\mu}_{2j}^0\|^2-2\cos(\alpha-\psi_r)\|\boldsymbol{\mu}_{1i}\|\|\boldsymbol{\mu}_{2j}^0\| && (5.33)\\
&= \|\mathbf{p}_{ij}(\psi_r,0)-\boldsymbol{\mu}_{2j}^0\|^2 && (5.34)
\end{aligned}$$

Therefore, for $\alpha > \psi_r$

$$\min_{\gamma,\theta}\|\mathbf{p}_{ij}(\gamma,\theta)-(\boldsymbol{\mu}_{2j}-\mathbf{t}_0)\| = \|\mathbf{p}_{ij}(\psi_r,0)-(\boldsymbol{\mu}_{2j}-\mathbf{t}_0)\|, \tag{5.35}$$

thus proving the lemma for both cases. □

Using this lemma, a lower bound on the minimum pairwise residual error within a transformation domain $\mathcal{C}_r \times \mathcal{C}_t$ can be found.

Theorem 5.2. *(Lower bound of the minimum pairwise residual error) For the transformation domain $\mathcal{C}_r \times \mathcal{C}_t$ centred at $(\mathbf{r}_0, \mathbf{t}_0)$, the lower bound of the minimum pairwise residual error can be chosen as*

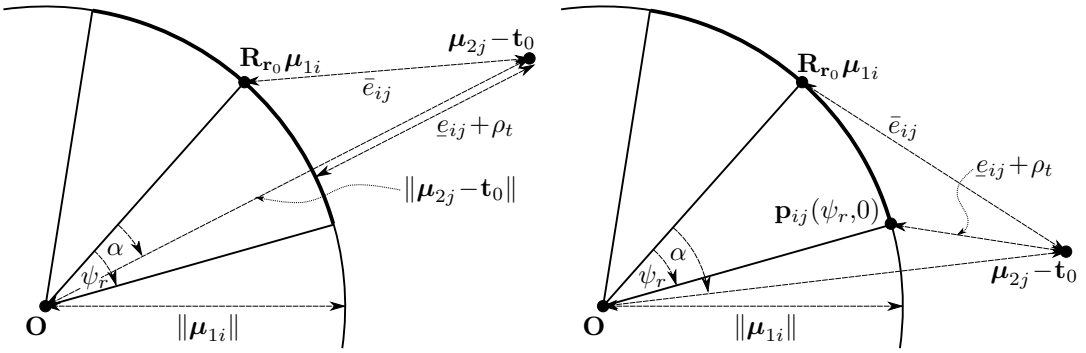
$$\underline{e}_{ij} \triangleq \begin{cases} \max\Big\{\big|\|\boldsymbol{\mu}_{1i}\| - \|\boldsymbol{\mu}_{2j} - \mathbf{t}_0\|\big| - \rho_t(\mathcal{C}_t), 0\Big\} & \text{for } \alpha \leqslant \psi_r(\mathcal{C}_r) \\ \max\Big\{\|\mathbf{p}_{ij}(\psi_r(\mathcal{C}_r), 0) - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\| - \rho_t(\mathcal{C}_t), 0\Big\} & \text{for } \alpha > \psi_r(\mathcal{C}_r) \end{cases} \tag{5.36}$$

Proof. Observe that $\forall(\mathbf{r}, \mathbf{t}) \in (\mathcal{C}_r \times \mathcal{C}_t)$,

$$\begin{aligned} e_{ij}(\mathbf{R}_\mathbf{r}, \mathbf{t}) &= \|\mathbf{R}_\mathbf{r}\boldsymbol{\mu}_{1i} - (\boldsymbol{\mu}_{2j} - \mathbf{t})\| && (5.37)\\ &= \|\mathbf{R}_\mathbf{r}\boldsymbol{\mu}_{1i} - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0) - (\mathbf{t}_0 - \mathbf{t})\| && (5.38)\\ &\geqslant \Big|\|\mathbf{R}_\mathbf{r}\boldsymbol{\mu}_{1i} - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\| - \|\mathbf{t}_0 - \mathbf{t}\|\Big| && (5.39)\\ &\geqslant \max\Big\{\|\mathbf{R}_\mathbf{r}\boldsymbol{\mu}_{1i} - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\| - \|\mathbf{t}_0 - \mathbf{t}\|, 0\Big\} && (5.40)\\ &\geqslant \max\Big\{\|\mathbf{R}_\mathbf{r}\boldsymbol{\mu}_{1i} - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\| - \rho_t(\mathcal{C}_t), 0\Big\} && (5.41)\\ &\geqslant \max\Big\{\min_{\mathbf{r}\in\mathcal{C}_r}\|\mathbf{R}_\mathbf{r}\boldsymbol{\mu}_{1i} - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\| - \rho_t(\mathcal{C}_t), 0\Big\} && (5.42)\\ &\geqslant \max\Big\{\min_{\gamma,\theta}\|\mathbf{p}_{ij}(\gamma, \theta) - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\| - \rho_t(\mathcal{C}_t), 0\Big\} && (5.43)\\ &= \begin{cases} \max\Big\{\big|\|\boldsymbol{\mu}_{1i}\| - \|\boldsymbol{\mu}_{2j} - \mathbf{t}_0\|\big| - \rho_t(\mathcal{C}_t), 0\Big\} & \text{for } \alpha \leqslant \psi_r(\mathcal{C}_r) \\ \max\Big\{\|\mathbf{p}_{ij}(\psi_r(\mathcal{C}_r), 0) - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\| - \rho_t(\mathcal{C}_t), 0\Big\} & \text{for } \alpha > \psi_r(\mathcal{C}_r) \end{cases} && (5.44) \end{aligned}$$

where (5.39) follows from the reverse triangle inequality $\|\mathbf{x}-\mathbf{y}\| \geqslant |\|\mathbf{x}\| - \|\mathbf{y}\||$, (5.40) states that the absolute value of a quantity is positive, (5.41) follows from Lemma 5.3, (5.42) follows from minimising the norm over the rotation domain, (5.43) states that the minimum distance to a constrained point on the spherical cap (that is, a point in the umbrella-shaped region) is greater than or equal to the minimum distance to an unconstrained point on the cap, and (5.44) follows from Lemma 5.4. Finally, since the inequality is true for all $(\mathbf{r}, \mathbf{t}) \in (\mathcal{C}_r \times \mathcal{C}_t)$, it is also true for $(\mathbf{r}^*, \mathbf{t}^*)$ that minimise $e_{ij}(\mathbf{R}_\mathbf{r}, \mathbf{t})$ over the transformation domain, that is $e^*_{ij} \geqslant \underline{e}_{ij}$. □

The geometric intuition for the lower bound of the minimum pairwise residual error (Theorem 5.2) is shown in Figure 5.7. The minimum distance to the spherical cap is equal to (i) the radial distance to the sphere if the point $(\boldsymbol{\mu}_{2j} - \mathbf{t}_0)$ lies within the rotation cone such that $\angle(\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i}, \boldsymbol{\mu}_{2j} - \mathbf{t}_0) \leqslant \psi_r$; or (ii) the distance to the edge of the cap if the point lies outside the rotation cone.



(a) Case 1: $(\boldsymbol{\mu}_{2j} - \mathbf{t}_0)$ is within the rotation cone $(\alpha \leqslant \psi_r)$

(b) Case 2: $(\boldsymbol{\mu}_{2j} - \mathbf{t}_0)$ is outside the rotation cone $(\alpha > \psi_r)$

Figure 5.7: Upper and lower bounds of the minimum pairwise residual error. A 2D cross-section in the plane defined by points $\{\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i}, \boldsymbol{\mu}_{2j} - \mathbf{t}_0, \mathbf{O}\}$ is shown. The spherical cap cross-section is depicted as a bold curve. Observe that the minimum distance to the spherical cap is equal to (i) the radial distance to the sphere or (ii) the distance to the edge of the cap, depending on the relative position of the point $(\boldsymbol{\mu}_{2j} - \mathbf{t}_0)$.

Bounds on the minimum of the GMA objective function (5.1) can be found by summing the kernelised upper and lower bounds of the pairwise residual errors in (5.13) and (5.36) for all $n_1 \times n_2$ Gaussian pairs.

Corollary 5.1. *(Bounds of the GMA objective function) For the 3D transformation domain $\mathcal{C}_r \times \mathcal{C}_t$ centred at $(\mathbf{r}_0, \mathbf{t}_0)$, the upper bound $\bar{f}$ and the lower bound $\underline{f}$ of the minimum objective function value $f(\mathbf{R}_\mathbf{r}, \mathbf{t})$ can be chosen as*

$$\bar{f} = -\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \frac{\phi_{1i}\phi_{2j}}{\left(\sigma_{1i}^2 + \sigma_{2j}^2\right)^{\frac{3}{2}}} \exp\left(-\frac{\bar{e}_{ij}^2}{2\left(\sigma_{1i}^2 + \sigma_{2j}^2\right)}\right) \tag{5.45}$$

$$\underline{f} = -\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \frac{\phi_{1i}\phi_{2j}}{\left(\sigma_{1i}^2 + \sigma_{2j}^2\right)^{\frac{3}{2}}} \exp\left(-\frac{\underline{e}_{ij}^2}{2\left(\sigma_{1i}^2 + \sigma_{2j}^2\right)}\right). \tag{5.46}$$

Comparison of Pairwise Lower Bounds

In Yang et al. [2016], a rotation uncertainty distance was derived that provided an upper bound on the maximum distance between the points $\mathbf{R}_\mathbf{r}\boldsymbol{\mu}_{1i}$ and $\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i}$ for $\mathbf{r} \in \mathcal{C}_r$. Using the notation of this chapter, their rotation uncertainty distance was given by

$$\rho_r(\boldsymbol{\mu}_{1i}, \mathcal{C}_r) \triangleq 2\sin\Big(\min\Big\{\sqrt{3}\delta_r/2, \pi/2\Big\}\Big)\|\boldsymbol{\mu}_{1i}\| \geqslant \max_{\mathbf{r}\in\mathcal{C}_r} \|\mathbf{R}_\mathbf{r}\boldsymbol{\mu}_{1i} - \mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i}\|. \tag{5.47}$$

From this, a weaker lower bound on the minimum pairwise residual error was given by

$$\underline{e}_{ij}^{w} \triangleq \max\Big\{\|\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i} - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\| - \rho_r(\boldsymbol{\mu}_{1i}, \mathcal{C}_r) - \rho_t(\mathcal{C}_t), 0\Big\}. \tag{5.48}$$

Neglecting translation, this represents the distance from $(\boldsymbol{\mu}_{2j} - \mathbf{t}_0)$ to a sphere a radius ρ_r enclosing the spherical cap.

The lower bound of the pairwise residual error presented in this chapter $\underline{e}_{ij}$ is greater than the weaker lower bound $\underline{e}_{ij}^{w}$ from Yang et al. [2016], leading to tighter bounds on the objective function. This improves the efficiency of the branch-and-bound algorithm, allowing sub-optimal branches to be pruned earlier. The inequality $\underline{e}_{ij} \geqslant \underline{e}_{ij}^{w}$ is proved in Lemma 5.5 and shown in Figure 5.8.

Lemma 5.5. *(Pairwise residual error inequality) For the transformation domain $\mathcal{C}_r \times \mathcal{C}_t$ centred at $(\mathbf{r}_0, \mathbf{t}_0)$,*

$$\underline{e}_{ij} \geqslant \underline{e}_{ij}^{w}. \tag{5.49}$$

Proof. To prove inequality (5.49) is to prove that

$$\min_{\gamma,\theta} \|\mathbf{p}_{ij}(\gamma,\theta) - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\| \geqslant \|\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i} - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\| - \rho_r(\boldsymbol{\mu}_{1i}, \mathcal{C}_r) \tag{5.50}$$

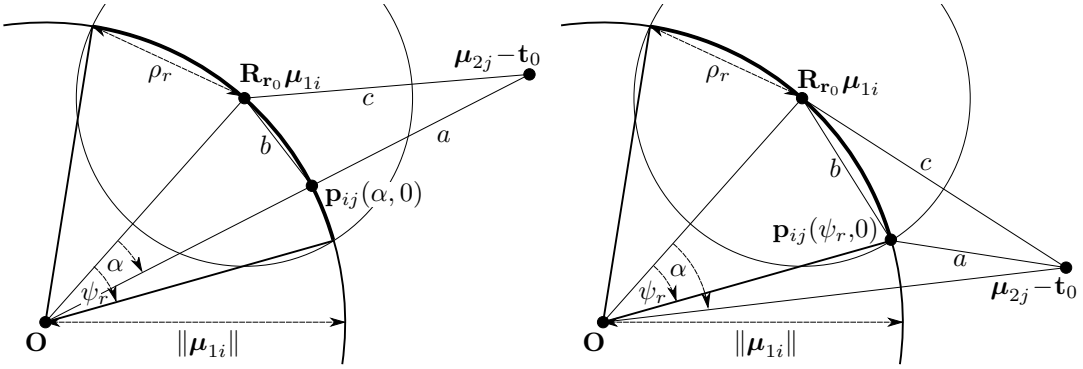
since the translation terms in (5.36) and (5.48) cancel out. For case 1 ($\alpha \leqslant \psi_r$),

$$\begin{aligned}
\min_{\gamma,\theta} \|\mathbf{p}_{ij}(\gamma,\theta) - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\| &= \|\mathbf{p}_{ij}(\alpha, 0) - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\| && (5.51)\\
&= \|(\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i} - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)) - (\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i} - \mathbf{p}_{ij}(\alpha, 0))\| && (5.52)\\
&\geqslant \|\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i} - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\| - \|\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i} - \mathbf{p}_{ij}(\alpha, 0)\| && (5.53)\\
&\geqslant \|\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i} - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\| - \rho_r(\boldsymbol{\mu}_{1i}, \mathcal{C}_r) && (5.54)
\end{aligned}$$

where (5.53) follows from the reverse triangle inequality and (5.54) follows from (5.47). Similarly for case 2 ($\alpha > \psi_r$),

$$\begin{aligned}
\min_{\gamma,\theta} \|\mathbf{p}_{ij}(\gamma,\theta) - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\| &= \|\mathbf{p}_{ij}(\psi_r, 0) - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\| && (5.55)\\
&= \|(\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i} - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)) - (\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i} - \mathbf{p}_{ij}(\psi_r, 0))\| && (5.56)\\
&\geqslant \|\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i} - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\| - \|\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i} - \mathbf{p}_{ij}(\psi_r, 0)\| && (5.57)\\
&= \|\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i} - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\| - \rho_r(\boldsymbol{\mu}_{1i}, \mathcal{C}_r). && (5.58)
\end{aligned}$$

See Figure 5.8 for diagrams of the relevant triangles. □



(a) Case 1: $(\boldsymbol{\mu}_{2j} - \mathbf{t}_0)$ is within the rotation cone ($\alpha \leqslant \psi_r$)

(b) Case 2: $(\boldsymbol{\mu}_{2j} - \mathbf{t}_0)$ is outside the rotation cone ($\alpha > \psi_r$)

Figure 5.8: Comparison of the pairwise lower bound. For both cases 1 and 2, the triangle inequality $a + b \geqslant c$ holds, with $a = \|\mathbf{p}_{ij}(\gamma, 0) - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\|$, $b = \|\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i} - \mathbf{p}_{ij}(\gamma, 0)\|$ and $c = \|\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i} - (\boldsymbol{\mu}_{2j} - \mathbf{t}_0)\|$ for $\gamma = \alpha$ and ψ_r respectively.

5.5 The GOGMA Algorithm

The Globally-Optimal Gaussian Mixture Alignment (GOGMA) algorithm is outlined in Algorithm 5.1. It employs branch-and-bound with depth-first search using a priority queue where the priority is inverse to the lower bound (Line 4). The algorithm terminates with ϵ-optimality, whereby the difference between the best function value so far f^* and the global lower bound $\underline{f}$ is less than ϵ (Line 5).

Algorithm 5.1 GOGMA: a branch-and-bound algorithm for globally-optimal Gaussian mixture alignment in $SE(3)$.

Input: two Gaussian mixture models with parameter sets $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_{ki}, \sigma^2_{ki}, \phi_{ki}\}_{i=1}^{n_k}$, means $\boldsymbol{\mu}_{ki}$, variances σ^2_{ki}, and mixture weights ϕ_{ki}; optimality tolerance ϵ; initial transformation domain $\Omega = \Omega_r \times \Omega_t$ centred at $(\mathbf{r}_0, \mathbf{t}_0)$

Output: ϵ-optimal value f^* and corresponding transformation $(\mathbf{r}^*, \mathbf{t}^*)$

1: Run local optimisation: $(f^*, \mathbf{r}^*, \mathbf{t}^*) \leftarrow \text{GMA}(\mathbf{r}_0, \mathbf{t}_0)$
2: Add transformation domain Ω to priority queue Q
3: **loop**
4: Remove hypercube $\mathcal{C} = \mathcal{C}_r \times \mathcal{C}_t$ with lowest lower-bound $\underline{f}$ from Q
5: **if** $f^* - \underline{f} \leqslant \epsilon$ **then** terminate
6: In parallel, evaluate $\bar{f}_i$ (5.45) and $\underline{f}_i$ (5.46) for all sub-hypercubes of $\mathcal{C}$
7: **for all** sub-hypercubes $\mathcal{C}_i$ **do**
8: **if** $\bar{f}_i < f^*$ **then** $(f^*, \mathbf{r}^*, \mathbf{t}^*) \leftarrow \text{GMA}(\mathbf{r}_{0i}, \mathbf{t}_{0i})$
9: **if** $\underline{f}_i < f^*$ **then** add $\mathcal{C}_i$ to queue: $Q \leftarrow \mathcal{C}_i$

In this implementation, the upper and lower bounds of 4096 sub-cubes are found simultaneously on the GPU (line 6). A higher branching factor can be used, although memory considerations must be taken into account to ensure that the priority queue

does not increase much faster than it can be pruned. A branching factor of 4096 performs well and does not require a high-end GPU. Other than the bound calculations, the code is executed entirely on the CPU.

Lines 1 and 8 show how local optimisation is integrated into the algorithm using Gaussian Mixture Alignment (GMA). Firstly, the best-so-far function value f^* and the associated transformation parameters are initialised using GMA (line 1). Within the main loop, GMA is run whenever the BB algorithm finds a sub-hypercube $\mathcal{C}_i$ that has an upper bound less than the best-so-far function value f^* (line 8). GMA is initialised with $(\mathbf{r}_{0i}, \mathbf{t}_{0i})$, the centre transformation of $\mathcal{C}_i$. In this way, BB and GMA collaborate, with GMA quickly converging to the closest local minimum and BB guiding the search into the convergence basins of increasingly lower local minima. Hence, BB jumps the search out of local minima and GMA accelerates convergence by refining f^*. Importantly, the faster f^* is refined, the more sub-hypercubes are discarded, since those with lower bounds higher than f^* are culled (line 9).

The algorithm is designed in such a way that early termination outputs the best-so-far transformation. Hence, if a limit is set on the runtime, a best-guess transformation can be provided for those alignment experiments that exceed the limit. While ϵ-optimality will not be guaranteed for them, in practise this is often adequate. In view of this, and to accelerate the removal of redundant sub-hypercubes, line 8 may be modified such that GMA is run for every sub-hypercube of the first subdivision and f^* is updated with the best function value of that batch. This is denoted as batch-initialised GOGMA.

In the following sections, convergence results and a time complexity analysis of the algorithm are provided.

5.5.1 Convergence of the Upper and Lower Bounds

A requirement of branch-and-bound is that the upper and lower bounds converge as the size of the branch tends to zero. The convergence of the bounds can be proved as follows. As the branch size tends to zero, the rotation uncertainty angle $\psi_r(\mathcal{C}_r)$ also tends to zero since it is linearly dependent on the half side-length δ_r of the rotation sub-cube $\mathcal{C}_r$. Furthermore, as $\psi_r(\mathcal{C}_r)$ tends to zero, $\mathbf{p}_{ij}(\gamma, \theta)$ (5.24) tends to $\mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i}$ for $\gamma \in [0, \psi_r \to 0]$ and $\theta \in [0, 2\pi)$. Similarly, the translation uncertainty distance $\rho_t(\mathcal{C}_t)$ tends to zero as the branch size tends to zero since it is linearly dependent on the half side-length δ_t of the translation sub-cube $\mathcal{C}_t$. From $\mathbf{p}_{ij}(\gamma, \theta) \to \mathbf{R}_{\mathbf{r}_0}\boldsymbol{\mu}_{1i}$ as $\delta_r \to 0$ and $\rho_t(\mathcal{C}_t) \to 0$ as $\delta_t \to 0$, it is clear that $\underline{e}_{ij}$ (5.36) tends to $\bar{e}_{ij}$ (5.13) as the branch size tends to zero. Hence, the lower bound of the objective function (5.46) converges to the upper bound (5.45) as the branch size decreases.

5.5.2 Time Complexity

For a given rotation tolerance η_r and translation tolerance η_t, it possible to derive a bound on the worst-case search tree depth and thereby obtain the time complexity of the algorithm. In terms of the size of the input, the GOGMA algorithm is $\mathcal{O}(n_1 n_2)$, where n_k is the number of Gaussian components in each mixture model. However, the notation conceals a very large constant. Including the constant factors that can be selected by the user yields $\mathcal{O}(\max\{\eta_r^{-6}, \delta_{t_0}^6 \eta_t^{-6}\} n_1 n_2)$ for the time complexity, where δ_{t_0} is the half side-length of the initial translation cubes, that is, one-quarter the half side-length τ of the translation domain.

Calculating the upper and lower bounds involves a summation over the components of both mixture models, therefore the complexity is $\mathcal{O}(n_1 n_2)$. However, it is as of yet unclear how the number of iterations (explored sub-hypercubes) depends on the inputs. The central finding is that branch-and-bound is exponential in the worst-case tree search depth D, but D is logarithmic in η_r^{-1} and η_t^{-1}. Therefore the complexity of BB is polynomial in η_r^{-1} and η_t^{-1}, the rotation and translation tolerances.

Theorem 5.3. *(Search Depth and Time Complexity) Let $\delta_{r_0} = \pi/4$ be the half side-length of the initial rotation sub-cubes $\mathcal{C}_{r_0}$ and $\delta_{t_0} = \tau/4$ be the half side-length of the initial translation sub-cubes $\mathcal{C}_{t_0}$. Then*

$$D = \max\left\{ \left\lceil \frac{1}{2} \log_2 \frac{\sqrt{3}\delta_{r_0}}{\eta_r} \right\rceil, \left\lceil \frac{1}{2} \log_2 \frac{\sqrt{3}\delta_{t_0}}{\eta_t} \right\rceil, 0 \right\} \tag{5.59}$$

is an upper bound on the worst-case search tree depth for a rotation tolerance η_r and translation tolerance η_t, and $\mathcal{O}(\max\{\eta_r^{-6}, \delta_{t_0}^6 \eta_t^{-6}\} n_1 n_2)$ is the time complexity of the GOGMA algorithm.

Proof. To achieve a rotation tolerance of at least η_r, then $\psi_r(\mathcal{C}_r) \leqslant \eta_r$. From (5.6),

$$\psi_r(\mathcal{C}_r) = \min\{\sqrt{3}\delta_r, \pi\} \leqslant \sqrt{3}\delta_r. \tag{5.60}$$

At the search tree depth D_r, the half side-length is given by

$$\delta_{r_{D_r}} = \frac{1}{4}\delta_{r_{D_r-1}} = 2^{-2D_r}\delta_{r_0}. \tag{5.61}$$

Substituting into (5.60) gives

$$\psi_r(\mathcal{C}_{r_{D_r}}) \leqslant \sqrt{3}\delta_{r_{D_r}} = 2^{-2D_r}\sqrt{3}\delta_{r_0}. \tag{5.62}$$

To find the worst-case search tree depth, the constraint $\psi_r(\mathcal{C}_r) \leqslant \eta_r$ is applied:

$$\psi_r(\mathcal{C}_{r_{D_r}}) \leqslant 2^{-2D_r}\sqrt{3}\delta_{r_0} \leqslant \eta_r. \tag{5.63}$$

Taking the logarithm of both sides yields

$$D_r \geqslant \frac{1}{2}\log_2 \frac{\sqrt{3}\delta_{r_0}}{\eta_r}. \tag{5.64}$$

Therefore, since D_r is required to be a non-negative integer,

$$D_r = \left\lceil \frac{1}{2}\log_2 \frac{\sqrt{3}\delta_{r_0}}{\eta_r} \right\rceil. \tag{5.65}$$

To achieve a translation tolerance of at least η_t, then $\rho_t(\mathcal{C}_t) \leqslant \eta_t$. From (5.9),

$$\rho_t(\mathcal{C}_t) = \sqrt{3}\delta_t. \tag{5.66}$$

At the search tree depth D_t, the half side-length is given by

$$\delta_{t_{D_t}} = \frac{1}{4}\delta_{t_{D_t-1}} = 2^{-2D_t}\delta_{t_0}. \tag{5.67}$$

Substituting into (5.66) gives

$$\rho_t(\mathcal{C}_{t_{D_t}}) \leqslant \sqrt{3}\delta_{t_{D_t}} = 2^{-2D_t}\sqrt{3}\delta_{t_0}. \tag{5.68}$$

To find the worst-case search tree depth, the constraint $\rho_t(\mathcal{C}_t) \leqslant \eta_t$ is applied:

$$\rho_t(\mathcal{C}_{t_{D_t}}) \leqslant 2^{-2D_t}\sqrt{3}\delta_{t_0} \leqslant \eta_t. \tag{5.69}$$

Taking the logarithm of both sides yields

$$D_t \geqslant \frac{1}{2}\log_2 \frac{\sqrt{3}\delta_{t_0}}{\eta_t}. \tag{5.70}$$

Therefore, since D_t is required to be a non-negative integer,

$$D_t = \left\lceil \frac{1}{2}\log_2 \frac{\sqrt{3}\delta_{t_0}}{\eta_t} \right\rceil. \tag{5.71}$$

Equation (5.59) follows from the requirement that $D = \max\{D_r, D_t\}$.

Now, the BB algorithm will have examined at most

$$N = 4096(1 + 4096 + 4096^2 + \cdots + 4096^D) = \frac{4096}{4095}\Big((2^{D+1})^{12} - 1\Big) \tag{5.72}$$

sub-hypercubes at search depth D, due to the hyperoctree structure. Finally, substituting (5.59) into (5.72) and simplifying using Bachmann–Landau notation gives

$$N = O\left(\max\left\{\left(\frac{\delta_{r_0}}{\eta_r}\right)^6, \left(\frac{\delta_{t_0}}{\eta_t}\right)^6\right\}\right) = O\Big(\max\Big\{\eta_r^{-6}, \delta_{r_0}^6\eta_t^{-6}\Big\}\Big). \tag{5.73}$$

The δ_{r_0} term is removed because it is a constant (equal to $\pi/4$) that is not selected by the user. For each sub-hypercube, the upper and lower bounds of the objective function are calculated with a time complexity of $\mathcal{O}(n_1 n_2)$. Combining this with the number of explored sub-hypercubes gives the time complexity of the GOGMA algorithm. □

While tolerances on rotation and translation are easily implemented by only branching sub-hypercubes above a certain size, in this implementation a single value ϵ is used instead. This limits the gap between the objective function bounds when the algorithm terminates and ensures that the optimal L_2 distance between the Gaussian mixtures is within ϵ of the output L_2 distance. As such, this time complexity analysis does not apply strictly to the implementation tested in the next section.

It is also important to observe that experimental evaluation of runtime is more revealing for BB algorithms than time complexity analysis. The main reason to use BB is that it can prune large regions of the search space, reducing the size of the problem. This is not reflected in the complexity analysis.

5.6 Results

The GOGMA algorithm was evaluated with respect to the baseline local algorithms Iterative Closest Point (ICP) [Besl and McKay, 1992] and Coherent Point Drift (CPD) [Myronenko and Song, 2010], and the nearest competitor Globally-optimal ICP (Go-ICP) [Yang et al., 2016] on two large-scale field datasets. It was also evaluated on 3D data collected under controlled laboratory conditions to test its optimality and the effect of different factors on the runtime.

In order to test the algorithms across a uniformly-distributed sample of rotation space $SO(3)$, the 72 base grid rotations from Incremental Successive Orthogonal Images (ISOI) [Yershova et al., 2010] were used. Translation perturbations were not applied since the point-sets were centred and scaled to $[-1, 1]^3$ in a pre-processing step before being converted to GMMs, which removes any translation perturbations. The

transformation domain was set to be $[-\pi, \pi]^3 \times [-0.5, 0.5]^3$. This translation domain corresponds to an explored volume over 3× larger than the bounding box of the largest point-set (1.5× per dimension).

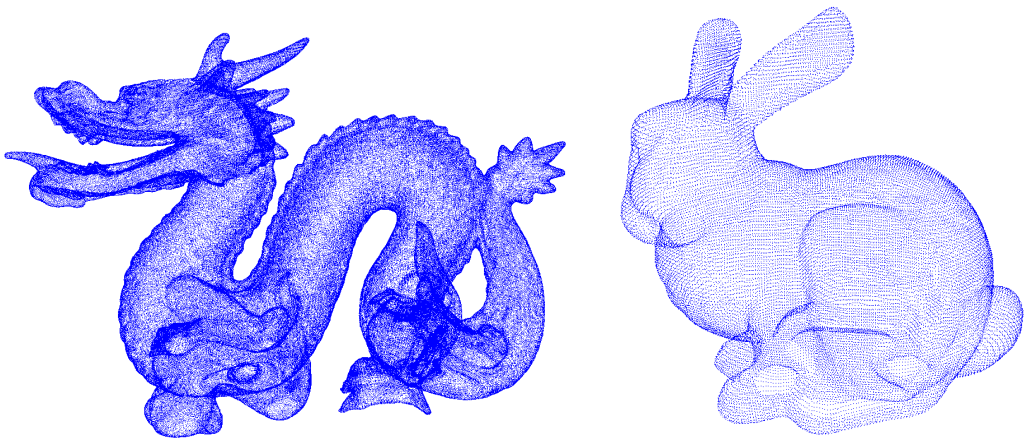
Except where otherwise specified, the convergence threshold was set to $\epsilon = 0.1$, the number of Gaussian components was set to $n_1, n_2 \approx 50$, batch initialisation was used and the GMMs were Support Vector–parametrised Gaussian Mixtures (SVGMs), whereby an SVM and a mapping are used to efficiently construct an adaptive GMM from point-set data [Campbell and Petersson, 2015]. SVGMs allow the user to specify the approximate number of components and set equal variances σ^2 automatically, based on the desired number of components.

Although GOGMA is a general-purpose Gaussian mixture alignment algorithm, the runtime results include the time required for GMM construction in order to facilitate comparison with other point-set registration algorithms. All experiments were run on a PC with a 3.7GHz Quad Core CPU with 32GB of RAM and a Nvidia GeForce GTX 980 GPU. The GOGMA code is written in unoptimised C++ and uses the VXL numerics library [VXL, 2014] for local GMA optimisation.

5.6.1 Fully-Overlapping Registration Experiments

To demonstrate optimality of the algorithm with respect to the objective function, fully-overlapping point-sets were used. That is, each point-set pair was sampled from the same surface in its entirety, with no structured outliers from partial-overlap or occlusion. For these experiments, the reconstructed DRAGON-RECON [Curless and Levoy, 2014] and BUNNY-RECON [Turk and Levoy, 2014] point-sets from the Stanford Computer Graphics Laboratory, shown in Figure 5.9, were aligned with transformed copies of themselves, using the 72 ISOI rotations. Identical point-sets were required in order to obtain the ground-truth optimal objective function values, because the global optimum does not necessarily coincide with the ground-truth transformation for partially-overlapping point-sets. The global optimum was found for all 144 registration experiments, with mean separations from the optimal value being 9×10^{-8} and 3×10^{-7} and mean runtimes being 17s and 14s, for DRAGON and BUNNY respectively. With batch initialisation, the mean separations were 8×10^{-8} and 7×10^{-8} and the mean runtimes were 33s and 29s, for DRAGON and BUNNY respectively.

The evolution of the global upper and lower bounds is shown in Figure 5.10. It can be seen that BB and GMA collaborate to reduce the upper bound: BB guides the search into the convergence basins of increasingly lower local minima and GMA refines the bound by jumping to the nearest local minimum. Discontinuities in the lower bound occur when an entire sub-hypercube level has been explored. With batch initialisation,



(a) DRAGON-RECON (b) BUNNY-RECON

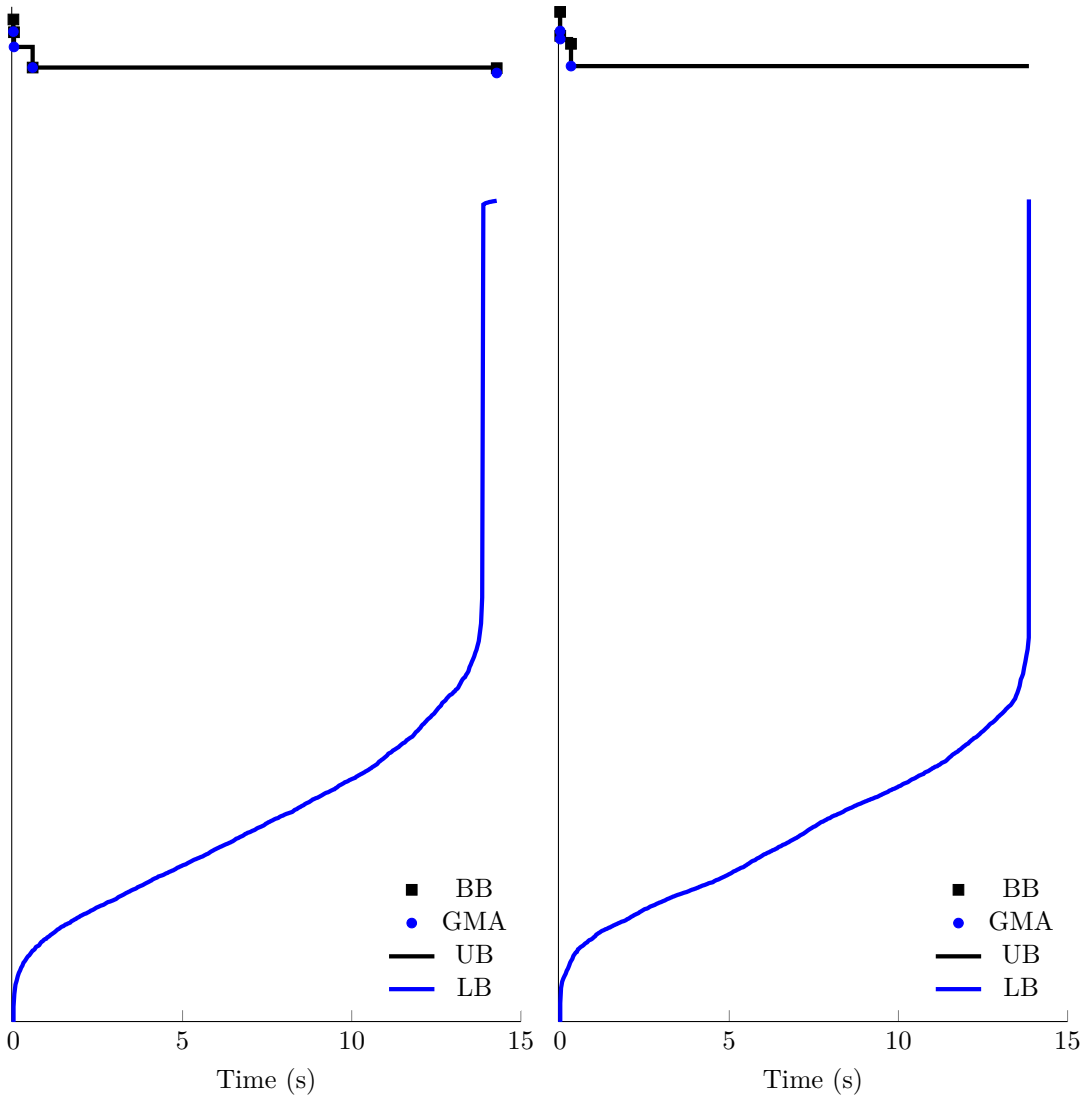
Figure 5.9: The DRAGON-RECON and BUNNY-RECON reconstructed models from the Stanford Computer Graphics Laboratory.

the global minimum is generally captured at the start of the algorithm. The remaining time is spent increasing the lower bound until ϵ-optimality can be guaranteed. While batch initialisation can increase the runtime for less challenging datasets or larger values of ϵ, it typically reduces runtime and is the preferred setting.

5.6.2 Partial-to-Full Registration Experiments

In this section, the more challenging problem of partial-to-full registration is addressed. That is, one of the point-sets was sampled from a subset of the surfaces that the other point-set was sampled from, resulting in structured outliers. In these experiments, the performance of the GOGMA algorithm was evaluated by aligning single-view partial scans with a full 3D model, a common registration task. The point-sets were drawn from the Stanford Computer Graphics Laboratory's DRAGON dataset [Curless and Levoy, 2014] and consist of one reconstructed model (DRAGON-RECON) and 15 partial scans (DRAGON-STAND). The 72 base ISOI rotations were used as the initial transformations for the partial scans. For the standard parameter settings, GOGMA found the correct alignment for all 1080 experiments, with all translation errors less than 0.01m and all rotation errors less than 3°. Quantitative results are given in the SVM column of Table 5.1.

To investigate the effect of other GMM types on the accuracy and runtime of the algorithm, the experiment was repeated with GMMs generated by fixed-bandwidth Kernel Density Estimation (KDE) [Jian and Vemuri, 2011] and Expectation Maximisation (EM) [Dempster et al., 1977]. The number of components was fixed ($n_1, n_2 = 50$), but the variances and mixture weights were set by the algorithms. For KDE, the variance



(a) DRAGON-RECON (b) BUNNY-RECON

Figure 5.10: Evolution of the upper and lower bounds for the reconstructed DRAGON-RECON and BUNNY-RECON models. The normalised objective function value is plotted against time. Updates to the best-so-far L_2 distance from BB are shown as black squares and updates from GMA are shown as blue dots.

was found by parameter search and the point-sets were randomly downsampled to n_1 points, the GMM mean vectors. The pose error for the optimally aligned KDE-GMMs was very high, since KDE was unable to represent the underlying surfaces sufficiently well with n_1 components. KDE-GMMs are typically highly over-parametrised, however the $\mathcal{O}(n_1 n_2)$ time complexity of GOGMA imposes tractability limits on the number of components used. The performance of the EM-GMMs show that they are a suitable input to GOGMA in terms of alignment accuracy, however the EM implementation

Table 5.1: Effect of GMM type on the accuracy and runtime of the GOGMA algorithm. The 15 single-view partial scans from DRAGON-STAND, perturbed by the 72 ISOI rotations, were aligned with the reconstructed model DRAGON-RECON using GMMs generated from a Support Vector Machine (SVM), fixed-bandwidth Kernel Density Estimation (KDE) and Expectation Maximisation (EM). The mean/max translation error, rotation error and runtime are reported.

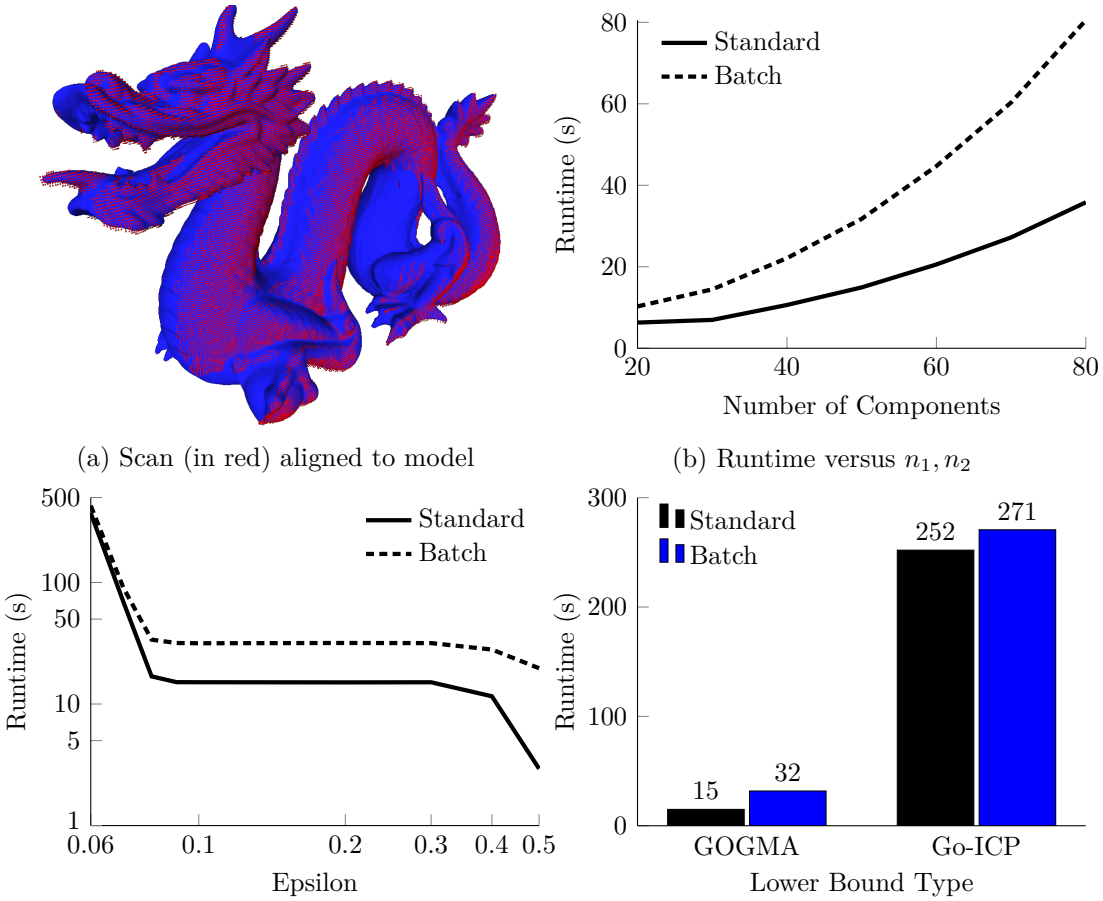
GMM Type	SVM	KDE	EM
Translation (m)	**0.004/0.008**	0.14/0.21	0.02/0.18
Rotation (°)	**1.5/2.7**	116/167	7.2/80
Runtime (s)	34/50	**15/15**	4960/4965

[Figueiredo and Jain, 2002] imposed significant runtime overheads, taking 4 663s to process the model (containing 437 645 points) and 256s on average to process each scan (containing 31 280 points on average), making it impractical unless more efficiently implemented. Considering both speed and accuracy, SVGMs are recommended.

To investigate the effect of other factors on the runtime, one was varied while the others were kept at the default settings: $n_1, n_2 \approx 50$, $\epsilon = 0.1$ and the GOGMA lower bound. The 72 ISOI rotations were applied to scan 0 from the DRAGON-STAND set and the mean runtimes were reported for standard and batch initialisations. The scan, aligned by GOGMA, is shown in Figure 5.11(a) in red. The results for differing numbers of Gaussian components n_1, n_2 are shown in Figure 5.11(b). The quadratic shape reflects the $\mathcal{O}(n_1 n_2)$ time complexity of the algorithm. The results for differing values of the convergence threshold ϵ are shown in Figure 5.11(c). For values of ϵ close to zero, the runtime increases steeply, while larger values allow the algorithm to terminate quicker, albeit with a looser optimality guarantee. The setting $\epsilon = 0.1$ is a suitable default value, having a 100% success rate for all experiments. For cases where many local minima have near-optimal alignments, such as for near-symmetries in the data, ϵ can be reduced. The runtime is also affected by the quality of the lower bound, as shown in Figure 5.11(d). The GOGMA lower bound, which uses the spherical cap distance (5.36), is tighter and more efficient than the Go-ICP lower bound [Yang et al., 2013b], which uses the distance to an uncertainty sphere containing the cap (5.48).

5.6.3 Partially-Overlapping Registration Experiments

In this section, the most challenging problem of partially-overlapping registration is addressed, where both point-sets are sampled from non-identical subsets of the underlying model or scene, resulting in many structured outliers in both point-sets. For these experiments, the performance of GOGMA was evaluated on two large-scale field datasets [Pomerleau et al., 2012], characterised in Table 5.2. STAIRS is a structured indoor/outdoor dataset with large and rapid variations in scanned volumes. WOOD-SUMMER is an unstructured outdoor dataset with dynamic objects. The symmetric in-



(a) Scan (in red) aligned to model

(b) Runtime versus n_1, n_2

(c) Runtime versus ϵ

(d) Runtime versus lower bound

Figure 5.11: Mean runtime of GOGMA on the DRAGON dataset with respect to different factors, for the alignment of DRAGON-RECON with point-set 0 of DRAGON-STAND, transformed by 72 uniformly distributed rotations. Note the logarithmic scale in (c).

lier fraction ω_Δ was used to calculate the overlap: the fraction of points from the joint set $I_{\mathcal{P}_1,\mathcal{P}_2}(\Delta) \cup I_{\mathcal{P}_2,\mathcal{P}_1}(\Delta)$ within Δ of a point from the other point-set. Here, $\Delta = 10\bar{d}$ where $\bar{d}$ is the mean closest point distance. The equations for the non-symmetric inlier set $I_{\mathcal{P}_1,\mathcal{P}_2}(\Delta)$ and the symmetric inlier fraction ω_Δ are

$$I_{\mathcal{P}_1,\mathcal{P}_2}(\Delta) = \{\mathbf{p}_1 \in \mathcal{P}_1 \,|\, \exists \mathbf{p}_2 \in \mathcal{P}_2 : \|\mathbf{p}_1 - \mathbf{p}_2\| \leqslant \Delta\} \tag{5.74}$$

$$\omega_\Delta = \frac{|I_{\mathcal{P}_1,\mathcal{P}_2}(\Delta)| + |I_{\mathcal{P}_2,\mathcal{P}_1}(\Delta)|}{|\mathcal{P}_1| + |\mathcal{P}_2|}. \tag{5.75}$$

For these experiments, sequential point-sets were aligned using GOGMA, Go-ICP [Yang et al., 2013b], ICP [Besl and McKay, 1992] and CPD [Myronenko and Song, 2010] with the 72 ISOI rotations as initial transformations. GOGMA with refinement was also tested, a variant that applies local GMA refinement with $n_1, n_2 \approx 1000$ compo-

Table 5.2: Characteristics of the large-scale field datasets from Pomerleau et al. [2012].

Dataset	STAIRS	WOOD-SUMMER
Number of point-sets	31	37
Mean number of points	191 000	182 000
Bounding box size	$21\times111\times27$m	$30\times53\times20$m
Mean overlap percentage	76%	77%
Number of alignments	31×72	37×72

Table 5.3: Alignment results for the STAIRS dataset. The mean translation error (in metres), rotation error (in degrees), and runtime (in seconds), and the coarse (C), medium (M) and fine (F) registration success rates (defined in the text) are reported. GOGMA with refinement is denoted by GOGMA_R and Go-ICP with $\epsilon=10^{-4}$ and $\epsilon=5\times10^{-5}$ by Go-ICP_a and Go-ICP_b.

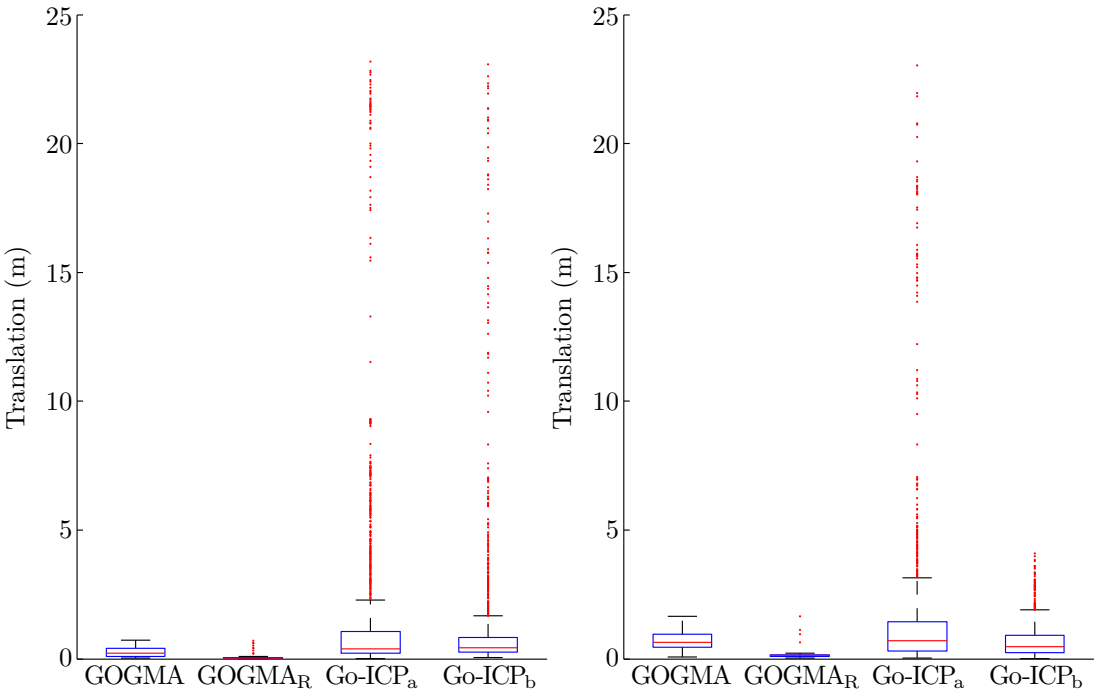
Method	GOGMA	GOGMA_R	Go-ICP_a	Go-ICP_b	ICP	CPD
Translation Error	0.26	**0.04**	1.63	1.17	4.67	5.24
Rotation Error	1.25	**0.32**	30.9	19.4	107	88.8
Success Rate (C)	**100**	**100**	71.8	80.9	15.5	38.8
Success Rate (M)	**100**	**100**	48.5	51.9	13.4	28.6
Success Rate (F)	80.0	**99.7**	19.6	21.2	6.5	7.1
Runtime	49.6	71.2	31.6	103	**0.38**	4.2

Table 5.4: Alignment results for the WOOD-SUMMER dataset. The mean translation error (in metres), rotation error (in degrees), and runtime (in seconds), and the coarse (C), medium (M) and fine (F) registration success rates (defined in the text) are reported.

Method	GOGMA	GOGMA_R	Go-ICP_a	Go-ICP_b	ICP	CPD
Translation Error	0.72	**0.13**	1.33	0.69	7.37	8.13
Rotation Error	3.09	**0.68**	9.66	5.19	109	90.7
Success Rate (C)	**100**	**100**	78.2	84.1	11.3	39.5
Success Rate (M)	75.0	**99.9**	36.6	64.5	10.8	19.3
Success Rate (F)	16.7	**99.9**	13.2	27.5	5.4	0.8
Runtime	29.5	49.6	26.2	77.7	**0.44**	4.2

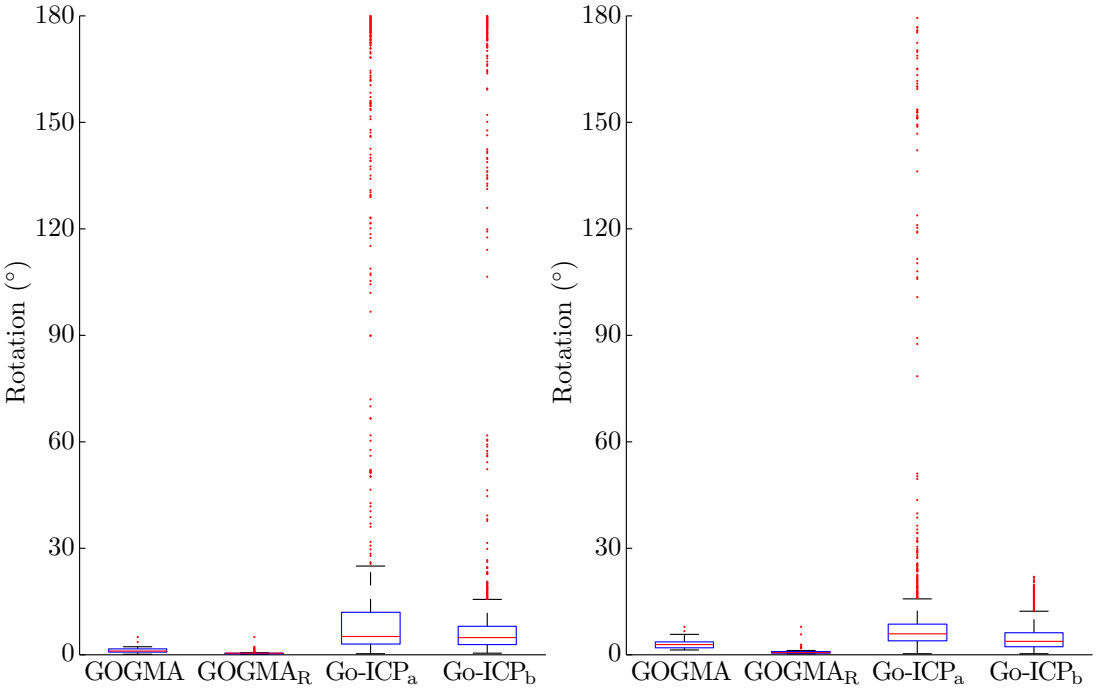
nents initialised with the output transformation of the standard GOGMA algorithm. Quantitative results are given in Tables 5.3 and 5.4 and Figure 5.12, and qualitative results in Figure 5.13. The coarse, medium and fine registration success rates are defined as the fraction of alignments with translation and rotation errors less than 2m/10°, 1m/5°, and 0.5m/2.5° respectively. The runtime values include the time to construct the GMMs (for GOGMA) and build the distance transform (for Go-ICP).

GOGMA significantly outperformed Go-ICP, ICP and CPD in these experiments, finding the correct transformation in all cases under the coarse criterion. Crucially, a subsequent refinement step (GOGMA_R) was able to find the correct transformation in virtually all cases under the fine criterion. This indicates that GOGMA without refinement was always able to find the correct alignment, up to the granularity of the 50 component representation. Go-ICP performed poorly with a loose convergence



(a) Translation error STAIRS dataset

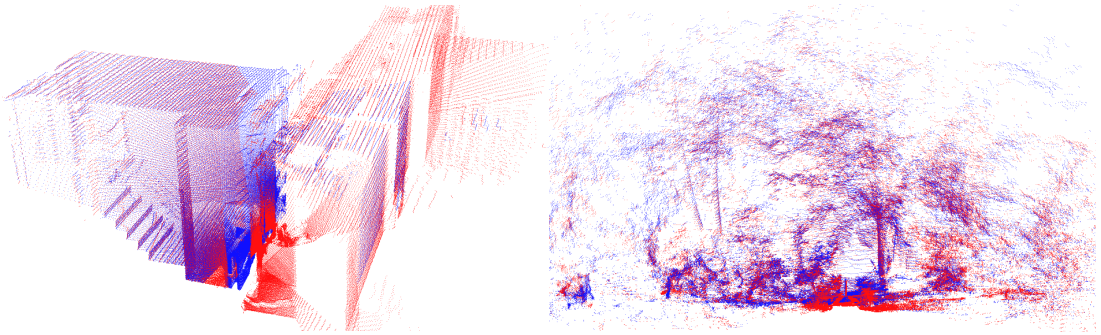
(b) Translation error WOOD-SUMMER dataset



(c) Rotation error STAIRS dataset

(d) Rotation error WOOD-SUMMER dataset

Figure 5.12: Box plots of the translation and rotation errors for the STAIRS and WOOD-SUMMER datasets. GOGMA with refinement is denoted by GOGMA$_R$ and Go-ICP with convergence threshold $\epsilon = 10^{-4}$ and $\epsilon = 5 \times 10^{-5}$ by Go-ICP$_a$ and Go-ICP$_b$. GOGMA generated few outliers, all of which were in the vicinity of the correct transformation. In contrast, Go-ICP generated many outliers, most of which were incorrect even by the coarsest success criterion.



(a) STAIRS point-sets 4 and 5 (b) WOOD-SUMMER point-sets 0 and 1

Figure 5.13: Qualitative results for two large-scale datasets. The blue scan was aligned by GOGMA from an arbitrary initial pose against the red scan, followed by GMA refinement.

threshold of $\epsilon = 10^{-4}$ and $N = 50$ points. With ϵ an order of magnitude smaller (10^{-5}), N an order of magnitude greater (500), or any trimming, the runtime became prohibitively slow. For example, the STAIRS experiments with $\epsilon = 10^{-5}$ and $N = 50$ did not terminate within 7 days, at which point it had completed 60% of the experiments with runtimes of up to 5000s per alignment. The tightest feasible ϵ ($5{\times}10^{-5}$) had a more reasonable runtime but still failed to coarsely align 19% of point-set pairs for STAIRS and 16% for WOOD-SUMMER. These failure cases were likely due to the prevalence of structured outliers in the data and are undesirable for a globally-optimal algorithm. Finally, the results show that ICP and CPD both perform poorly without a good pose prior, converging to local minima for most initialisations.

The box-plots in Figure 5.12 provide a detailed look at the results for the GOGMA and Go-ICP algorithms. GOGMA generated few alignment outliers, all of which were in the vicinity of the correct transformation, demonstrating the robustness of the approach. In contrast, Go-ICP generated many alignment outliers, most of which were incorrect even by the coarsest success criterion. These alignment outliers were likely due to the structured outliers inherent to partially-overlapping point-sets. Under the Go-ICP framework, trimming would be required to handle these outliers, however any trimming made the runtime prohibitive for these datasets.

5.6.4 Application: The Kidnapped Robot Problem

A specific application of partial-to-full registration is the kidnapped robot problem: finding the pose of a sensor within a 3D map. Also known as global localisation, this requires a 3D map of the entire environment in which the sensor could be located and does not assume a pose prior is available (hence having been 'kidnapped'). For this experiment, the APARTMENT dataset from Pomerleau et al. [2012] was used, which provides a global map of the apartment with a bounding box of $17 \times 10 \times 3$m, single-

Table 5.5: Sensor localisation results for scans of four rooms (A–D) from the APARTMENT dataset. The mean translation error (in metres), rotation error (in degrees), and runtime (in seconds), and the fine (F) registration success rate (defined in the text) are reported.

Room Scan	A	B	C	D
Translation Error	0.16	0.22	0.40	0.35
Rotation Error	0.93	0.89	1.95	2.35
Success Rate (F)	100	100	100	100
Runtime	328	383	379	409

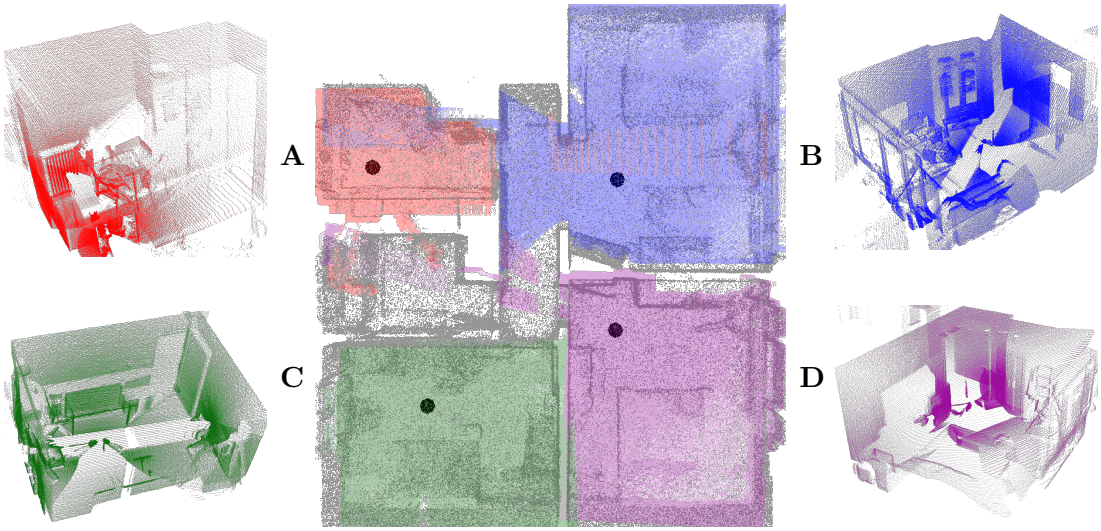


Figure 5.14: Pose estimates (black spheres) of the sensor locations for 4 room scans (red, blue, green and purple) found by aligning each scan with the entire map (grey) using GOGMA.

viewpoint scans with an average of 365 000 points and an accurate ground-truth. The dataset contains many dynamic elements, including moved boxes, chairs and people.

Four sensor positions (A–D) were selected, which correspond to four different rooms in the apartment. The scans from these positions were centred with respect to their bounding boxes, removing any translation prior, and were perturbed by the 72 ISOI rotations, removing any rotation prior, to simulate being lost or kidnapped. Finally, the GOGMA algorithm was used to localise the scans within the map. As shown in Table 5.5 and Figure 5.14, all positions were correctly localised.

5.7 Discussion

As discussed in Section 3.4, finding the global optimum of an objective function does not necessarily imply finding the ground-truth transformation. For Gaussian mixture alignment, there are two confounding factors: the quality of the representation and structured outliers induced by partial overlap and occlusion. The former refers to how

well the Gaussian mixtures describe the underlying surfaces. Increasing the number of components or using non-isotropic covariances can enable a Gaussian mixture to model increasingly complex surfaces, depending on the method used to learn the mixture. However, this is limited by the sampling rate of the point-sets. A mixture with too many components may model the sampled point-set very well, but may model the underlying surfaces less well. A smaller number of components is often more satisfactory, since it regularises the mixture, creating a smoother surface while still adapting to local surface complexity. However, if the number of components is too small, the mixture may not model the underlying surfaces well and hence the optimal alignment of this mixture with another may not occur at the true alignment of the surfaces.

Structured outliers induced by partially-overlapping point-sets and occlusion may also cause the optimal transformation to diverge from the ground-truth transformation. For Gaussian mixtures generated from partially-overlapping point-sets, there may exist an alignment that produces a smaller function value than the ground-truth alignment. However, the L_2 density distance objective function is much less susceptible to partial overlap and occlusion than other objective functions, being robust to structured outliers. This is reflected in the results presented in Section 5.6, which show that optimality with respect to the L_2 distance measure closely corresponds to optimality with respect to the true alignment. See Section 3.4.6 for more details on how the robustness of this objective function arises.

It is important to ask whether it is necessary to find the global optimum instead of a local optimum, with respect to accuracy, reliability and runtime. The results presented in Section 6.6 indicate that this is certainly the case for accuracy and especially reliability. This is unsurprising, since the objective function is highly non-convex, having a very large number of local minima. Local solvers, such as ICP and CPD, are likely to become trapped at a local minimum near the pose prior with which the algorithm was initialised. However, whether the optimality–runtime trade-off is acceptable depends on the application.

GOGMA provides an additional level of flexibility for this trade-off, since the representation resolution (the number of Gaussian components) can be reduced, thereby reducing the runtime. The experiments in Section 5.6.3 using GOGMA with refinement show how a highly accurate result can be obtained by running GOGMA with a relatively small number of components (~50), then refining the result with local optimisation with a higher number of components (~1000). While optimality cannot be guaranteed with respect to the 1000-component mixtures, it is very probable that the result is optimal. Evidence for this is shown in the box-plots of Figure 5.12, where the outlier (incorrect) alignments for 50-component GOGMA were few and always in the vicinity of the correct transformation. This means that the alignment of the low resolu-

tion mixtures corresponded closely with the alignment of the high resolution mixtures despite being orders of magnitude faster.

The GOGMA algorithm has two significant limitations. The first is the coupling of rotation and translation. In many cases it would be preferable for these transformation parameters to be decoupled. For example, when the translation domain is large, it may be desirable to branch further over the translation sub-cubes, without branching further over the rotation sub-cubes. Alternatively, the datasets may have many rotational near-symmetries, for which further subdivision of the rotation but not translation sub-cubes may be necessary. Moreover, if a specific application provides prior information about the pose, such as restricting the set of possible rotations, a decoupled approach would be better. This could be done using a nested octree structure, such as that in Yang et al. [2016], where translation search is nested inside rotation search. However, this structure is less easily parallelisable than the hyperoctree structure.

The second limitation is time complexity. The GOGMA algorithm has a time complexity of $\mathcal{O}(n_1 n_2)$ and therefore cannot handle large numbers of Gaussian components without increasing the runtime substantially. As a result, scenes cannot be modelled to a high resolution, increasing the ambiguity of the alignment problem. One solution is to introduce a data structure analogous to the distance transform that stores the set of K least-attenuated Gaussians at each point in $\mathbb{R}^3$, reducing the time complexity to $\mathcal{O}(K n_1)$. This makes use of the observation that Gaussians far from any given point have very little influence on the function value at that point, due to the rapid attenuation of Gaussians. By reducing the time complexity in this way, both Gaussian mixtures could contain more components, and one mixture could be substantially larger. This would enable the algorithm to be used for the kidnapped robot problem in large environments, where the map is many times the size of the single-view scan.

5.8 Summary

This chapter developed a theoretical framework for robust and globally-optimal 3D–3D registration by solving the Gaussian mixture alignment problem under the L_2 distance. The algorithm applied the branch-and-bound paradigm to guarantee global optimality regardless of initialisation and used local optimisation to accelerate convergence. The pivotal contribution was the derivation of the objective function bounds using the geometry of $SE(3)$. The algorithm outperformed other local and global methods on challenging field datasets, due to an objective function that is robust to structured outliers induced by partial-overlap and occlusion. The experimental evaluation provided evidence that a robust objective function and global optimality are critical for reliable 3D–3D alignment.

There are several areas that warrant further investigation with regard to the implementation of the algorithm. Firstly, runtime benefits could be realised by implementing the local optimisation on the GPU instead of the CPU. Furthermore, using a dynamic branching factor would allow more parallelism for the same memory requirements. Finally, a serial implementation could be developed to enable devices without a GPU to run the algorithm, with a time-efficient nested branch-and-bound structure. Beyond improving the implementation, there are also elements of the theory for which further work would be justified. Firstly, the rotation and translation uncertainty bounds could be tightened by inspecting the rotation and translation cubes directly, rather than using the spherical cap and circumsphere. Finally, extending the lower bound to handle full covariances with the Mahalanobis distance would enable the algorithm to be applied to more expressive Gaussian mixtures.

The following chapter will extend the investigation of robust objective functions and global optimality to the 2D–3D geometric alignment problem. There will be some elements in common with this chapter, including the branch-and-bound framework, however much of the material is characteristic to the problem. This predominantly stems from combining directional and positional data for 2D–3D alignment. In addition, the bounds on the transformation uncertainty proposed in this chapter will be extended in the next, deriving tighter and more sophisticated bounds applicable to both the 2D–3D and the 3D–3D problems.

Chapter 6

Robust and Globally-Optimal 2D–3D Alignment

The focus of this chapter is the geometric alignment of 2D directional sensor data, such as an image, with 3D positional sensor data, such as a laser scan, where the data may be corrupted by noise and random or structured outliers. This can be used to solve the problem of estimating the six degrees-of-freedom pose of a camera (or the viewpoint of a multi-camera system) from a single image relative to a precomputed 3D point-set. Perspective-n-Point (PnP) solvers are routinely used for camera pose estimation, but are contingent on the provision of a good quality set of 2D–3D correspondences. Finding cross-modality correspondences between 2D and 3D points is non-trivial, particularly when only geometric (position) information is known. Existing approaches to the 2D–3D simultaneous pose and correspondence problem use local optimisation, and are therefore unlikely to find the optimal solution without a good pose initialisation, or introduce restrictive assumptions and non-robust objective functions. Since a large proportion of outliers and many local optima are common for this problem, a useful alignment solver needs to be both robust and global. Globally-optimal approaches have the additional advantage of reliability, providing a guarantee that the solution is the global optimum.

In this chapter, a novel inlier set cardinality maximisation algorithm is proposed to jointly and robustly estimate the optimal camera pose and correspondences. The approach employs branch-and-bound to search the 6D space of camera poses, guaranteeing global optimality without requiring a pose prior. The geometry of $SE(3)$ is used to find novel upper and lower bounds for the objective function and local optimisation is integrated to accelerate convergence. Evaluation on a range of synthetic and real data empirically supports the optimality proof and shows that the method performs much more robustly than existing approaches, with runtime characteristics for the GPU implementation that are competitive with non-optimal approaches. Finally, another robust and globally-optimal approach based on minimising the L_2 distance

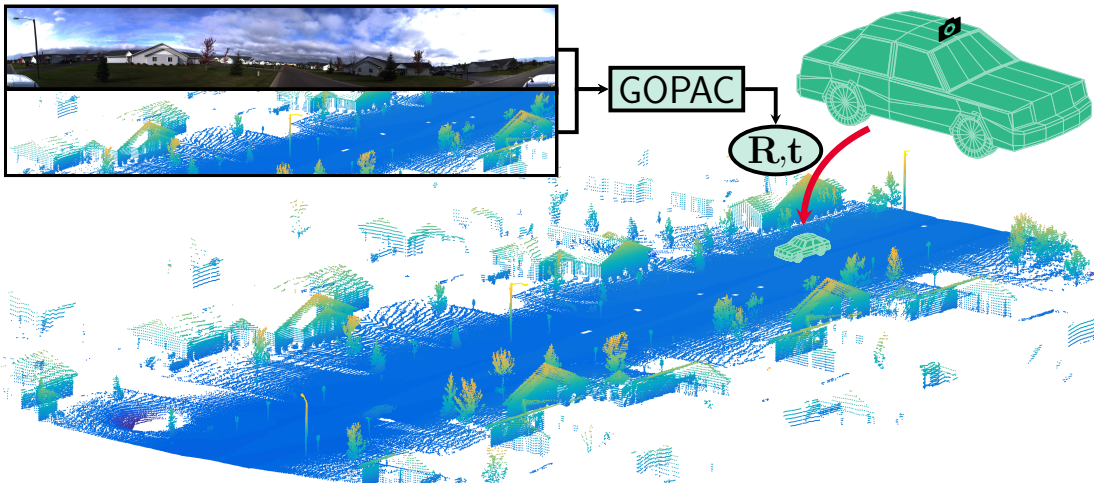


Figure 6.1: Estimating the 6-DoF absolute pose of a calibrated camera from a single image, relative to a 3D point-set, with no 2D–3D correspondences. The GOPAC algorithm solves this general case of the absolute camera pose problem with minimal assumptions about the data, simultaneously solving for the position and orientation of the camera in the world coordinate frame and the 2D–3D correspondences. To do so, a globally-optimal branch-and-bound approach is used, with tight novel bounds on the cardinality of the inlier set. In this example, the algorithm is used for the visual localisation of a car from a camera, with respect to a large-scale, unorganised 3D point-set captured by vehicle-mounted laser scanner.

between mixture models is outlined using the same framework developed for the cardinality maximisation algorithm. The outline demonstrates how the theoretical insights from the first algorithm can be transferred to develop algorithms with other objective functions and sensor data representations.

6.1 Introduction

Estimating the pose of a calibrated camera given a set of 2D points in the camera frame and a set of 3D points in the world frame, as shown in Figure 6.1, is a fundamental part of the general 2D–3D registration problem of aligning an image with a 3D scene or model. The ability to find the pose of a camera and map visual information onto a 3D model and vice versa is useful for many tasks, including camera localisation and tracking [Fischler and Bolles, 1981; Nöll et al., 2011; Kneip et al., 2015], augmented reality [Marchand et al., 2016], motion segmentation [Olson, 2001] and object recognition [Huttenlocher and Ullman, 1990; Mundy, 2006; Aubry et al., 2014].

When correspondences are known, this becomes the Perspective-n-Point (PnP) problem for which many solutions exist [Haralick et al., 1994; Lepetit et al., 2009; Kneip et al., 2011; Hesch and Roumeliotis, 2011]. However, while hypothesise-and-test

frameworks such as RANSAC [Fischler and Bolles, 1981] can mitigate the sensitivity of PnP solvers to outliers in the correspondence set, few approaches are able to handle the case where 2D–3D correspondences are not known in advance.

There are many circumstances under which correspondences may be difficult to ascertain, including the general case of aligning an image with a textureless 3D point-set or CAD model. While feature extraction techniques provide a relatively robust and reproducible way to detect interest points such as edges or corners within each modality, finding correspondences across the two modalities is much more challenging, as shown in Figure 6.2. Even when the point-set has sufficient visual information associated with it, such as colour, reflectance or SIFT features [Lowe, 2004], repetitive elements, occlusions and perspective distortion make the correspondence problem non-trivial. Moreover, appearance and thus visual features may change significantly between viewpoints, lighting conditions, weather and seasons, whereas scene geometry is often less affected. When re-localising a camera in a previously mapped environment or bootstrapping a tracking algorithm, this thesis contends that geometry is often more reliable. Therefore, there is a need for methods that solve for both pose and correspondences.

Efficient local optimisation algorithms for solving this joint problem have been proposed [David et al., 2004; Moreno-Noguer et al., 2008]. However, they require a pose prior, search only for local optima and do not provide an optimality guarantee, yielding erroneous pose estimates without a reliable means of detecting failure. Hypothesise-and-test approaches such as RANSAC [Fischler and Bolles, 1981], when applied to the correspondence-free problem [Grimson, 1990], are global methods that are not reliant on pose priors but quickly become computationally intractable as the number of points and outliers increase and do not provide an optimality guarantee. More recently, a global and ϵ-suboptimal method has been proposed [Brown et al., 2015], which uses a branch-and-bound approach to find a camera pose whose trimmed geometric error is within ϵ of the global minimum.

In this chapter, the first globally-optimal inlier set cardinality maximisation solution to the simultaneous pose and correspondence problem is proposed. Named GOPAC, the algorithm has three key features, summarised in Figure 6.3. The approach employs a branch-and-bound framework to guarantee global optimality without requiring a pose prior, ensuring that it is not susceptible to local optima. The space of rigid motions, the Special Euclidean group $SE(3)$, is parametrised in a way that facilitates branching and allows tight and novel bounds on the objective function to be derived for each branch. In addition, local optimisation methods are tightly integrated to accelerate convergence without voiding the optimality guarantee. A multi-threaded implementation on the GPU provides an additional means for greatly accelerating the algorithm.

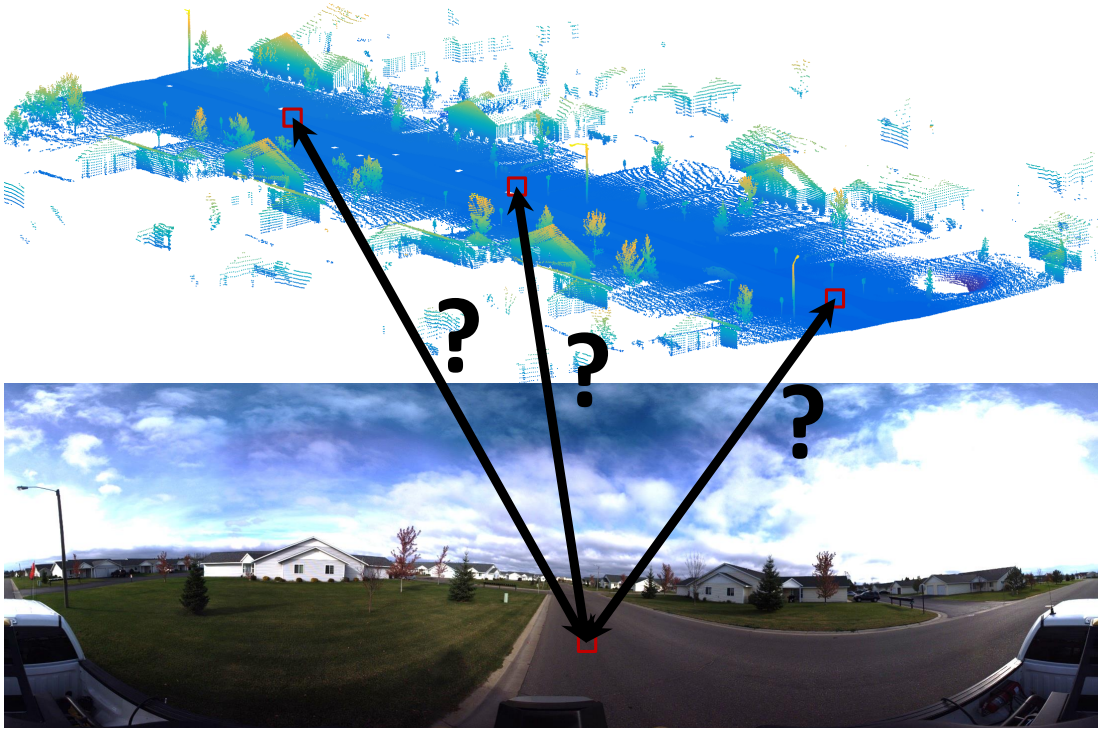


Figure 6.2: The cross-modality correspondence problem. 2D–3D correspondences are difficult to obtain, particularly across modalities. In the situation illustrated here, the point-set was captured by a laser scanner and has no visual information associated with it, making the correspondence problem challenging.

There are several advantages to using a cardinality maximisation approach. Firstly, it allows an exact optimiser to be found, unlike the ϵ-suboptimality inherent to the continuous objective function used in Brown et al. [2015]. More critically, cardinality maximisation is inherently robust to 2D and 3D outliers without smoothing the function surface and thereby moving or concealing the location of the global optimum. In contrast, other techniques to robustify geometric alignment objective functions, such as trimming or using robust loss functions, smooth and distort the surface of the original objective function. This may move the location of the global optimum and reduce its prominence with respect to other optima. In addition, trimming requires the user to specify the inlier fraction, which can rarely be known and is less intuitive to select than a geometrically meaningful inlier threshold. If the inlier fraction is over- or under-estimated, this approach may converge to the wrong pose, without a means to detect failure. Figure 3.4 demonstrates how the global optimum of a trimmed objective function, as used by Brown et al. [2015] and Yang et al. [2016] for registration problems, may not occur at the true pose, a problem that is exacerbated when the inlier fraction is guessed incorrectly. A final advantage of cardinality maximisation is

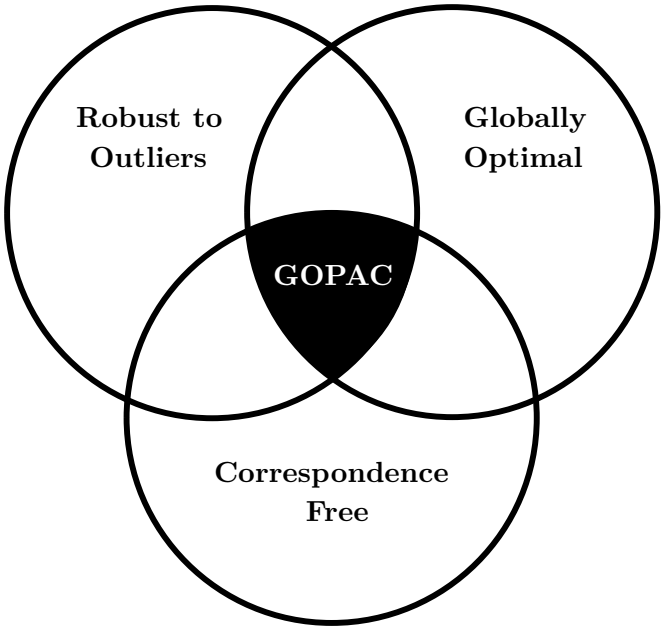


Figure 6.3: Key features of the GOPAC algorithm, a globally-optimal algorithm for camera pose and correspondence recovery. GOPAC lies in the intersection region of three desirable features for 2D–3D registration algorithms. It is robust to outliers since it maximises the number of inliers directly, unlike Brown et al. [2015]. It uses branch-and-bound with tight novel bounds to find the global maximum of the objective function, and therefore does not need a pose prior, unlike SoftPOSIT [David et al., 2004] and BlindPnP [Moreno-Noguer et al., 2008]. Finally, it jointly solves for pose and correspondences using only geometry and therefore does not need any correct correspondences to be provided, unlike PnP methods including robust P3P–RANSAC.

that it operates directly on discrete sensor data representations, 2D and 3D points, without making assumptions about the underlying structure of the data. As a result, it can be applied in situations where the structure is not obvious from the data, such as for sparse point-sets. For example, it could be used to align an image and a point-set captured simultaneously of a flock of birds or a swarm of drones.

While this cardinality maximisation approach operates directly on discrete sensor data, not every pixel in an image is geometrically meaningful. For example, there are no 3D points in a point-set that correspond to sky pixels in an image. Similarly, there may be no 3D points that correspond to pixels of dynamic objects such as vehicles, since they may not have been present in the scene when the point-set was captured. Consequently, a pre-processing step must be applied to extract 2D points that may correspond to elements in the point-set. Emphatically, this step does not seek to find putative 2D–3D correspondences, it seeks to isolate points in the image that may appear in the point-set, that is, structural elements of the scene.

The chapter is organised as follows: the problem is contextualised by summarising the relevant literature in Section 6.2; a robust cardinality maximisation objective function for 2D–3D alignment is introduced in Section 6.3; a parametrisation of the domain of 3D motions, a branching strategy and a derivation of the bounds are developed in Section 6.4; an algorithm is proposed for globally-optimal pose and correspondence in Section 6.5; and its performance is evaluated and discussed in Sections 6.6 and 6.7. Finally, the transferability of the theoretical framework is demonstrated by applying the main results to another robust objective function in Section 6.8.

6.2 Related Work

A large body of work exists for solving the 2D–3D registration problem when correspondences are provided. When the correspondences are known perfectly, Perspective-n-Point (PnP) solvers [Haralick et al., 1994; Lepetit et al., 2009; Kneip et al., 2011; Hesch and Roumeliotis, 2011] are able to estimate the pose of a calibrated camera given a set of noisy image points and their corresponding 3D points. When outliers are present in the correspondence set, the RANSAC framework [Fischler and Bolles, 1981; Chum and Matas, 2008] or robust global optimisation [Enqvist and Kahl, 2008; Li, 2009; Enqvist et al., 2012; Ask et al., 2013; Svärm et al., 2014; Enqvist et al., 2015; Svärm et al., 2016] can be used to find the inlier set. Alternatively, outlier removal schemes can make the problem more tractable [Sim and Hartley, 2006; Olsson et al., 2010; Yu et al., 2011; Chin et al., 2016]. Other methods develop sophisticated matching strategies to avoid outlier correspondences at the outset [Li et al., 2010; Sattler et al., 2011, 2012; Li et al., 2012]. However, these methods are only feasible when some correct correspondences are available. For this reason, they are often only practical for 3D models that have been constructed using stereopsis or Structure-from-Motion (SfM). These models associate an image feature with each 3D point, facilitating inter-modality feature matching. Generic point-sets do not have this property; a point may lie anywhere on the underlying surfaces in a laser scan, not just where strong image gradients occur. It should be observed that some of these approaches [Fischler and Bolles, 1981; Enqvist and Kahl, 2008] can be extended to the correspondence-free case by providing the algorithm with all possible permutations of the correspondence set. However, this leads to a hard combinatorial problem that quickly becomes infeasible.

When correspondences are unknown, the problem becomes more challenging. For the 2D–2D case, problems such as correspondence-free rigid registration [Besl and McKay, 1992; Breuel, 2003], SfM [Dellaert et al., 2000; Makadia et al., 2007; Lin et al., 2012] and relative camera pose [Fredriksson et al., 2016] have been addressed. For the 2D–3D case, solutions have been proposed for registering a collection of images

[Paudel et al., 2015b] or multiple cameras [Paudel et al., 2015a] to a 3D point-set. The more general problem, however, is pose estimation from a single image. David et al. [2004] proposed the SoftPOSIT algorithm to efficiently solve the simultaneous pose and correspondences problem from a single image. It alternates correspondence assignment using SoftAssign [Gold and Rangarajan, 1996] with an iterative pose update algorithm POSIT [Dementhon and Davis, 1995], applying deterministic annealing to encourage a large basin of convergence. A similar approach was taken by Moreno-Noguer et al. [2008] with the BlindPnP algorithm, which represents the pose prior as a Gaussian mixture model from which a Kalman filter is initialised for matching. It outperforms SoftPOSIT when large amounts of clutter, occlusions and repetitive patterns are present but is otherwise comparable. However, both of these approaches are susceptible to local optima, require good pose priors and cannot guarantee that the global optimum is attained.

Grimson [1990] applied a RANSAC-like approach to the correspondence-free case, removing the need for a pose prior, but the method was not optimal and quickly became intractable as the number of points increased. In contrast, globally-optimal methods find a camera pose that is guaranteed to be an optimiser of an objective function without requiring a pose prior, but tractability remains a challenge. A Branch-and-Bound (BB) [Land and Doig, 1960] strategy may be applied in these cases, for which bounds need to be derived. For example, Breuel [2003] used BB for 2D–2D registration problems, Hartley and Kahl [2009] for optimal relative pose estimation by bounding the group of 3D rotations, Li and Hartley [2007] for rotation-only 3D–3D registration, Olsson et al. [2009] for 3D–3D registration with known correspondences, Yang et al. [2016] for full 3D–3D registration and Campbell and Petersson [2016] for robust 3D–3D registration. While not optimal, Jurie [1999] used an approach similar to BB for 2D–3D alignment with a linear approximation of perspective projection. More recently, Brown et al. [2015] proposed a global and ϵ-suboptimal method using BB. It found a camera pose whose trimmed geometric error, the sum of angular distances between the bearings and their rotationally-closest 3D points, was within ϵ of the global minimum. While not susceptible to local minima, it required the inlier fraction to be specified, which can rarely be known in advance, in order to trim outliers. In addition, the use of ϵ annealing in their framework invalidates the guarantee of ϵ-suboptimality, since branches containing the correct pose may be pruned early.

The work presented in this chapter is the first globally-optimal inlier set cardinality maximisation solution to the simultaneous pose and correspondence problem. It is guaranteed to find the exact global optimum without requiring a pose prior and is robust to 2D and 3D outliers while avoiding the distortion of trimming.

6.3 Inlier Set Cardinality Maximisation

The cardinality of the inlier set is a robust objective function, discussed in detail in Section 3.4.4, that counts the number of inliers given a specific transformation of the data. A data element is classified as a member of the inlier or outlier sets with reference to a distance measure and a threshold θ. The objective function can operate directly on raw data representations without making assumptions about the structure of the data and is inherently robust to outliers without smoothing the objective function and thereby distorting or concealing the location of the global optimum.

For 2D–3D directional sensor data alignment, the inlier set consists of those bearing vectors that are within θ of any point in the point-set with respect to the angular distance metric, as shown in Figure 6.4. Let $\mathbf{p} \in \mathbb{R}^3$ be a 3D point and $\mathbf{f} \in \mathbb{R}^3$ be a bearing vector with unit norm, corresponding to a 2D point imaged by a calibrated camera. That is, $\mathbf{f} \propto \mathtt{K}^{-1}\hat{\mathbf{x}}$ where $\mathtt{K}$ is the matrix of intrinsic camera parameters and $\hat{\mathbf{x}}$ is the homogeneous image point. Given a set of points $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^{M}$ and bearing vectors $\mathcal{F} = \{\mathbf{f}_i\}_{i=1}^{N}$ and an inlier threshold θ, the objective is to find a rotation $\mathbf{R} \in SO(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$ that maximises the cardinality ν of the inlier set $\mathcal{S}_I$

$$\nu^* = \max_{\mathbf{R},\,\mathbf{t}} |\mathcal{S}_I| \tag{6.1}$$

$$\mathcal{S}_I = \{\mathbf{f} \in \mathcal{F} \mid \exists \mathbf{p} \in \mathcal{P} : \angle(\mathbf{f}, \mathbf{R}(\mathbf{p} - \mathbf{t})) \leqslant \theta\} \tag{6.2}$$

where $\angle(\cdot,\cdot)$ denotes the angular distance between vectors. An equivalent formulation is given by

$$\nu^* = \max_{\mathbf{R},\,\mathbf{t}} f(\mathbf{R}, \mathbf{t}) \tag{6.3}$$

$$\nu = f(\mathbf{R}, \mathbf{t}) = \sum_{\mathbf{f} \in \mathcal{F}} \max_{\mathbf{p} \in \mathcal{P}} \mathbf{1}(\theta - \angle(\mathbf{f}, \mathbf{R}(\mathbf{p} - \mathbf{t}))) \tag{6.4}$$

where $\mathbf{1}(x) \triangleq \mathbf{1}_{\mathbb{R}_{\geq 0}}(x)$ is the indicator function that has the value 1 for all elements of the non-negative real numbers and the value 0 otherwise. All correspondences $(\mathbf{f}_i, \mathbf{p}_j)$ with respect to θ can be found from the optimal transformation parameters $\mathbf{R}^*$ and $\mathbf{t}^*$ by identifying all pairs for which $\angle(\mathbf{f}_i, \mathbf{R}^*(\mathbf{p}_j - \mathbf{t}^*)) \leqslant \theta$.

An important consideration is which asymmetric inlier measure to apply. If the number of 3D point inliers were maximised, a set of degenerate poses would be found, where all 3D points were inliers with respect to a single bearing vector. These degenerate poses position the camera far from the point-set such that all points fall within the inlier cone of one bearing vector, as shown in Figure 3.5. Instead, the number of bearing vector inliers is maximised. However, this can also result in degenerate poses, where the camera is positioned close to a region of 3D points, such as a wall. If the

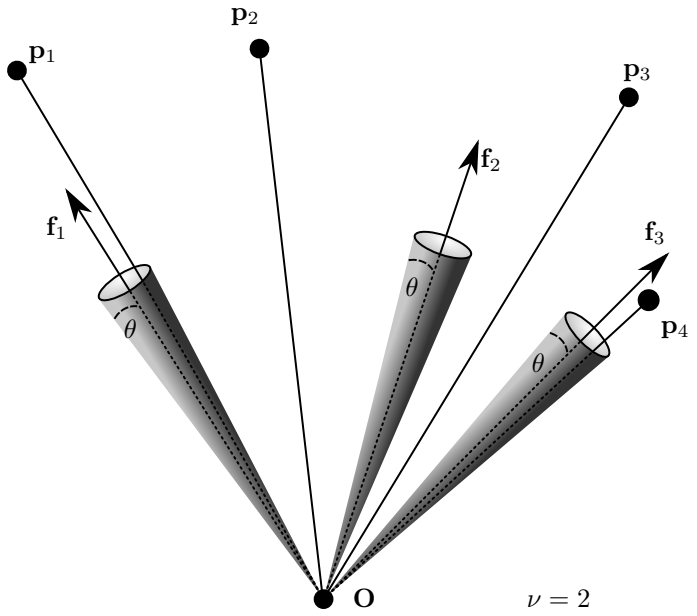
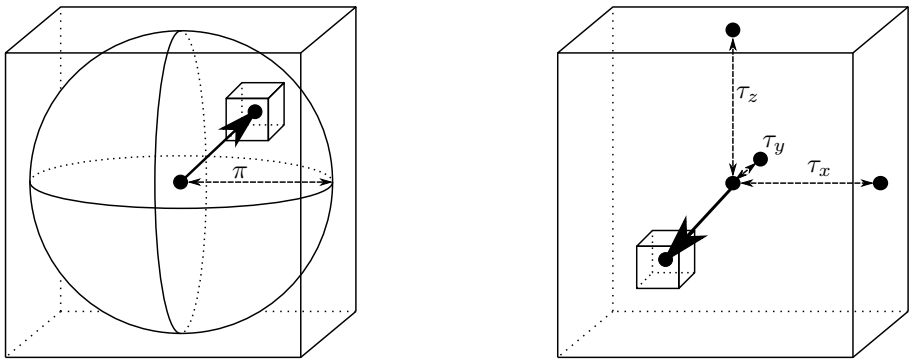


Figure 6.4: Definition of the inlier set for 2D–3D directional sensor data alignment. The inlier set is defined as the set of those bearing vectors in $\mathcal{F}$ that are within θ of any point in $\mathcal{P}$ with respect to the angular distance metric.

region is sufficiently densely-sampled and the camera field-of-view is less than 180°, all bearing vectors can become inliers with respect to several of the 3D points. However, this is much less common in real datasets, particularly for panoramic imagery, and can be avoided by setting a minimum point-to-camera distance. The relative advantages and disadvantages of symmetric objective functions will be discussed in Section 6.7.

6.4 Branch-and-Bound

To solve the highly non-convex cardinality maximisation problem (6.1), the global optimisation technique of Branch-and-Bound (BB) [Land and Doig, 1960] may be applied. To do so, a suitable means of parametrising and branching (partitioning) the function domain must be found, as well as an efficient way to calculate upper and lower bounds of the function for each branch that converge as the branch size tends to zero. While the bounds need to be computationally efficient to calculate, the time and memory efficiency of the algorithm also depends on how tight the bounds are, since tighter bounds reduce the search space quicker by allowing suboptimal branches to be pruned. These two factors are generally in opposition and must be optimised together. Many more details on the branch-and-bound algorithm and its application to the geometric alignment problem can be found in Section 3.7.



(a) Rotation Domain Ω_r (b) Translation Domain Ω_t

Figure 6.5: Parametrisation of $SE(3)$. (a) The rotation space $SO(3)$ is parametrised by angle-axis 3-vectors in a solid radius-π ball. (b) The translation space $\mathbb{R}^3$ is parametrised by 3-vectors bounded by a cuboid with half-widths $[\tau_x, \tau_y, \tau_z]$. The domain is branched into sub-cuboids as shown using nested octree data structures.

6.4.1 Parametrising and Branching the Domain

To find a globally-optimal solution, the cardinality of the inlier set $\mathcal{S}_I$ must be optimised over the domain of 3D motions, that is, the group $SE(3) = SO(3) \times \mathbb{R}^3$. However, the space of these transformations is unbounded. Therefore, to apply the BB paradigm, the space of translations is restricted to be within the bounded set Ω_t. For a suitably large set, it is reasonable to assume that the camera centre lies within Ω_t. That is, the camera can be assumed to be a finite distance from the 3D points. The domains are shown in Figure 6.5.

Rotation space $SO(3)$ is minimally parametrised with angle-axis 3-vectors $\mathbf{r}$ with rotation angle $\|\mathbf{r}\|$ and rotation axis $\mathbf{r}/\|\mathbf{r}\|$. The notation $\mathbf{R}_\mathbf{r} \in SO(3)$ is used to denote the rotation matrix obtained from the matrix exponential map of the skew-symmetric matrix $[\mathbf{r}]_\times$ induced by $\mathbf{r}$. The Rodrigues' rotation formula (3.10) can be used to efficiently calculate this mapping. See Section 3.1.2 for more details. Using this parametrisation, the space of all 3D rotations can be represented as a solid ball of radius π in $\mathbb{R}^3$. The mapping is one-to-one on the interior of the π-ball and two-to-one on the surface. For ease of manipulation, the 3D cube circumscribing the π-ball is used as the rotation domain Ω_r, as in Li and Hartley [2007].

Translation space $\mathbb{R}^3$ is parametrised with 3-vectors in a bounded domain chosen as the cuboid Ω'_t containing the bounding box of $\mathcal{P}$. If the camera is known to be inside the 3D scene, Ω'_t can be set to the bounding box, otherwise it is set to an expansion of the bounding box. To avoid the non-physical case where a 3D point is located within a very small value ζ of the camera centre, the translation domain is restricted such that $\Omega_t = \Omega'_t \cap \{\mathbf{t} \in \mathbb{R}^3 \mid \|\mathbf{p} - \mathbf{t}\| \geqslant \zeta, \forall \mathbf{p} \in \mathcal{P}\}$.

In this implementation of BB, the domain is branched into sub-cuboids using nested octree data structures. They are defined as

$$\mathcal{C}(\mathbf{x}_0, \boldsymbol{\delta}) = \{\mathbf{x} \in \mathbb{R}^3 \mid \mathbf{e}_i^\intercal(\mathbf{x} - \mathbf{x}_0) \in [-\delta_i, \delta_i], i = 1, 2, 3\} \tag{6.5}$$

where $\boldsymbol{\delta}$ is the vector of half side-lengths of the cuboid and $\mathbf{e}_i$ is the i^{th} standard basis vector. To simplify the notation, $\mathcal{C}_r = \mathcal{C}(\mathbf{r}_0, \boldsymbol{\delta}_r)$ and $\mathcal{C}_t = \mathcal{C}(\mathbf{t}_0, \boldsymbol{\delta}_t)$ is used for the rotation and translation sub-cuboids respectively.

6.4.2 Bounding the Branches

The success of a branch-and-bound algorithm is predicated on the quality of its bounds. For inlier set maximisation, the objective function (6.4) needs to be bounded within a transformation domain. Some preparatory material is now presented.

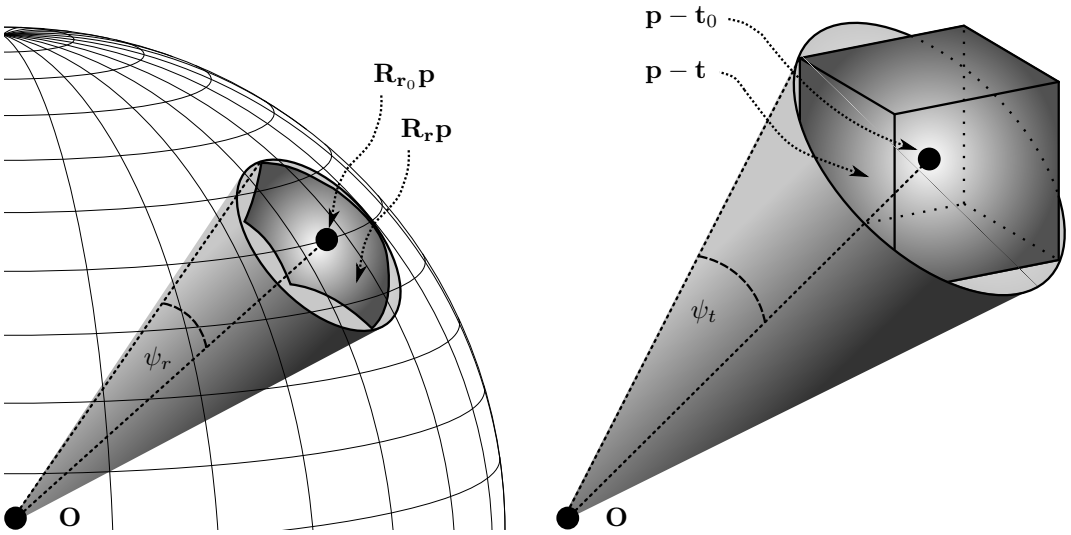
Uncertainty Angle Bounds

If a branch contained a single rotation or translation, then the new position of a point transformed by that branch would be known with certainty. However, each branch contains a set of (infinitely) many different rotations or translations. Transforming a point by a contiguous set of rotations or translations induces a transformation region, shown in Figure 6.6. The transformation region lies on a sphere for rotations and in $\mathbb{R}^3$ for translations.

To bound the objective function on a branch, a bound on the maximum or worst-case angular deviation needs to be calculated, with respect to some arbitrary reference transformation in the branch. For simplicity, the reference transformation is the rotation or translation associated with the centre of the cuboidal branch. In this work, the maximum deviation is termed the *uncertainty angle* because it expresses how far from the reference transformation the optimal in-branch transformation might be.

The uncertainty angles induced by a rotation and translation sub-cuboid on a point $\mathbf{p}$ are shown in Figure 6.6. The transformed point lies within a cone with aperture angle equal to the sum of the rotation and translation uncertainty angles.

A weak bound on the uncertainty angle due to rotation was derived in Hartley and Kahl [2009] using a proof, summarised in Lemma 5.1, that the angle between two rotated vectors is less than the Euclidean distance between their rotations' angle-axis representations in $\mathbb{R}^3$. From this, a bound on the maximum angle between a vector $\mathbf{p}$ rotated by $\mathbf{r}_0$ and $\mathbf{p}$ rotated by $\mathbf{r} \in \mathcal{C}_r$ for a cube of rotation angle-axis vectors $\mathcal{C}_r$ can be found. For reference, the bound is reproduced here.



(a) Rotation Uncertainty Angle (b) Translation Uncertainty Angle

Figure 6.6: Uncertainty angles induced by rotation and translation sub-cuboids. (a) Rotation uncertainty angle ψ_r for $\mathcal{C}_r$. The optimal rotation of $\mathbf{p}$ may be anywhere within the umbrella-shaped region on the sphere, which is entirely contained by the cone defined by $\mathbf{R}_{\mathbf{r}_0}\mathbf{p}$ and ψ_r. (b) Translation uncertainty angle ψ_t for $\mathcal{C}_t$. The optimal translation of $\mathbf{p}$ may be anywhere within the cuboidal region, which is entirely contained by the cone defined by $\mathbf{p}-\mathbf{t}_0$ and ψ_t.

Lemma 6.1. *(Weak rotation uncertainty angle bound) Given a 3D point* $\mathbf{p}$ *and a rotation cube* $\mathcal{C}_r$ *of half side-length* δ_r *centred at* $\mathbf{r}_0$*, then* $\forall \mathbf{r} \in \mathcal{C}_r$*,*

$$\angle(\mathbf{R}_{\mathbf{r}}\mathbf{p}, \mathbf{R}_{\mathbf{r}_0}\mathbf{p}) \leqslant \min\{\sqrt{3}\delta_r, \pi\} \triangleq \psi_r^w(\mathcal{C}_r). \tag{6.6}$$

Proof. Inequality (6.6) can be derived as follows:

$$\angle(\mathbf{R}_{\mathbf{r}}\mathbf{p}, \mathbf{R}_{\mathbf{r}_0}\mathbf{p}) \leqslant \min\{\|\mathbf{r}-\mathbf{r}_0\|, \pi\} \tag{6.7}$$

$$\leqslant \min\{\sqrt{3}\delta_r, \pi\} \tag{6.8}$$

where (6.7) follows from Lemma 5.1 and the maximum possible angle between points on a sphere and (6.8) follows from $\max \|\mathbf{r}-\mathbf{r}_0\| = \sqrt{3}\delta_r$, the half space diagonal of the rotation cube, for $\mathbf{r} \in \mathcal{C}_r$. □

However, a tighter bound can be found by observing that a point rotated about an axis parallel to the position vector of the point is not displaced. Therefore, equally-sized rotation cubes will displace a point by differing amounts depending on the angle between the point vector and the angle-axis vectors. To exploit this, the angle $\angle(\mathbf{R}_{\mathbf{r}}\mathbf{p}, \mathbf{R}_{\mathbf{r}_0}\mathbf{p})$ is instead maximised over the surface $\mathcal{S}_r$ of the cube $\mathcal{C}_r$ as follows.

Lemma 6.2. *(Rotation uncertainty angle bound) Given a 3D point* $\mathbf{p}$ *and a rotation cube* $\mathcal{C}_r$ *centred at* $\mathbf{r}_0$ *with surface* $\mathcal{S}_r$*, then* $\forall \mathbf{r} \in \mathcal{C}_r$*,*

$$\angle(\mathbf{R_r p}, \mathbf{R}_{\mathbf{r}_0}\mathbf{p}) \leqslant \min\left\{\max_{\mathbf{r}\in\mathcal{S}_r} \angle(\mathbf{R_r p}, \mathbf{R}_{\mathbf{r}_0}\mathbf{p}), \pi\right\} \triangleq \psi_r(\mathbf{p}, \mathcal{C}_r). \tag{6.9}$$

Proof. Inequality (6.9) can be derived as follows:

$$\angle(\mathbf{R_r p}, \mathbf{R}_{\mathbf{r}_0}\mathbf{p}) \leqslant \min\left\{\max_{\mathbf{r}\in\mathcal{C}_r} \angle(\mathbf{R_r p}, \mathbf{R}_{\mathbf{r}_0}\mathbf{p}), \pi\right\} \tag{6.10}$$

$$= \min\left\{\max_{\mathbf{r}\in\mathcal{S}_r} \angle(\mathbf{R_r p}, \mathbf{R}_{\mathbf{r}_0}\mathbf{p}), \pi\right\} \tag{6.11}$$

where (6.10) follows from maximising the angle over the rotation cube $\mathcal{C}_r$ and capping the angle at the maximum possible angle between points on a sphere and (6.11) is a consequence of the order-preserving mapping, with respect to the radial angle, from the convex cube of angle-axis vectors to the spherical surface patch (see Figure 6.6(a)), since the mapping is obtained by projecting from the centre of the sphere to the surface of the sphere. See Section 6.5.4 for further details. □

A weak bound on the uncertainty angle due to translation was derived in Brown et al. [2015] by enclosing the translation cuboid within a circumsphere of radius ρ_t. From this, a bound on the maximum angle between a vector $\mathbf{p}$ translated by $\mathbf{t}_0$ and $\mathbf{p}$ translated by $\mathbf{t} \in \mathcal{C}_t$ for a cube of translation vectors $\mathcal{C}_t$ can be found. For reference, the bound is reproduced here.

Lemma 6.3. *(Weak translation uncertainty angle bound) Given a 3D point* $\mathbf{p}$ *and a translation cuboid* $\mathcal{C}_t$ *centred at* $\mathbf{t}_0$ *with half space diagonal* ρ_t*, then* $\forall \mathbf{t} \in \mathcal{C}_t$*,*

$$\angle(\mathbf{p}-\mathbf{t}, \mathbf{p}-\mathbf{t}_0) \leqslant \begin{cases} \arcsin\left(\dfrac{\rho_t}{\|\mathbf{p}-\mathbf{t}_0\|}\right) & \text{if } \|\mathbf{p}-\mathbf{t}_0\| \geqslant \rho_t \\ \pi & \text{else} \end{cases} \triangleq \psi_t^w(\mathbf{p}, \mathcal{C}_t). \tag{6.12}$$

Proof. As given in Brown et al. [2015]. □

However, a tighter bound can be found by using the cuboid of translated points (Figure 6.6(b)) directly instead of its circumsphere. When the cuboid does not contain the origin, the angle can be found by maximising over the cuboid vertices.

Lemma 6.4. *(Translation uncertainty angle bound) Given a 3D point* $\mathbf{p}$ *and a translation cuboid* $\mathcal{C}_t$ *centred at* $\mathbf{t}_0$ *with vertices* $\mathcal{V}_t$*, then* $\forall \mathbf{t} \in \mathcal{C}_t$*,*

$$\angle(\mathbf{p}-\mathbf{t}, \mathbf{p}-\mathbf{t}_0) \leqslant \begin{cases} \max\limits_{\mathbf{t}\in\mathcal{V}_t} \angle(\mathbf{p}-\mathbf{t}, \mathbf{p}-\mathbf{t}_0) & \text{if } \mathbf{p} \notin \mathcal{C}_t \\ \pi & \text{else} \end{cases} \triangleq \psi_t(\mathbf{p}, \mathcal{C}_t). \tag{6.13}$$

Proof. Observe that for $\mathbf{p} \in \mathcal{C}_t$, the cuboid containing all translated points $\mathbf{p} - \mathbf{t}$ also contains the origin. Therefore the vectors $\mathbf{p}-\mathbf{t}$ and $\mathbf{p}-\mathbf{t}_0$ can be antiparallel (oppositely directed) and thus the maximum angle is π. For $\mathbf{p} \notin \mathcal{C}_t$,

$$\angle(\mathbf{p}-\mathbf{t}, \mathbf{p}-\mathbf{t}_0) \leqslant \max_{\mathbf{t}\in\mathcal{C}_t} \angle(\mathbf{p}-\mathbf{t}, \mathbf{p}-\mathbf{t}_0) \tag{6.14}$$

$$= \max_{\mathbf{t}\in\mathcal{V}_t} \angle(\mathbf{p}-\mathbf{t}, \mathbf{p}-\mathbf{t}_0) \tag{6.15}$$

where (6.14) follows from maximising the angle over the translation cuboid $\mathcal{C}_t$ and (6.15) follows from the convexity of the angle function in this domain. The maximum of a convex function over a convex set must occur at one of its extreme points, which for this set are the vertices. Geometrically, the cuboid $\mathbf{p} - \mathbf{t}$ for $\mathbf{t} \in \mathcal{C}_t$ and $\mathbf{p} \notin \mathcal{C}_t$ projects to a spherical hexagon on the unit sphere. The geodesic from an arbitrary fixed point in the hexagon to any point in the hexagon is maximised when the variable point is a vertex of the hexagon. □

Objective Function Bounds

The preceding lemmas are used to bound the maximum of the objective function (6.4) within a transformation domain $\mathcal{C}_r \times \mathcal{C}_t$. A lower bound can be found by evaluating the function at any transformation in the branch. In this case, the transformation at the centre of the rotation and translation cuboids is convenient and quick to evaluate.

Theorem 6.1. *(Lower bound) For the transformation domain $\mathcal{C}_r \times \mathcal{C}_t$ centred at $(\mathbf{r}_0, \mathbf{t}_0)$, the lower bound of the inlier set cardinality can be chosen as*

$$\underline{\nu} \triangleq f(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}_0). \tag{6.16}$$

Proof. The validity of the lower bound follows from

$$f(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}_0) \leqslant \max_{\substack{\mathbf{r}\in\mathcal{C}_r \\ \mathbf{t}\in\mathcal{C}_t}} f(\mathbf{R}_{\mathbf{r}}, \mathbf{t}). \tag{6.17}$$

That is, the function value at a specific point within the domain is less than or equal to the maximum within the domain. □

An upper bound on the objective function within a transformation domain $\mathcal{C}_r \times \mathcal{C}_t$ can be found using the bounds on the uncertainty angles ψ_r and ψ_t. The geometric intuition for the upper bound is that it relaxes the inlier threshold by the two uncertainty angles, creating a more permissive inlier set, as shown in Figure 6.7.

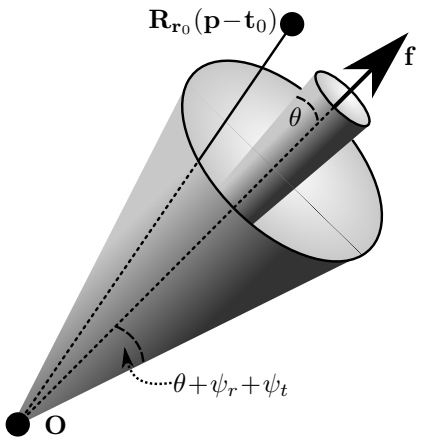


Figure 6.7: Geometric intuition for the upper bound. The inlier threshold is relaxed by the two uncertainty angles ψ_r and ψ_t, creating a more permissive inlier set and hence an upper bound on the cardinality.

Theorem 6.2. *(Upper bound) For the transformation domain $\mathcal{C}_r \times \mathcal{C}_t$ centred at $(\mathbf{r}_0, \mathbf{t}_0)$, the upper bound of the inlier set cardinality can be chosen as*

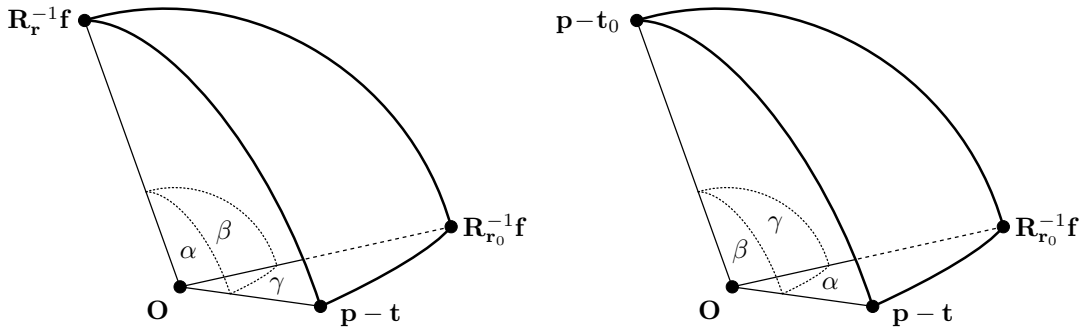
$$\bar{\nu} \triangleq \sum_{\mathbf{f}\in\mathcal{F}} \max_{\mathbf{p}\in\mathcal{P}} \mathbf{1}\Big(\theta - \angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p}-\mathbf{t}_0)) + \psi_r(\mathbf{f},\mathcal{C}_r) + \psi_t(\mathbf{p},\mathcal{C}_t)\Big). \tag{6.18}$$

Proof. Observe that $\forall(\mathbf{r},\mathbf{t}) \in (\mathcal{C}_r \times \mathcal{C}_t)$,

$$\begin{aligned}
\angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}}(\mathbf{p}-\mathbf{t})) &= \angle(\mathbf{R}_{\mathbf{r}}^{-1}\mathbf{f}, \mathbf{p}-\mathbf{t}) && (6.19)\\
&\geqslant \angle(\mathbf{R}_{\mathbf{r}_0}^{-1}\mathbf{f}, \mathbf{p}-\mathbf{t}) - \angle(\mathbf{R}_{\mathbf{r}}^{-1}\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}^{-1}\mathbf{f}) && (6.20)\\
&\geqslant \angle(\mathbf{R}_{\mathbf{r}_0}^{-1}\mathbf{f}, \mathbf{p}-\mathbf{t}_0) - \angle(\mathbf{R}_{\mathbf{r}}^{-1}\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}^{-1}\mathbf{f}) - \angle(\mathbf{p}-\mathbf{t}, \mathbf{p}-\mathbf{t}_0) && (6.21)\\
&\geqslant \angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p}-\mathbf{t}_0)) - \psi_r(\mathbf{f},\mathcal{C}_r) - \psi_t(\mathbf{p},\mathcal{C}_t) && (6.22)
\end{aligned}$$

where (6.20) and (6.21) follow from the triangle inequality in spherical geometry (see Figure 6.8) and (6.22) follows from Lemmas 6.2 and 6.4. Substituting (6.22) into (6.4) completes the proof. □

By inspecting the translation component of Theorem 6.2 and removing one of the two applications of the triangle inequality (6.21), a tighter upper bound can be found. A similar approach cannot be taken for the rotation component since $\mathbf{R}_{\mathbf{r}}^{-1}\mathbf{f}$ is a complex surface due to the nonlinear conversion from angle-axis to rotation matrix representations. To reduce computation, it is only necessary to evaluate this tighter bound when $\angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p}-\mathbf{t}_0)) \leqslant \theta + \psi_r(\mathbf{f},\mathcal{C}_r) + \psi_t(\mathbf{p},\mathcal{C}_t)$, since otherwise the point is definitely an outlier and does not need to be investigated further.



(a) Triangle inequality for (6.20) (b) Triangle inequality for (6.21)

Figure 6.8: The triangle inequality in spherical geometry, given by $\gamma \leqslant \alpha + \beta$. (a) First inequality: $\angle\big(\mathbf{R}_{\mathbf{r}_0}^{-1}\mathbf{f}, \mathbf{p}-\mathbf{t}\big) \leqslant \angle\big(\mathbf{R}_{\mathbf{r}}^{-1}\mathbf{f}, \mathbf{p}-\mathbf{t}\big) + \angle\big(\mathbf{R}_{\mathbf{r}}^{-1}\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}^{-1}\mathbf{f}\big)$. (b) Second inequality: $\angle\big(\mathbf{R}_{\mathbf{r}_0}^{-1}\mathbf{f}, \mathbf{p}-\mathbf{t}_0\big) \leqslant \angle\big(\mathbf{R}_{\mathbf{r}_0}^{-1}\mathbf{f}, \mathbf{p}-\mathbf{t}\big) + \angle(\mathbf{p}-\mathbf{t}, \mathbf{p}-\mathbf{t}_0)$. The transformed points have been normalised to lie on the unit sphere.

Theorem 6.3. *(Tighter upper bound) For the transformation domain $\mathcal{C}_r \times \mathcal{C}_t$ centred at $(\mathbf{r}_0, \mathbf{t}_0)$, the upper bound of the inlier set cardinality can be chosen as*

$$\bar{\nu} \triangleq \sum_{\mathbf{f}\in\mathcal{F}} \max_{\mathbf{p}\in\mathcal{P}} \Gamma(\mathbf{f}, \mathbf{p}) \tag{6.23}$$

where

$$\Gamma(\mathbf{f}, \mathbf{p}) = \max_{\mathbf{t}\in\mathcal{C}_t} \mathbf{1}\big(\theta - \angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p}-\mathbf{t})) + \psi_r(\mathbf{f}, \mathcal{C}_r)\big). \tag{6.24}$$

Proof. Observe that $\forall(\mathbf{r}, \mathbf{t}) \in (\mathcal{C}_r \times \mathcal{C}_t)$,

$$\begin{aligned}
\mathbf{1}\big(\theta - \angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}}(\mathbf{p}-\mathbf{t}))\big) &= \mathbf{1}\big(\theta - \angle(\mathbf{R}_{\mathbf{r}}^{-1}\mathbf{f}, \mathbf{p}-\mathbf{t})\big) && (6.25)\\
&\leqslant \mathbf{1}\big(\theta - \angle\big(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p}-\mathbf{t})\big) + \angle\big(\mathbf{R}_{\mathbf{r}}^{-1}\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}^{-1}\mathbf{f}\big)\big) && (6.26)\\
&\leqslant \max_{\mathbf{t}\in\mathcal{C}_t} \mathbf{1}\big(\theta - \angle\big(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p}-\mathbf{t})\big) + \psi_r(\mathbf{f}, \mathcal{C}_r)\big) && (6.27)
\end{aligned}$$

where (6.26) follows from the triangle inequality in spherical geometry (see Figure 6.8) and (6.27) follows from Lemma 6.2 and maximising over $\mathbf{t}$. Substituting (6.27) into (6.4) completes the proof. See Section 6.5.4 for implementation details. □

Comparison of Uncertainty Angle Bounds

The weaker sphere-based uncertainty angle bounds ψ_r^w and ψ_t^w given in (6.6) and (6.12) appeared originally in Hartley and Kahl [2009] and Brown et al. [2015] respectively. The tighter cuboid-based uncertainty angle bounds ψ_r and ψ_t given in (6.9) and (6.13) are original to this work and lead to tighter bounds on the objective function. This can be seen from Theorems 6.1 and 6.2, where it is clear that $\bar{\nu} - \underline{\nu}$ is smaller when

the uncertainty angle bounds ψ_r and ψ_t are smaller. The proofs that $\psi_r \leqslant \psi_r^w$ and $\psi_t \leqslant \psi_t^w$ will now be given.

Lemma 6.5. *(Rotation uncertainty angle bounds inequality) Given a 3D point* $\mathbf{p}$ *and a rotation cube* $\mathcal{C}_r$ *centred at* $\mathbf{r}_0$ *with surface* $\mathcal{S}_r$ *and half side-length* δ_r*, then*

$$\psi_r(\mathbf{p}, \mathcal{C}_r) \leqslant \psi_r^w(\mathcal{C}_r). \tag{6.28}$$

Proof. Inequality (6.28) can be derived as follows:

$$\begin{aligned}
\psi_r(\mathbf{p}, \mathcal{C}_r) &= \min\Big\{\max_{\mathbf{r}\in\mathcal{S}_r} \angle(\mathbf{R}_\mathbf{r}\mathbf{p}, \mathbf{R}_{\mathbf{r}_0}\mathbf{p}), \pi\Big\} && (6.29)\\
&= \min\{\angle(\mathbf{R}_{\mathbf{r}^*}\mathbf{p}, \mathbf{R}_{\mathbf{r}_0}\mathbf{p}), \pi\} && (6.30)\\
&\leqslant \min\Big\{\sqrt{3}\delta_r, \pi\Big\} && (6.31)\\
&= \psi_r^w(\mathcal{C}_r) && (6.32)
\end{aligned}$$

where (6.30) replaces the maximisation with an arg max rotation $\mathbf{r}^*$ and (6.31) follows from Lemma 6.1. □

Lemma 6.6. *(Translation uncertainty angle bounds inequality) Given a 3D point* $\mathbf{p}$ *and a translation cuboid* $\mathcal{C}_t$ *centred at* $\mathbf{t}_0$ *with vertices* $\mathcal{V}_t$ *and half space diagonal* ρ_t*, then*

$$\psi_t(\mathbf{p}, \mathcal{C}_t) \leqslant \psi_t^w(\mathbf{p}, \mathcal{C}_t). \tag{6.33}$$

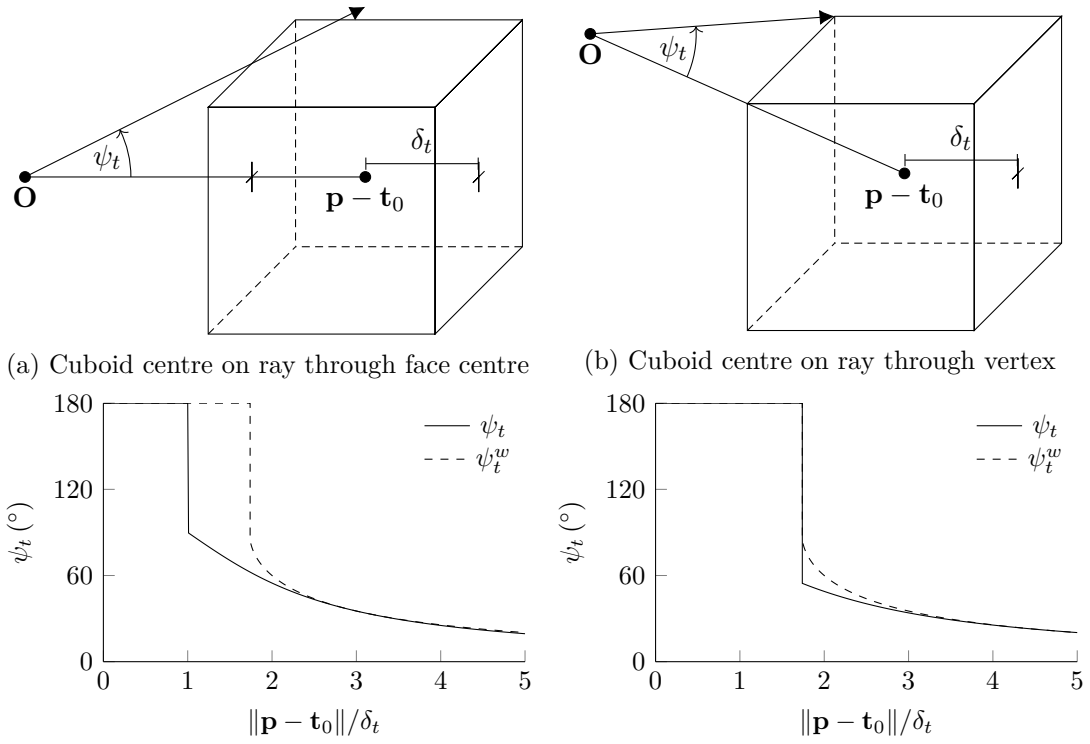
Proof. Inequality (6.33) can be derived as follows. For $\|\mathbf{p} - \mathbf{t}_0\| \geqslant \rho_t$, which is guaranteed for $\rho_t \leqslant \zeta$,

$$\begin{aligned}
\psi_t(\mathbf{p}, \mathcal{C}_t) &= \max_{\mathbf{t}\in\mathcal{V}_t} \angle(\mathbf{p} - \mathbf{t}, \mathbf{p} - \mathbf{t}_0) && (6.34)\\
&\leqslant \max_{\mathbf{t}\in S_t^2} \angle(\mathbf{p} - \mathbf{t}, \mathbf{p} - \mathbf{t}_0) && (6.35)\\
&= \arcsin\Big(\frac{\rho_t}{\|\mathbf{p} - \mathbf{t}_0\|}\Big) && (6.36)\\
&= \psi_t^w(\mathbf{p}, \mathcal{C}_t) && (6.37)
\end{aligned}$$

where (6.35) follows from maximising the angle over the circumsphere S_t^2 of the cuboid instead of the vertices and (6.36) is shown in Brown et al. [2015] with ρ_t being the half space diagonal of the translation sub-cuboid $\mathcal{C}_t$. For the alternate case $\|\mathbf{p} - \mathbf{t}_0\| < \rho_t$,

$$\psi_t(\mathbf{p}, \mathcal{C}_t) \leqslant \pi = \psi_t^w(\mathbf{p}, \mathcal{C}_t). \tag{6.38}$$

□



(a) Cuboid centre on ray through face centre

(b) Cuboid centre on ray through vertex

(c) Uncertainty angle vs scaled distance to cuboid centre (ray through face centre)

(d) Uncertainty angle vs scaled distance to cuboid centre (ray through vertex)

Figure 6.9: Comparison of translation uncertainty angle bounds when the centre $\mathbf{p} - \mathbf{t}_0$ of the translation cuboid $\mathbf{p} - \mathbf{t}$ lies along a ray from the origin towards (a) any face centre and (b) any vertex. For clarity, the translation cuboids are cubes with half side-lengths equal to δ_t. (c)–(d) The novel bound ψ_t is tighter across the entire domain in both cases.

Therefore, the weaker uncertainty angle bounds ψ_r^w and ψ_t^w are larger than the uncertainty angle bounds ψ_r and ψ_t for a given rotation or translation sub-cuboid. Consequently, the objective function bounds using ψ_r and ψ_t are tighter than those using ψ_r^w and ψ_t^w. More specifically, the maximum angular difference between ψ_t and ψ_t^w is $\pi - \arctan(\sqrt{2}/\sqrt{3}-1) = 117°$ for translation cuboids with equal side-lengths (cubes). The difference is even more pronounced for cuboids with non-equal side-lengths. Figure 6.9 compares both translation uncertainty angle bounds across a range of values.

6.5 The GOPAC Algorithm

The Globally-Optimal Pose And Correspondences (GOPAC) algorithm for a calibrated camera is outlined in Algorithms 6.1 and 6.2.

Algorithm 6.1 GOPAC: a branch-and-bound algorithm for globally-optimal camera pose and correspondence estimation

Input: bearing vector set $\mathcal{F}$, point-set $\mathcal{P}$, inlier threshold θ, initial domains Ω_r and Ω_t
Output: optimal number of inliers ν^*, camera pose $(\mathbf{r}^*, \mathbf{t}^*)$, 2D–3D correspondences

$\nu^* \leftarrow 0$
Add translation domain Ω_t to priority queue Q_t
loop
  Update greatest upper bound $\bar{\nu}_t$ from Q_t
  Remove cuboid $\mathcal{C}_t$ with greatest width δ_{tx} from Q_t
  if $\nu^* \geqslant \bar{\nu}_t$ **then** terminate
  for all sub-cuboids $\mathcal{C}_{ti} \in \mathcal{C}_t$ **do**
    $(\underline{\nu}_{ti}, \mathbf{r}) \leftarrow \mathrm{RBB}(\nu^*, \mathbf{t}_{0i}, \psi_t = 0)$
    if $\nu^* < 2\underline{\nu}_{ti}$ **then** $(\nu^*, \mathbf{r}^*, \mathbf{t}^*) \leftarrow \mathrm{Refine}(\mathbf{r}, \mathbf{t}_{0i})$
    $(\bar{\nu}_{ti}, \emptyset) \leftarrow \mathrm{RBB}(\nu^*, \mathbf{t}_{0i}, \psi_t)$
    if $\nu^* < \bar{\nu}_{ti}$ **then** add $\mathcal{C}_{ti}$ to queue Q_t

Algorithm 6.2 RBB: a rotation search subroutine for GOPAC

Input: bearing vector set $\mathcal{F}$, point-set $\mathcal{P}$, inlier threshold θ, initial domain Ω_r, best-so-far cardinality ν^*, translation $\mathbf{t}_0$, translation uncertainty ψ_t
Output: optimal number of inliers ν_r^*, rotation $\mathbf{r}^*$

$\nu_r^* \leftarrow \nu^*$
Add rotation domain Ω_r to priority queue Q_r
loop
  Remove cube $\mathcal{C}_r$ with greatest upper bound $\bar{\nu}_r$ from Q_r
  if $\nu_r^* \geqslant \bar{\nu}_r$ **then** terminate
  for all sub-cubes $\mathcal{C}_{ri} \in \mathcal{C}_r$ **do**
    Calculate $\underline{\nu}_{ri}$ by (6.39) or (6.41) with parameters $\mathbf{r}_{0i}$, $\mathbf{t}_0$, ψ_t
    if $\nu_r^* < \underline{\nu}_{ri}$ **then** $\nu_r^* \leftarrow \underline{\nu}_{ri}, \mathbf{r}^* \leftarrow \mathbf{r}_0$
    Calculate $\bar{\nu}_{ri}$ by (6.40) or (6.42) with parameters $\mathbf{r}_{0i}$, $\mathbf{t}_0$, ψ_t, ψ_r
    if $\nu_r^* < \bar{\nu}_{ri}$ **then** add $\mathcal{C}_{ri}$ to queue Q_r

6.5.1 Nested Branch-and-Bound Structure

As in Yang et al. [2016], a nested branch-and-bound structure is employed for computational efficiency. In the outer breadth-first BB search, upper and lower bounds are found for each translation cuboid $\mathcal{C}_t \in \Omega_t$ by running an inner best-first BB search over rotation space $SO(3)$ (denoted RBB). The upper bound $\bar{\nu} \triangleq \bar{\nu}_t$ (6.18) for the cuboid $\mathcal{C}_t$ is found by running RBB until convergence with the following bounds

$$\underline{\nu}_r \triangleq \sum_{\mathbf{f}\in\mathcal{F}} \max_{\mathbf{p}\in\mathcal{P}} \mathbf{1}(\theta - \angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p} - \mathbf{t}_0)) + \psi_t(\mathbf{p}, \mathcal{C}_t)) \tag{6.39}$$

$$\bar{\nu}_r \triangleq \sum_{\mathbf{f}\in\mathcal{F}} \max_{\mathbf{p}\in\mathcal{P}} \mathbf{1}(\theta - \angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p} - \mathbf{t}_0)) + \psi_t(\mathbf{p}, \mathcal{C}_t) + \psi_r(\mathbf{f}, \mathcal{C}_r)). \tag{6.40}$$

The tighter upper bound (6.23) instead uses

$$\underline{\nu}_r \triangleq \sum_{\mathbf{f}\in\mathcal{F}} \max_{\mathbf{p}\in\mathcal{P}} \max_{\mathbf{t}\in\mathcal{C}_t} \mathbf{1}(\theta - \angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p}-\mathbf{t}))) \tag{6.41}$$

$$\bar{\nu}_r \triangleq \sum_{\mathbf{f}\in\mathcal{F}} \max_{\mathbf{p}\in\mathcal{P}} \max_{\mathbf{t}\in\mathcal{C}_t} \mathbf{1}(\theta - \angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p}-\mathbf{t})) + \psi_r(\mathbf{f}, \mathcal{C}_r)). \tag{6.42}$$

The lower bound $\underline{\nu} \triangleq \underline{\nu}_t$ (6.16) is found by running RBB using bounds (6.39) and (6.40) with ψ_t set to zero. That is,

$$\underline{\nu}_r \triangleq \sum_{\mathbf{f}\in\mathcal{F}} \max_{\mathbf{p}\in\mathcal{P}} \mathbf{1}(\theta - \angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p}-\mathbf{t}_0))) \tag{6.43}$$

$$\bar{\nu}_r \triangleq \sum_{\mathbf{f}\in\mathcal{F}} \max_{\mathbf{p}\in\mathcal{P}} \mathbf{1}(\theta - \angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p}-\mathbf{t}_0)) + \psi_r(\mathbf{f}, \mathcal{C}_r)). \tag{6.44}$$

The nested structure has better memory and computational efficiency than directly branching over 6D transformation space, since it maintains a queue for each 3D sub-problem, rather than one for the entire 6D problem. This requires significantly fewer simultaneously enqueued sub-cubes, reducing the runtime of priority queue operations. Moreover, with rotation search nested inside translation search, ψ_t only has to be calculated once per translation $\mathbf{t}$ not once per pose $(\mathbf{r}, \mathbf{t})$, and $\mathcal{F}$ can be rotated (by $\mathbf{R}^{-1}$) instead of $\mathcal{P}$ which typically has more elements. This makes it possible to precompute the rotated bearing vectors and rotation bounds for the top five levels of the rotation octree to reduce the amount of computation required in the inner BB subroutine. Finally, nesting does not weaken the optimality guarantee of this algorithm. In contrast, ϵ-suboptimality cannot be guaranteed when ϵ-suboptimal BB algorithms are nested.

6.5.2 Integrating Local Optimisation

Line 9 of Algorithm 6.1 shows how local optimisation methods are incorporated into the algorithm to refine the camera pose, in a similar manner to Brown et al. [2015] and Yang et al. [2016]. Whenever the BB algorithm finds a sub-cube pair $(\mathcal{C}_r, \mathcal{C}_t)$ with a greater lower bound $\underline{\nu}$ than half the best-so-far cardinality ν^*, the Perspective-n-Point (PnP) problem is solved, with correspondences given by the inlier pairs at the pose $(\mathbf{r}_0, \mathbf{t}_0)$. This solves the sub-problem of finding the camera pose given the correspondences. For this algorithm, a nonlinear optimisation solver [Kneip and Furgale, 2014] was selected, minimising the sum of angular distances between corresponding bearing vectors and points. The local optimisation method SoftPOSIT [David et al., 2004] is also applied at this stage to refine the camera pose without using correspondences. If a greater number of inliers ν is found by these refinement methods, ν^* is updated. In this way, BB and the refinement methods collaborate, with PnP finding the best pose given correspondences,

SoftPOSIT finding the nearest local maxima without correspondences and BB guiding the search for correspondences and jumping out of local maxima. PnP and SoftPOSIT accelerate convergence since the faster ν^* is increased, the sooner sub-cubes (with $\bar{\nu} \leqslant \nu^*$) can be culled (Alg. 6.1, Line 11).

6.5.3 Parallel Implementations

To improve the runtime characteristics of GOPAC, optional CPU multithreading was used. This variant of the algorithm divides the initial translation domain into sub-domains and runs GOPAC for each sub-domain in separate threads. It returns the greatest ν^* and the associated pose and correspondences. However, this approach has the disadvantage that sub-optimal sub-domains may be culled slower than a single-threaded implementation because their individual best ν values may be lower than the best ν value for the entire domain. Therefore a good parallel implementation should communicate the best ν value found so far between threads.

In view of this, a massively parallel version of the algorithm was implemented on the GPU with regular communication between the threads. It directly branches over 6D transformation space with each thread computing the bounds for a single branch. In this work, 16384 concurrent threads were used and an adaptive branching strategy was implemented that chooses to subdivide the rotation or translation dimensions based on which has the greater angular uncertainty, substantially reducing redundant branching and computation.

6.5.4 Further Implementation Details

Initialising the Number of Inliers

If the best-so-far number of inliers ν^* is initialised to a value close to the optimal value, sub-optimal branches are pruned sooner, reducing the overall runtime. However, the user is unlikely to know a tight lower bound on the optimal value of inliers. Therefore, in this work a P3P-RANSAC strategy and a guess-and-verify strategy are implemented. The former estimates a lower bound on the number of inliers using the RANSAC algorithm with randomly-sampled correspondences. The latter guesses a putative lower bound on the number of inliers and uses GOPAC to verify if an optimal solution is attainable from that initialisation. If not, it reduces its guess. This does not void the guarantee of optimality or distort the objective function, unlike an incorrectly guessed outlier fraction for a trimming strategy, and provides especial benefit when 2D outliers are rare. It proceeds as follows: set $\nu^* = n$; run GOPAC; stop if an optimality guarantee is found, otherwise update $n \leftarrow \max(n - s, 0)$ and repeat. The initial value of n is set to $N - 1$ and s is set to $\lceil 0.1N \rceil$.

Rotation Uncertainty Angle Bound

Lemma 6.2 (rotation uncertainty angle) requires the evaluation of the angle maximiser $\max_{\mathbf{r}\in\mathcal{S}_r} \angle(\mathbf{R}_\mathbf{r}\mathbf{p}, \mathbf{R}_{\mathbf{r}_0}\mathbf{p})$, where $\mathbf{p}$ is any 3D point, $\mathbf{r}_0$ is the angle-axis vector at the centre of rotation cube $\mathcal{C}_r$ with surface $\mathcal{S}_r$ and $\mathbf{R}_\mathbf{r}$ is the rotation matrix induced by angle-axis vector $\mathbf{r} \in \mathcal{C}_r$. While it is possible to calculate the bound by sampling the cube surface using a grid of step-size σ_g, evaluating the angle at each sample and adding $\sqrt{2}/2 \times \sigma_g$ to the greatest angle calculated (by Lemma 5.1), it is significantly more computationally efficient to use a different approach.

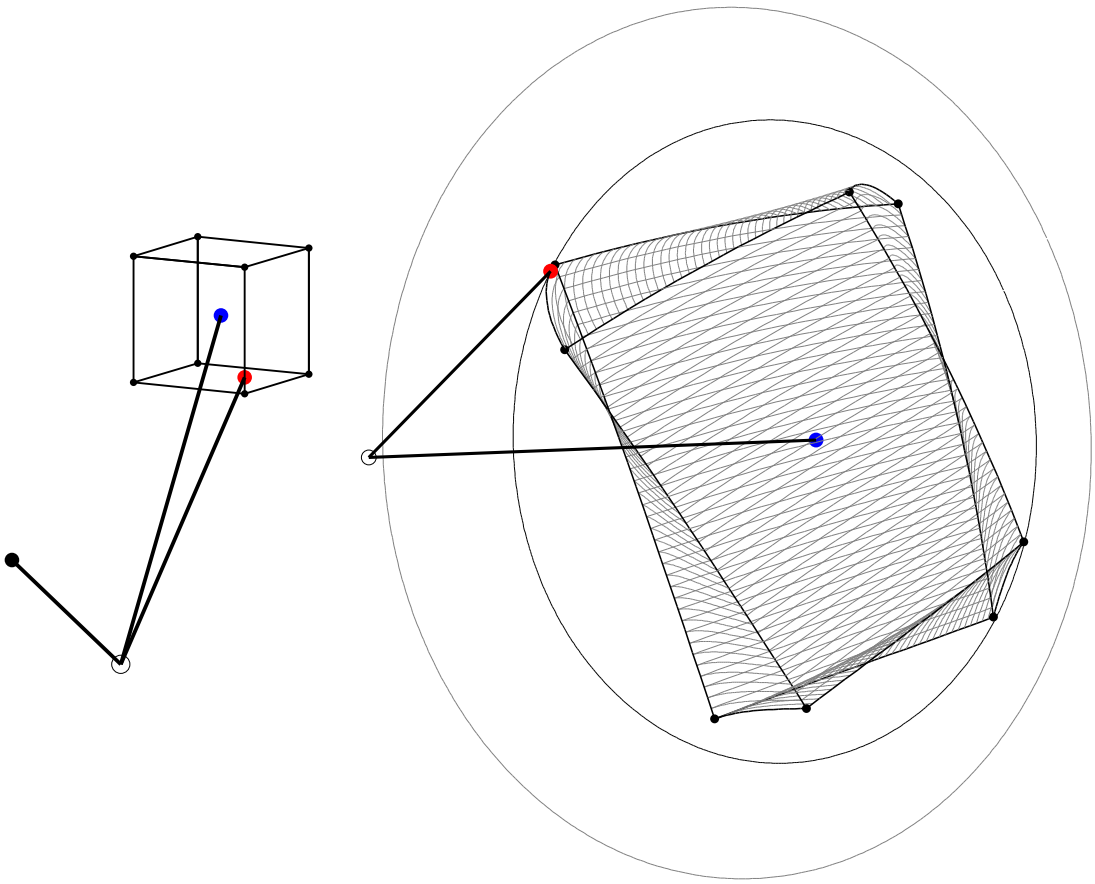
The alternative approach is contingent on two assumptions: (i) the maximum always occurs on the cube skeleton (edges and vertices), not the faces; and (ii) the angle function along each edge is quasiconvex or concave (specifically unimodal). Assumption (i) has been demonstrated empirically in simulations and can be seen in Figure 6.10, where it can be observed that rotation vectors on the cube faces are not projected beyond the convex hull of the projection of the edges for a given point. Therefore, the projected angle maximiser can always be found on an edge or vertex. Assumption (ii) has also been demonstrated empirically in simulations for all rotation cubes used in the GOPAC algorithm (that is, octree subdivisions of the angle-axis cube $[-\pi, \pi]^3$). In the vast majority of cases, the function is (quasi)convex. Consequently, the angle maximiser occurs at one of the two vertices joined by the edge (the extreme points). In a small fraction of cases, the maximum occurs on the edge, as in Figure 6.10. In these cases the assumption of unimodality enables the use of an efficient search routine, golden-section search, which does not require the time-consuming evaluation of the derivative. However, the sign of the derivative at the vertices needs to be evaluated to identify when the angle maximiser occurs on an edge. The derivative of the rotation angle function is obtained in Lemma 6.7.

Lemma 6.7. *(Derivative of the rotation angle function) Given a unit 3D bearing vector* $\mathbf{f}$ *and a rotation cube* $\mathcal{C}_r$ *centred at* $\mathbf{r}_0$ *with vertices* $\{\mathbf{r}_i\}_{i\in[1,8]}$*, then the derivative of the rotation angle function*

$$A_{ij}(\lambda) = \arccos\big((\mathbf{R}_{\mathbf{r}_0}^{-1}\mathbf{f}) \cdot (\mathbf{R}_{\mathbf{r}_{ij}(\lambda)}^{-1}\mathbf{f})\big) \tag{6.45}$$

with respect to λ*, for an edge parametrisation of* $\mathbf{r}_{ij}(\lambda) = \mathbf{r}_i + \lambda(\mathbf{r}_j - \mathbf{r}_i)$ *with* $\lambda \in [0, 1]$*, is given by*

$$\frac{dA_{ij}}{d\lambda} = \frac{-\mathbf{f}^\intercal \mathbf{R}_{\mathbf{r}_0} \mathbf{R}_{\mathbf{r}_{ij}(\lambda)}^\intercal [\mathbf{f}]_\times \Big(\mathbf{r}_{ij}(\lambda)\mathbf{r}_{ij}(\lambda)^\intercal - (\mathbf{R}_{\mathbf{r}_{ij}(\lambda)} - I)[\mathbf{r}_{ij}(\lambda)]_\times\Big)\Big(\mathbf{r}_j - \mathbf{r}_i\Big)}{\|\mathbf{r}_{ij}(\lambda)\|^2 \sqrt{1 - (\mathbf{f}^\intercal \mathbf{R}_{\mathbf{r}_0} \mathbf{R}_{\mathbf{r}_{ij}(\lambda)}^\intercal \mathbf{f})^2}}. \tag{6.46}$$



(a) Rotation cube in angle-axis space, with centre $\mathbf{r}_0$ (blue dot), projected angle maximiser $\mathbf{r}^*$ (red dot), origin (black circle) and unrotated 3D point $\mathbf{p}$ (black dot). Cube edges and vertices are shown as thin black lines and small black dots respectively.

(b) Rotation of 3D point $\mathbf{p}$ by angle-axis vectors on the surface of the cube, with centre-rotated point $\mathbf{R}_{\mathbf{r}_0}\mathbf{p}$ (blue dot), angle maximiser $\mathbf{R}_{\mathbf{r}^*}\mathbf{p}$ (red dot) and origin (black circle). 40 equally-spaced lines across each face are plotted in grey. All points and lines, other than the origin and lines to the origin, lie on the surface of a sphere with radius $\|\mathbf{p}\|$. Cube edges and vertices are shown as thin black lines and small black dots respectively. The weak rotation uncertainty angle ψ_r^w corresponds to the aperture angle of the cone formed by the origin and the large grey circle. The tighter rotation uncertainty angle ψ_r corresponds to the aperture angle of the cone formed by the origin and the black circle.

Figure 6.10: A rotation cube of angle-axis vectors and the surface induced by rotating a 3D point by all angle-axis vectors on the surface of that cube. Observe that the rotation vector that maximises the angle $\angle(\mathbf{R}_\mathbf{r}\mathbf{p}, \mathbf{R}_{\mathbf{r}_0}\mathbf{p})$ lies on an edge of the cube. Also observe that rotation vectors on the face of the cube (grey lines in the projection) do not rotate the point beyond the convex hull of the point rotated by the edges.

Proof. Equation (6.46) can be derived from the following differential.

$$\mathrm{d}A_{ij} = \frac{-1}{\sqrt{1-(\mathbf{f}^\intercal \mathbf{R}_{\mathbf{r}_0} \mathbf{R}^\intercal_{\mathbf{r}_{ij}(\lambda)} \mathbf{f})^2}} \mathrm{d}\big((\mathbf{R}^{-1}_{\mathbf{r}_0}\mathbf{f}) \cdot (\mathbf{R}^{-1}_{\mathbf{r}_{ij}(\lambda)}\mathbf{f})\big) \tag{6.47}$$

$$= \frac{-1}{\sqrt{1-(\mathbf{f}^\intercal \mathbf{R}_{\mathbf{r}_0} \mathbf{R}_{\mathbf{r}_{ij}(\lambda)}^\intercal \mathbf{f})^2}} \mathbf{f}^\intercal \mathbf{R}_{\mathbf{r}_0} \mathrm{d}(\mathbf{R}_{\mathbf{r}_{ij}(\lambda)}^\intercal \mathbf{f}) \tag{6.48}$$

$$= \frac{-1}{\sqrt{1-(\mathbf{f}^\intercal \mathbf{R}_{\mathbf{r}_0} \mathbf{R}_{\mathbf{r}_{ij}(\lambda)}^\intercal \mathbf{f})^2}} \mathbf{f}^\intercal \mathbf{R}_{\mathbf{r}_0} \mathrm{d}(\mathbf{R}_{-\mathbf{r}_{ij}(\lambda)} \mathbf{f}) \tag{6.49}$$

$$= \frac{\mathbf{f}^\intercal \mathbf{R}_{\mathbf{r}_0} \mathbf{R}_{-\mathbf{r}_{ij}(\lambda)} [\mathbf{f}]_\times}{\sqrt{1-(\mathbf{f}^\intercal \mathbf{R}_{\mathbf{r}_0} \mathbf{R}_{\mathbf{r}_{ij}(\lambda)}^\intercal \mathbf{f})^2}} \frac{(-\mathbf{r}_{ij}(\lambda))(-\mathbf{r}_{ij}(\lambda))^\intercal + (\mathbf{R}_{-\mathbf{r}_{ij}(\lambda)}^\intercal - I)[-\mathbf{r}_{ij}(\lambda)]_\times}{\| -\mathbf{r}_{ij}(\lambda)\|^2} \mathrm{d}(-\mathbf{r}_{ij}(\lambda)) \tag{6.50}$$

$$= \frac{-\mathbf{f}^\intercal \mathbf{R}_{\mathbf{r}_0} \mathbf{R}_{\mathbf{r}_{ij}(\lambda)}^\intercal [\mathbf{f}]_\times}{\sqrt{1-(\mathbf{f}^\intercal \mathbf{R}_{\mathbf{r}_0} \mathbf{R}_{\mathbf{r}_{ij}(\lambda)}^\intercal \mathbf{f})^2}} \frac{\mathbf{r}_{ij}(\lambda)\mathbf{r}_{ij}(\lambda)^\intercal - (\mathbf{R}_{\mathbf{r}_{ij}(\lambda)} - I)[\mathbf{r}_{ij}(\lambda)]_\times}{\|\mathbf{r}_{ij}(\lambda)\|^2} (\mathbf{r}_j - \mathbf{r}_i) \mathrm{d}\lambda \tag{6.51}$$

where (6.50) uses Result 1 from Gallego and Yezzi [2015]. □

Since the derivative is computationally expensive to calculate, only the sign of the derivative at the vertices is evaluated. Corollary 6.1 presents the simplified equations.

Corollary 6.1. *(Sign of the derivative of the rotation angle function at the vertices) Given a unit 3D bearing vector* $\mathbf{f}$ *and a rotation cube* $\mathcal{C}_r$ *centred at* $\mathbf{r}_0$ *with vertices* $\{\mathbf{r}_i\}_{i\in[1,8]}$*, then*

$$\operatorname{sgn}\left(\left. \frac{dA_{ij}}{d\lambda} \right|_{\lambda=0} \right) = \operatorname{sgn}\Big(-\mathbf{f}^\intercal \mathbf{R}_{\mathbf{r}_0} \mathbf{R}_{\mathbf{r}_i}^\intercal [\mathbf{f}]_\times \Big(\mathbf{r}_i \mathbf{r}_i^\intercal - (\mathbf{R}_{\mathbf{r}_i} - I)[\mathbf{r}_i]_\times \Big) \Big(\mathbf{r}_j - \mathbf{r}_i \Big) \Big) \tag{6.52}$$

and

$$\operatorname{sgn}\left(\left. \frac{dA_{ij}}{d\lambda} \right|_{\lambda=1} \right) = \operatorname{sgn}\Big(-\mathbf{f}^\intercal \mathbf{R}_{\mathbf{r}_0} \mathbf{R}_{\mathbf{r}_j}^\intercal [\mathbf{f}]_\times \Big(\mathbf{r}_j \mathbf{r}_j^\intercal - (\mathbf{R}_{\mathbf{r}_j} - I)[\mathbf{r}_j]_\times \Big) \Big(\mathbf{r}_j - \mathbf{r}_i \Big) \Big). \tag{6.53}$$

The method for calculating the rotation uncertainty angle $\psi_r(\mathbf{f}, \mathcal{C}_r)$ for a bearing vector $\mathbf{f}$ and a rotation cube $\mathcal{C}_r$, centred at angle-axis vector $\mathbf{r}_0$ with vertices $\{\mathbf{r}_i\}_{i\in[1,8]}$ and an edge parametrisation of $\mathbf{r}_{ij}(\lambda) = \mathbf{r}_i + \lambda(\mathbf{r}_j - \mathbf{r}_i)$, is as follows:

(i) for each edge, evaluate the sign of the derivative of the angle function $A_{ij}(\lambda) = \angle(\mathbf{R}_{\mathbf{r}_{ij}(\lambda)}^{-1}\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}^{-1}\mathbf{f})$ with respect to λ at $\lambda = 0$ and $\lambda = 1$ using (6.52) and (6.53);

(ii) if (6.52) is positive and (6.53) is negative, use golden-section search [Kiefer, 1953] with a tolerance of $\pi/2048$ to find the angle maximiser on that edge and add the tolerance $\pi/2048$ to the result;

(iii) otherwise, the angle maximiser on that edge is one of the vertices: evaluate the angle with respect to the projected cube centre at both vertices and choose the maximum; and

(iv) choose the maximum angle over all edges as ψ_r.

Note that golden-section search terminates at a tolerance of $\pi/2048$. By Lemma 5.1, the bound is therefore incorrect by at most $\pi/2048 = 0.088°$, a value that is added to the upper bound to ensure optimality.

Tighter Upper Bound

The upper bound given in Theorem 6.3 requires the evaluation of $\Gamma(\mathbf{f}, \mathbf{p})$ for a given translation cuboid $\mathcal{C}_t$. Γ may be evaluated by observing that the minimum angle between a ray $\mathbf{f}$ and a cuboid $\mathbf{R}_{\mathbf{r}_0}(\mathbf{p}-\mathbf{t})$ for $\mathbf{t} \in \mathcal{C}_t$ is (a) zero if the ray passes through the cuboid or (b) the angle between the ray and the point on the skeleton of the cuboid (vertices and edges) with least angular displacement from $\mathbf{f}$. Thus, for the translation domain $\mathcal{C}_t$ with skeleton $\mathcal{S}k_t$,

$$\Gamma(\mathbf{f}, \mathbf{p}) = \begin{cases} \max\limits_{\mathbf{t} \in \mathcal{S}k_t} \mathbf{1}(\theta - \angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p}-\mathbf{t})) + \psi_r(\mathbf{f}, \mathcal{C}_r)) & \text{if } \angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p}-\mathbf{t}_0)) > \psi_t(\mathbf{p}, \mathcal{C}_t) \\ 1 & \text{else.} \end{cases} \tag{6.54}$$

The key here is finding $\angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p}-\mathbf{t}))$ which maximises Γ over the skeleton. For the first case in (6.54), this can be done by finding $\mathbf{p}-\mathbf{t}$ with least angular displacement from $\mathbf{R}_{\mathbf{r}_0}^{-1}\mathbf{f}$. The following technique is applied:

(i) find the octant of $\mathbf{p}-\mathbf{t}_0$ with respect to the coordinate axes and project the cube to the unit sphere as a spherical hexagon;
(ii) determine in which lune induced by the spherical hexagon $\mathbf{R}_{\mathbf{r}_0}^{-1}\mathbf{f}$ resides; and
(iii) solve for the point on the hexagon edge in that lune with least angular displacement from $\mathbf{R}_{\mathbf{r}_0}^{-1}\mathbf{f}$.

As a result of the design of the data structure, it is known that the cuboid of translated points $\mathbf{p}-\mathbf{t}$ for $\mathbf{t} \in \mathcal{C}_t$ lies entirely in one octant of $\mathbb{R}^3$. By finding the octant (i), the cuboid can be projected to a spherical hexagon on the unit sphere, as shown in Figure 6.11. That is, the 6 vertices and edges of the cuboid that project to the spherical hexagon can be determined. This simplifies the problem to finding the closest point $\hat{\mathbf{v}}^*$ on the hexagon to the rotated bearing vector. Finding in which lune the rotated bearing vector lies (ii) further simplifies the problem to one of finding the closest point on a geodesic to the rotated bearing vector. This can be solved in closed form (iii):

$$\mathbf{v}^* = \begin{cases} \mathbf{v} \text{ if } \lambda \leqslant 0 \\ \mathbf{v} + 2\delta_{ti}\lambda\mathbf{e}_i \text{ if } \lambda \in (0,1) \\ \mathbf{v} + 2\delta_{ti}\mathbf{e}_i \text{ if } \lambda \geqslant 1 \end{cases} \tag{6.55}$$

$$\lambda = \frac{(\mathbf{R}_{\mathbf{r}_0}^{-1}\mathbf{f}) \cdot \mathbf{e}_i - ((\mathbf{R}_{\mathbf{r}_0}^{-1}\mathbf{f}) \cdot \hat{\mathbf{v}})(\hat{\mathbf{v}} \cdot \mathbf{e}_i)}{(\mathbf{R}_{\mathbf{r}_0}^{-1}\mathbf{f}) \cdot \hat{\mathbf{v}} - ((\mathbf{R}_{\mathbf{r}_0}^{-1}\mathbf{f}) \cdot \mathbf{e}_i)(\hat{\mathbf{v}} \cdot \mathbf{e}_i)} \frac{\|\mathbf{v}\|}{2\delta_{ti}} \tag{6.56}$$

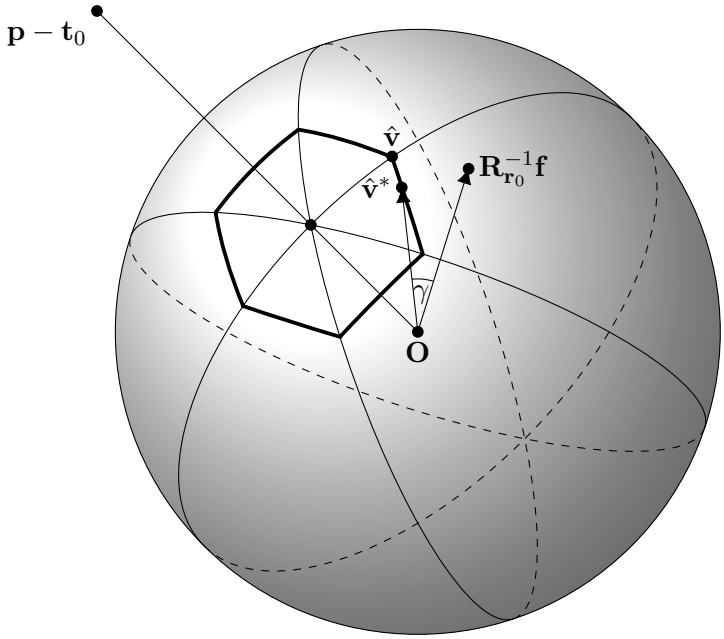


Figure 6.11: Unit sphere onto which the vertices and edges of the translation cuboid $\mathbf{p} - \mathbf{t}$ for $\mathbf{t} \in \mathcal{C}_t$ have been projected. The resulting spherical hexagon, comprising 6 vertices and 6 edges of the projected cuboid, simplifies the angle calculation by reducing it to finding in which spherical lune (surface of the spherical wedge) the rotated bearing vector $\mathbf{R}_{\mathbf{r}_0}^{-1}\mathbf{f}$ resides and then solving for the closest point on the geodesic of the hexagon edge in that lune. This angle $\gamma = \angle(\mathbf{R}_{\mathbf{r}_0}^{-1}\mathbf{f}, \hat{\mathbf{v}}^*)$ is the smallest angle between $\mathbf{R}_{\mathbf{r}_0}^{-1}\mathbf{f}$ and any point in the translation cuboid.

where $\mathbf{v}$, $\mathbf{e}_i$ and δ_{ti} are the cuboid vertex $\mathbf{v} = \mathbf{p} - \mathbf{t}_v$ that projects to the hexagon vertex $\hat{\mathbf{v}} = \mathbf{v}/\|\mathbf{v}\|$, the i^{th} standard basis vector in the direction of the next cuboid vertex in the hexagon cycle and the half side-length of the cuboid in that direction respectively.

Precomputing Angles on the Sphere

To reduce the time complexity of the bound calculations, the angle between the translated 3D points $\mathbf{p} - \mathbf{t}_0$ and any location on the unit sphere may be precomputed. Thus, for a fixed translation, the angle between *any* rotated bearing vector and its rotationally-closest 3D point may be precomputed. This is the analogue in S^2 of a distance transform in $\mathbb{R}^n$, in that the surface of the sphere is discretised and a look-up table constructed. It exploits the nested structure of the algorithm, since many bounds for different rotations are calculated for a single translation. By using this precomputation, the max operations in (6.39)–(6.44) are reduced from $\mathcal{O}(M)$ to $\mathcal{O}(1)$.

The procedure for constructing the look-up table is as follows. The sphere is subdivided into 98304 regions by projecting it onto an enclosing cube whose faces are

partitioned with quad-trees. A linear projection onto the cube is used to facilitate the rapid conversion from a unit vector to a location in the data structure. The disadvantage of a linear projection is that the cell sizes are not uniform. However, the configuration chosen ensures that the maximum angle between an arbitrary point and its nearest cell centre is 0.6°. Hence, the effect of using this data structure is to relax θ by up to 0.6°. This means that the result is no longer optimal with respect to θ. Nonetheless, for most practical uses, the optimal result is still obtained and it may be useful for solving problems for which the number of 3D points M is large. For the purposes of evaluating the algorithm, this feature was not used in the experiments.

6.5.5 Convergence of the Upper and Lower Bounds

A requirement of branch-and-bound is that the upper and lower bounds converge as the size of the branch tends to zero. The convergence of the bounds can be proved as follows. It is clear that the upper bound (6.18) is equal to the lower bound (6.16) when the uncertainty angle bounds $\psi_r(\mathbf{f}, \mathcal{C}_r)$ and $\psi_t(\mathbf{p}, \mathcal{C}_t)$ are zero. Similarly, the tighter upper bound (6.23) is equal to the lower bound when the rotation uncertainty angle $\psi_r(\mathbf{f}, \mathcal{C}_r)$ is zero and the translation sub-cuboid $\mathcal{C}_t$ is of size zero, since then $\angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p}-\mathbf{t})) = \angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p}-\mathbf{t}_0))$ for $\mathbf{t} \in \mathcal{C}_t$. It remains to be seen that $\psi_r(\mathbf{f}, \mathcal{C}_r)$ and $\psi_t(\mathbf{p}, \mathcal{C}_t)$ tend to zero as the size of the sub-cuboids $\mathcal{C}_r$ and $\mathcal{C}_t$ tend to zero, irrespective of the values of $\mathbf{f}$ and $\mathbf{p}$.

The rotation uncertainty angle bound $\psi_r(\mathbf{f}, \mathcal{C}_r)$ involves a maximisation over all rotations on the surface of the sub-cube $\mathcal{C}_r$. As the sub-cube size tends to zero, in the limit the surface and centre of the cube become identified and therefore the angle $\angle(\mathbf{R}_\mathbf{r}\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}\mathbf{f})$ equals zero. The translation uncertainty angle bound $\psi_t(\mathbf{p}, \mathcal{C}_t)$ involves a maximisation over all translations on the vertices of the sub-cuboid $\mathcal{C}_t$. As the sub-cuboid size tends to zero, in the limit the vertices and centre of the cuboid become identified and therefore the angle $\angle(\mathbf{p}-\mathbf{t}, \mathbf{p}-\mathbf{t}_0)$ equals zero. The point $\mathbf{p}$ cannot lie inside the sub-cuboid for a sufficiently small cuboid, since the translation domain has been restricted to exclude translations for which $\|\mathbf{p}-\mathbf{t}\| < \zeta$. Therefore the upper and lower bounds converge as the size of sub-cuboids (branches) tend to zero.

However, an advantage of the inlier maximisation formulation is that the gap between the bounds becomes exactly zero substantially before the branch size becomes infinitesimal. There are nonetheless critical configurations of points and bearing vectors for which the bounds will only converge in the limit. The simplest case is illustrated in Figure 6.12. In this rotation-only example, the angle between the 3D point vectors is infinitesimally less than $\pi - 2\theta$. To prove that the maximum number of inliers is 1, infinitesimally small rotation sub-cubes would be required.

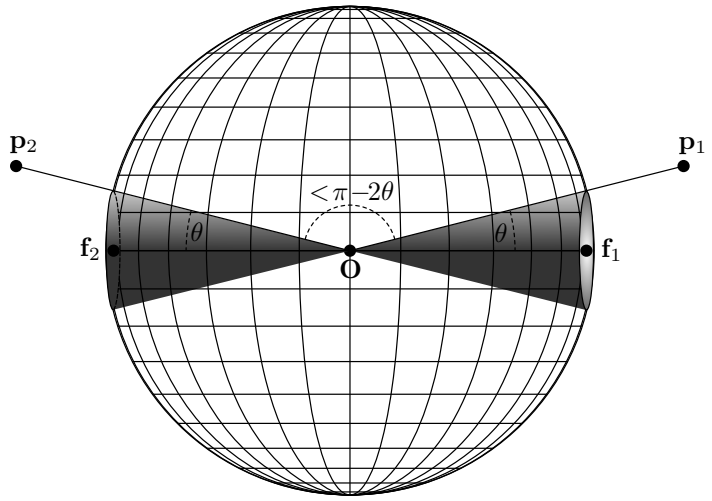


Figure 6.12: Example of a critical configuration (rotation only). The angle between the 3D point vectors $\angle(\mathbf{p}_1, \mathbf{p}_2)$ is infinitesimally less than $\pi - 2\theta$. To prove that the maximum number of inliers is 1, infinitesimally small rotation sub-cubes would be required.

In order to guarantee that the algorithm terminates in finite time, a small tolerance value η must be subtracted from the uncertainty angles. That is, the uncertainty angles in all the formulae must be replaced with their primed versions: $\psi'_r = \psi_r - \eta$ and $\psi'_t = \psi_t - \eta$. For the tighter upper bound, η also has to be added to $\angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p} - \mathbf{t}))$. The upper bound $\bar{\nu} \triangleq \bar{\nu}_t$ (6.18) for the translation cuboid $\mathcal{C}_t$, rewritten with the tolerances, is found by running rotation BB until convergence with the following bounds

$$\underline{\nu}_r \triangleq \sum_{\mathbf{f} \in \mathcal{F}} \max_{\mathbf{p} \in \mathcal{P}} \mathbf{1}(\theta - \angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p} - \mathbf{t}_0)) + \psi'_t(\mathbf{p}, \mathcal{C}_t)) \tag{6.57}$$

$$\bar{\nu}_r \triangleq \sum_{\mathbf{f} \in \mathcal{F}} \max_{\mathbf{p} \in \mathcal{P}} \mathbf{1}(\theta - \angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p} - \mathbf{t}_0)) + \psi'_t(\mathbf{p}, \mathcal{C}_t) + \psi'_r(\mathbf{f}, \mathcal{C}_r)). \tag{6.58}$$

The tighter upper bound (6.23) instead uses

$$\underline{\nu}_r \triangleq \sum_{\mathbf{f} \in \mathcal{F}} \max_{\mathbf{p} \in \mathcal{P}} \max_{\mathbf{t} \in \mathcal{C}_t} \mathbf{1}(\theta - \angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p} - \mathbf{t})) - \eta) \tag{6.59}$$

$$\bar{\nu}_r \triangleq \sum_{\mathbf{f} \in \mathcal{F}} \max_{\mathbf{p} \in \mathcal{P}} \max_{\mathbf{t} \in \mathcal{C}_t} \mathbf{1}(\theta - \angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p} - \mathbf{t})) - \eta + \psi'_r(\mathbf{f}, \mathcal{C}_r)). \tag{6.60}$$

The lower bound $\underline{\nu} \triangleq \underline{\nu}_t$ (6.16) for $\mathcal{C}_t$ is found by running rotation BB until convergence using bounds (6.57) and (6.58) with ψ'_t set to zero. That is, the bounds

$$\underline{\nu}_r \triangleq \sum_{\mathbf{f} \in \mathcal{F}} \max_{\mathbf{p} \in \mathcal{P}} \mathbf{1}(\theta - \angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p} - \mathbf{t}_0))) \tag{6.61}$$

$$\bar{\nu}_r \triangleq \sum_{\mathbf{f} \in \mathcal{F}} \max_{\mathbf{p} \in \mathcal{P}} \mathbf{1}(\theta - \angle(\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}(\mathbf{p} - \mathbf{t}_0)) + \psi'_r(\mathbf{f}, \mathcal{C}_r)). \tag{6.62}$$

In this work, η was set to machine epsilon for maximal precision. In C++, this can be accessed by using the command `std::numeric_limits<float>::epsilon()`.

6.5.6 Time Complexity

Explicitly including the tolerance η in the bound formulae makes it possible to derive a bound on the worst-case search tree depth and thereby obtain the time complexity of the algorithm. In terms of the size of the input, the GOPAC algorithm is $\mathcal{O}(MN)$, or $\mathcal{O}(N)$ if angle precomputation is used, where M is the number of 3D points and N is the number of bearing vectors. However, the notation conceals a very large constant. Including the constant factors that can be selected by the user yields $\mathcal{O}(\rho_{t_0}^3 \zeta^{-3} \eta^{-6} MN)$, where ρ_{t_0} is the half space diagonal of the initial translation cuboids, that is one-quarter the space diagonal of the translation domain, and ζ and η are small previously-defined constants set by the user.

Calculating the upper and lower bounds involves a summation over $\mathcal{F}$ and a maximisation over $\mathcal{P}$, therefore the complexity is $\mathcal{O}(MN)$. If angle precomputation is used, the maximisation becomes a constant-time lookup leading to a bound complexity of $\mathcal{O}(N)$. However, it is as of yet unclear how the number of iterations (explored sub-cuboids) depends on the inputs. The central finding is that branch-and-bound is exponential in the worst-case tree search depth D, but D is logarithmic in η^{-1}. Therefore the complexity of BB is polynomial in η^{-1}, where η is the angle tolerance. Rotation and translation search will be treated separately before being combined into an analysis of nested rotation and translation search.

Theorem 6.4. *(Rotation Search Depth and Complexity) Let* $\rho_{r_0} = \sqrt{3}\delta_{r_0} = \sqrt{3}\pi/2$ *be the half space diagonal of the initial rotation sub-cube* $\mathcal{C}_{r_0}$*. Then*

$$D_r = \max\left\{\left\lceil \log_2 \frac{\rho_{r_0}}{\eta} \right\rceil, 0\right\} \tag{6.63}$$

is an upper bound on the worst-case rotation tree search depth for an uncertainty angle tolerance η *and* $\mathcal{O}(\eta^{-3})$ *is the time complexity of rotation BB search.*

Proof. Rotation BB converges when $\underline{\nu}_r \geqslant \bar{\nu}_r$. For any $\psi_t'(\mathbf{p}, \mathcal{C}_t)$ in (6.57) and (6.58), $\underline{\nu}_r \geqslant \bar{\nu}_r$ when $\psi_r'(\mathbf{f}, \mathcal{C}_r) \leqslant 0$ or equivalently $\psi_r(\mathbf{f}, \mathcal{C}_r) \leqslant \eta$ for all $\mathbf{f} \in \mathcal{F}$. Now,

$$\begin{aligned}
\psi_r(\mathbf{f}, \mathcal{C}_r) &= \min\left\{\max_{\mathbf{r} \in \mathcal{S}_r} \angle(\mathbf{R}_\mathbf{r}\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}\mathbf{f}), \pi\right\} && (6.64)\\
&= \min\{\angle(\mathbf{R}_{\mathbf{r}^*}\mathbf{f}, \mathbf{R}_{\mathbf{r}_0}\mathbf{f}), \pi\} && (6.65)\\
&\leqslant \min\left\{\sqrt{3}\delta_r, \pi\right\} && (6.66)\\
&\leqslant \rho_r && (6.67)
\end{aligned}$$

where (6.65) replaces the maximisation with the arg max rotation $\mathbf{r}^*$, (6.66) follows from Lemma 6.1 (6.6) and ρ_r is the half space diagonal of the rotation sub-cube $\mathcal{C}_r$. At rotation search tree depth D_r, the half space diagonal is given by

$$\rho_{r_{D_r}} = \frac{1}{2}\rho_{r_{D_r-1}} = \frac{1}{2^{D_r}}\rho_{r_0}. \tag{6.68}$$

Substituting into (6.67) gives

$$\psi_r(\mathbf{f}, \mathcal{C}_{r_{D_r}}) \leqslant \rho_{r_{D_r}} = 2^{-D_r}\rho_{r_0}. \tag{6.69}$$

To find the worst-case rotation search tree depth, the constraint $\psi_r(\mathbf{f}, \mathcal{C}_r) \leqslant \eta$ is applied:

$$\psi_r(\mathbf{f}, \mathcal{C}_{r_{D_r}}) \leqslant 2^{-D_r}\rho_{r_0} \leqslant \eta. \tag{6.70}$$

Taking the logarithm of both sides yields

$$D_r \geqslant \log_2 \frac{\rho_{r_0}}{\eta}. \tag{6.71}$$

Equation (6.63) follows from the requirement that D_r be a non-negative integer. Now, rotation BB will have examined at most

$$N_r = 8(1 + 8 + 8^2 + \cdots + 8^{D_r}) = 8\frac{8^{D_r+1} - 1}{8 - 1} = \frac{8}{7}\Big((2^{D_r+1})^3 - 1\Big) \tag{6.72}$$

sub-cubes at search depth D_r, due to the octree structure. Finally, substituting (6.63) into (6.72) and simplifying using Bachmann–Landau notation gives

$$N_r = O\left(\left(\frac{\rho_{r_0}}{\eta}\right)^3\right) = O\Big(\eta^{-3}\Big). \tag{6.73}$$

The ρ_{r_0} term is removed because it is a fixed constant not set by the user. □

The analysis of the worst-case search depth and time complexity for translation search proceeds in a similar manner.

Theorem 6.5. *(Translation Search Depth and Complexity) Let ρ_{t_0} be the half space diagonal of the initial translation sub-cuboid $\mathcal{C}_{t_0}$. Then*

$$D_t = \max\left\{\left\lceil \log_2 \frac{\rho_{t_0}}{\zeta \sin \eta} \right\rceil, 0\right\} \tag{6.74}$$

is an upper bound on the worst-case translation tree search depth for an uncertainty angle tolerance η and $\mathcal{O}(\rho_{t_0}^3 \zeta^{-3} \eta^{-3})$ is the time complexity of translation BB search.

Proof. Translation BB converges when $\underline{\nu}_t \geqslant \bar{\nu}_t$. This condition is met when $\psi'_t(\mathbf{p}, \mathcal{C}_t) \leqslant 0$ or equivalently $\psi_t(\mathbf{p}, \mathcal{C}_t) \leqslant \eta$ for all $\mathbf{p} \in \mathcal{P}$. This can be seen by inspecting (6.57) and (6.61) and noting that at convergence the upper and lower rotation bounds will be equal. Now for $\|\mathbf{p} - \mathbf{t}_0\| \geqslant \rho_t$, which is guaranteed for $\rho_t \leqslant \zeta$,

$$\psi_t(\mathbf{p}, \mathcal{C}_t) = \max_{\mathbf{t} \in \mathcal{V}_t} \angle(\mathbf{p} - \mathbf{t}, \mathbf{p} - \mathbf{t}_0) \tag{6.75}$$

$$\leqslant \max_{\mathbf{t} \in S_t^2} \angle(\mathbf{p} - \mathbf{t}, \mathbf{p} - \mathbf{t}_0) \tag{6.76}$$

$$= \arcsin\left(\frac{\rho_t}{\|\mathbf{p} - \mathbf{t}_0\|}\right) \tag{6.77}$$

$$\leqslant \arcsin\left(\frac{\rho_t}{\zeta}\right) \tag{6.78}$$

where (6.76) follows from maximising the angle over the circumsphere S_t^2 of the cuboid instead of the vertices, (6.77) is shown in Brown et al. [2015], and (6.78) follows from the restriction $\|\mathbf{p} - \mathbf{t}\| \geqslant \zeta$. At depth D_t, the half space diagonal of $\mathcal{C}_{t_{D_t}}$ is

$$\rho_{t_{D_t}} = 2^{-1}\rho_{t_{D_t-1}} = 2^{-D_t}\rho_{t_0}. \tag{6.79}$$

Substituting into (6.78) gives

$$\psi_t(\mathbf{p}, \mathcal{C}_{t_{D_t}}) \leqslant \arcsin\left(\frac{\rho_{t_{D_t}}}{\zeta}\right) = \arcsin\left(\frac{\rho_{t_0}}{\zeta 2^{D_t}}\right). \tag{6.80}$$

To find the worst-case translation search tree depth, the constraint $\psi_t \leqslant \eta$ is applied

$$\psi_t(\mathbf{p}, \mathcal{C}_{t_{D_t}}) \leqslant \arcsin\left(\frac{\rho_{t_0}}{\zeta 2^{D_t}}\right) \leqslant \eta. \tag{6.81}$$

Taking the sine and logarithm of both sides yields

$$D_t \geqslant \log_2 \frac{\rho_{t_0}}{\zeta \sin\eta}. \tag{6.82}$$

Equation (6.74) follows from the requirement that D_t be a non-negative integer. Now, translation BB will have examined at most

$$N_t = 8(1 + 8 + 8^2 + \cdots + 8^{D_t}) = 8\frac{8^{D_t+1} - 1}{8 - 1} = \frac{8}{7}\Big((2^{D_t+1})^3 - 1\Big) \tag{6.83}$$

sub-cuboids at search depth D_t. Finally, substituting (6.74) into (6.83) gives

$$N_t = O\Big(\rho_{t_0}^3 \zeta^{-3} (\sin\eta)^{-3}\Big) = O\Big(\rho_{t_0}^3 \zeta^{-3} \eta^{-3}\Big) \tag{6.84}$$

using Bachmann–Landau simplification and the Taylor expansion of $\sin\eta$. □

In the nested BB search structure detailed at the beginning of Section 6.5, for every translation sub-cuboid examined, rotation BB search is run once to find the lower translation bound and again to find the upper translation bound. Thus the number of rotation sub-cubes examined is at worst equal to $2N_tN_r$. For each rotation sub-cube, the upper and lower bounds are calculated with a time complexity of $\mathcal{O}(MN)$. Thus the number of bound calculations is at worst equal to $4N_tN_r$. Combining the time complexity analyses (6.73) and (6.84) with the time complexity of the bound calculations leads to the following corollary.

Corollary 6.2. *(Time Complexity of GOPAC) Let ρ_{t_0} be the half space diagonal of the initial translation sub-cuboid $\mathcal{C}_{t_0}$, ζ be the translation restriction parameter, η be the uncertainty angle tolerance, M be the number of 3D points and N be the number of bearing vectors, then the time complexity of the GOPAC algorithm is given by*

$$\mathcal{O}\Big(\rho_{t_0}^3 \zeta^{-3} \eta^{-6} MN\Big). \tag{6.85}$$

It is important to observe that experimental evaluation of runtime is more revealing for BB algorithms than time complexity analysis. The main reason to use BB is that it can prune large regions of the search space, reducing the size of the problem. This is not reflected in the complexity analysis.

6.6 Results

The GOPAC algorithm, denoted GP, was evaluated with respect to the baseline algorithms RANSAC [Fischler and Bolles, 1981], SoftPOSIT [David et al., 2004] and BlindPnP [Moreno-Noguer et al., 2008], denoted RS, SP and BP respectively, using both synthetic and real data. The RANSAC approach uses the OpenGV framework [Kneip and Furgale, 2014] and the P3P algorithm [Kneip et al., 2011] with randomly-sampled correspondences. SoftPOSIT and BlindPnP are local optimisation algorithms and hence require a pose prior. Therefore, a torus or cube prior was used in the synthetic experiments to allow a fair comparison. In general, the space of camera poses is much larger than the restrictive torus prior and a good prior can rarely be known in advance. The registration algorithm in Brown et al. [2015] was not evaluated because the code and 2D–3D feature sets were not released publicly and the method is not easily reimplementable. However, it was shown theoretically in Section 6.4.2 that the bounds in this work are tighter and this will be shown experimentally in Section 6.6.1.

Except where otherwise specified, the inlier threshold θ was set to 1°, the lower bound from Theorem 6.1 and the upper bound from Theorem 6.2 were used, SoftPOSIT and nonlinear PnP refinement were applied and the point-to-camera limit ζ was set to

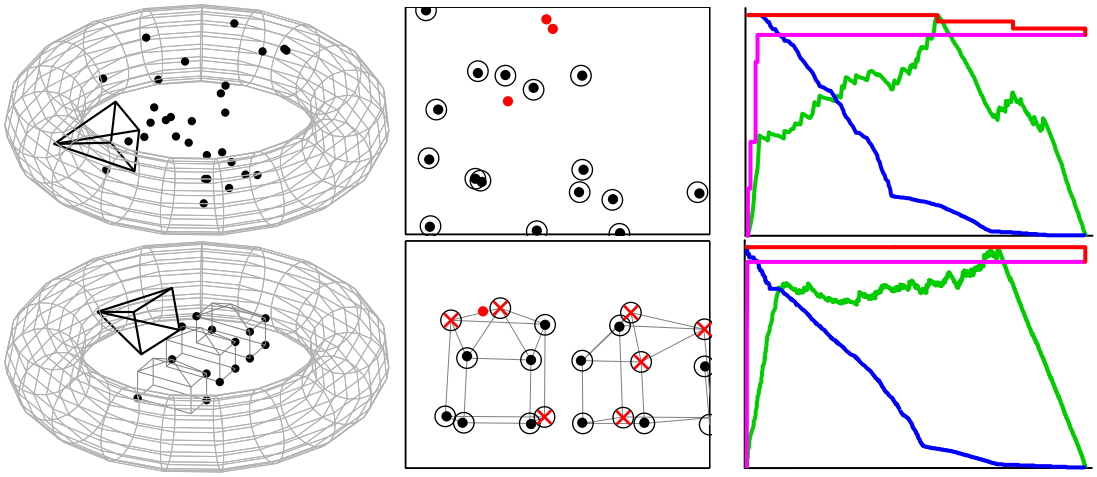
0.1. All experiments were run on a PC with a 3.4GHz quad core CPU, 8 threads were used for CPU multithreading, and up to 4 GeForce GTX 1080 Ti GPUs were used for GPU multithreading. The GOPAC code was written in unoptimised C++ and uses the Eigen library [Guennebaud et al., 2010] for matrix calculations, the OpenGV library [Kneip and Furgale, 2014] for RANSAC and PnP refinement and the Armadillo library [Sanderson and Curtin, 2016] for SoftPOSIT refinement.

6.6.1 Synthetic Data Experiments

To evaluate GOPAC in a setting where the true camera pose was known, 50 independent Monte Carlo simulations were performed per parameter setting, using the framework of Moreno-Noguer et al. [2008]: M random 3D points were generated from $[-1, 1]^3$; a fraction ω_{3D} of the 3D points were randomly selected as outliers to model occlusion; the inliers were projected to a 640×480 virtual image with an effective focal length of 800; normal noise was added to the 2D points with a standard deviation σ of 2 pixels; and random points were added to the image such that a fraction ω_{2D} of the 2D points were outliers. In addition to these random point experiments, the same procedure was applied to a repetitive CAD structure with $M = 27$ 3D points. Examples of both datasets and 2D alignment results are shown in Figure 6.13(a)–(b).

The evolution over time of the global lower and upper bounds is shown in Figure 6.13(c) for both examples. Branch-and-Bound (BB) and the refinement methods (PnP and SoftPOSIT) collaborate to increase the lower bound with BB guiding the search into convergence basins with increasingly higher local maxima and the refinement methods jumping to the nearest local maximum (the staircase pattern). It can be observed that the majority of the runtime is spent decreasing the upper bound, indicating that it will often find the global optimum when terminated early, albeit without an optimality guarantee.

To facilitate fair comparison with SoftPOSIT and BlindPnP, pose priors were used for these experiments. The torus prior constrains the camera centre to a torus around the 3D point-set with the optical axis directed towards the model, replicating the experimental design of Moreno-Noguer et al. [2008]. For BlindPnP, the poses were represented by a 20 component Gaussian mixture model generated from the torus. For SoftPOSIT, the 20 mean poses from the mixture model were used to initialise the algorithm. For GOPAC, the torus was approximated by a translation domain formed from a set of 12 translation cubes with side-length 1. However, GOPAC was given no rotation prior, giving the local methods a very significant advantage. The cube prior constrains the camera centre to a cube centred randomly in $[-1, 1]^3$ with side-length 0.5 and has no restriction on rotation. This is a more realistic prior than the torus



(a) 3D Results (b) 2D Results (c) Bound Evolution

Figure 6.13: Sample 2D and 3D results for two experiments using the random points and CAD structure datasets. (a) 3D models, true and GOPAC-estimated camera fulcra (completely overlapping) and toroidal pose priors. Only non-occluded 3D points are shown. (b) 2D alignment results. True projections of non-occluded 3D points are shown as black dots, 2D outliers as red dots, GOPAC projections as black circles and GOPAC-classified 3D outliers as red crosses. (c) Evolution over time of the upper (red) and lower (magenta) bounds, remaining unexplored translation volume (blue) and translation queue size (green) as a fraction of their maximum values.

since it assumes much less about the camera pose and does not impose restrictive constraints on camera rotation and translation. The intent of this prior is to simulate the task of camera pose estimation with respect to a scene, such as locating a camera inside a building, in contrast to the torus prior which simulates the task of camera pose estimation with respect to an object, such as a teapot on a table, for a known camera height. For BlindPnP, the poses were represented by a 50 component Gaussian mixture model generated from a set of 200 random camera centres in the cube and 200 uniform random rotation matrices [Mezzadri, 2007]. While Moreno-Noguer et al. [2008] recommend 20 components, the increased number was necessary to model the increased rotation uncertainty. For SoftPOSIT, the 50 mean poses from the mixture model were used to initialise the algorithm. For GOPAC, the prior was passed directly to the algorithm as the translation domain.

The results are shown in Tables 6.1 and 6.2 and Figures 6.14 and 6.15. Two success rates are reported: the fraction of trials where the true maximum number of inliers was found and the fraction where the correct pose was found, where the angle between the output rotation and the ground truth rotation is less than 0.1 radians and the camera centre error $\|\mathbf{t} - \mathbf{t}_{\mathrm{GT}}\|/\|\mathbf{t}_{\mathrm{GT}}\|$ relative to the ground truth $\mathbf{t}_{\mathrm{GT}}$ is less than 0.1, as used in Moreno-Noguer et al. [2008]. The 2D and 3D outlier fractions

Table 6.1: Camera pose results for the random points ($M = 80$) dataset with the torus prior and 50% 3D outliers ($\omega_{3D} = 0.5$). Quartiles ($Q_2{}^{Q_3}_{Q_1}$) for translation error ($\times 10^2$), rotation error and runtime and the mean inlier recall and success rates are reported.

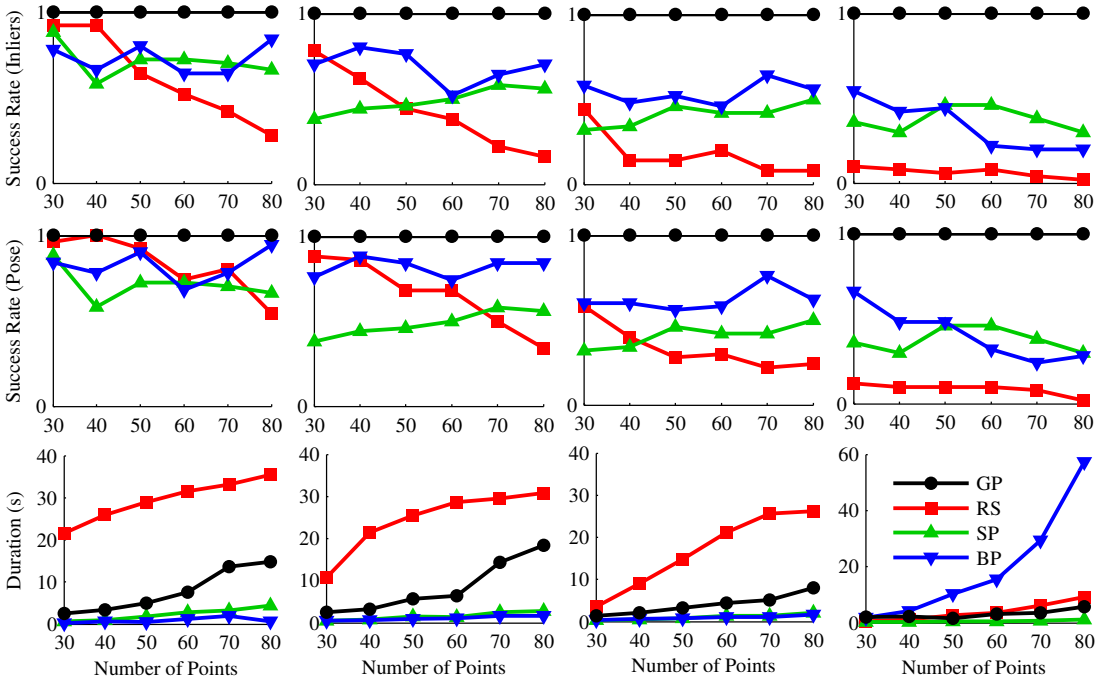
Method	GOPAC	RANSAC	SoftPOSIT	BlindPnP
Translation Error	$\mathbf{0.89}^{1.95}_{0.41}$	$52.5^{112}_{17.2}$	$6.81^{42.6}_{0.23}$	$1.02^{26.7}_{0.29}$
Rotation Error (°)	$\mathbf{0.49}^{0.66}_{0.33}$	$113^{134}_{9.13}$	$8.48^{101}_{0.14}$	$0.62^{28.3}_{0.19}$
Recall (Inliers)	**1.00**	0.43	0.66	0.67
Success Rate (Inliers)	**1.00**	0.08	0.50	0.56
Success Rate (Pose)	**1.00**	0.24	0.50	0.62
Runtime (s)	$8.02^{10.4}_{3.85}$	$26.3^{26.5}_{26.2}$	$2.05^{2.12}_{1.99}$	$1.68^{1.88}_{0.78}$

Table 6.2: Camera pose results for the CAD structure ($M = 27$) dataset with the torus prior and 50% 3D outliers ($\omega_{3D} = 0.5$). Quartiles ($Q_2{}^{Q_3}_{Q_1}$) for translation error ($\times 10^2$), rotation error and runtime and the mean inlier recall and success rates are reported.

Method	GOPAC	RANSAC	SoftPOSIT	BlindPnP
Translation Error	$\mathbf{0.80}^{1.51}_{0.31}$	$4.20^{50.9}_{1.55}$	$39.3^{70.4}_{1.30}$	$1.71^{29.1}_{0.48}$
Rotation Error (°)	$\mathbf{0.54}^{0.94}_{0.28}$	$1.80^{168}_{0.89}$	$13.8^{133}_{0.45}$	$0.94^{82.8}_{0.37}$
Recall (Inliers)	**1.00**	0.81	0.69	0.74
Success Rate (Inliers)	**1.00**	0.36	0.30	0.52
Success Rate (Pose)	**1.00**	0.56	0.30	0.60
Runtime (s)	$1.22^{2.16}_{1.00}$	$2.42^{2.62}_{2.25}$	$0.72^{0.79}_{0.65}$	$0.71^{0.92}_{0.37}$

were fixed to 0 when not being varied and CPU multithreading was used when 2D outliers were present ($\omega_{2D} > 0$). GOPAC outperformed the other methods, reliably finding the global optimum while still being relatively efficient, particularly when the fraction of 2D outliers was low. For the repetitive CAD structure, GOPAC retrieved some incorrect poses when 75% of the 3D points were occluded, while still finding the optimal number of inliers, due to the highly symmetric nature of the model. For the cube prior, SoftPOSIT was rarely able to find the correct pose, principally because it is unable to handle 3D points behind the camera, something the torus prior prevents. BlindPnP is also sensitive to this and, with 50 mixture model components, the method is relatively slow. However, it was necessary to use 50 components in order to obtain reasonable results for the camera pose. Moreover, both local methods have significant difficulty finding the camera pose without a strong rotation prior.

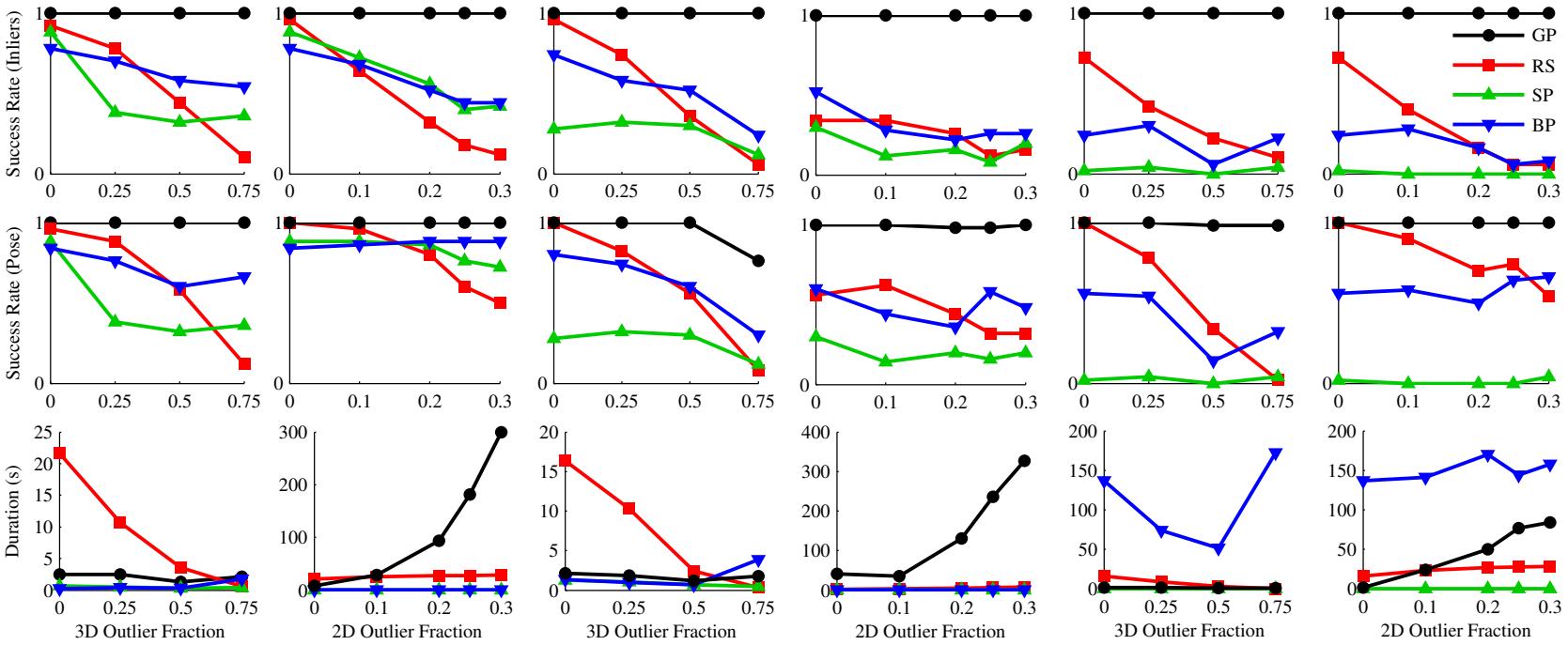
An early termination strategy, "truncated GOPAC", was also investigated following the observation that the majority of the runtime of the algorithm was spent decreasing the upper bound once the global optimum had already been attained. The experiments with the random points dataset and the torus prior were repeated with the GOPAC



(a) $\omega_{3D} = 0$ (b) $\omega_{3D} = 0.25$ (c) $\omega_{3D} = 0.5$ (d) $\omega_{3D} = 0.75$

Figure 6.14: Results for the random points dataset with the torus prior. The mean success rates and median runtimes are plotted with respect to the number of random 3D points and the 3D outlier fraction, for 50 Monte Carlo simulations per parameter value.

algorithm terminated after 30s. At termination, the algorithm returned the best-so-far cardinality and camera pose, as well as a flag to indicate that the result was not guaranteed to be optimal. Despite being terminated early, the algorithm achieved the same 100% success rates while capping the runtime at ~30s. For some applications, it may be worth sacrificing optimality for the sometimes significant decrease in runtime.



(a) Random Points, Torus Prior (b) CAD Structure, Torus Prior (c) Random Points, Cube Prior

Figure 6.15: Results for the random points ($M = 30$) and CAD structure ($M = 27$) datasets with the torus and cube priors. The mean success rates and median runtimes are plotted with respect to the 3D and 2D outlier fractions, for 50 Monte Carlo simulations per parameter value.

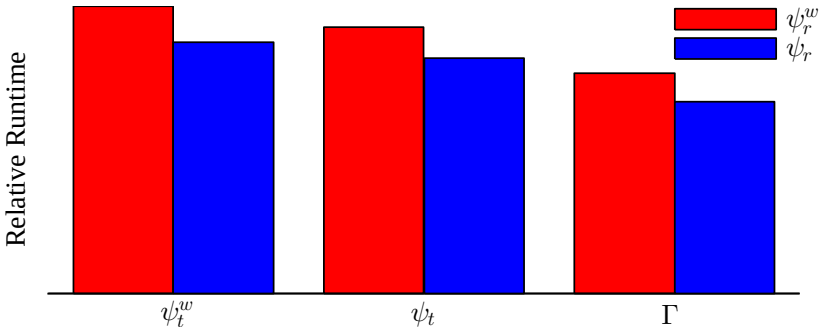


Figure 6.16: Comparison of the different upper bound functions. Runtime is plotted relative to the maximum value. The weakest upper bound using the uncertainty angles ψ_r^w and ψ_t^w (leftmost) is 50% slower than the tightest upper bound using the uncertainty angle ψ_r and bounding function Γ (rightmost).

To show the improvement attributable to the tighter upper bounds derived, the runtime of the algorithm was measured using the different upper bounds, with 10 random 3D points and 50% 2D outliers, as shown in Figure 6.16. The weak sphere-based bounding functions, using the uncertainty angles (6.6) and (6.12), are denoted ψ_r^w and ψ_t^w respectively, the tighter cuboid-based bounding functions, using the uncertainty angles (6.9) and (6.13), are denoted ψ_r and ψ_t respectively and the bounding function from (6.23) is denoted Γ. The weakest upper bound, which uses the weak uncertainty angles ψ_r^w and ψ_t^w, is 50% slower than the tightest upper bound, which uses the uncertainty angle ψ_r and the bounding function Γ.

6.6.2 Real Data Experiments

To evaluate the algorithm on real data, the Data61/2D3D (formerly NICTA) [Namin et al., 2015] and Stanford 2D-3D-Semantics (2D-3D-S) [Armeni et al., 2017] datasets were used. They are both large and repetitive multi-modal datasets with panoramic 2D images, large-scale 3D point-sets, and semantic annotations for both modalities. The former is an outdoor dataset collected from a survey vehicle with a laser scanner and 360° camera. Usefully, each 3D point is annotated with its corresponding pixels, with many points being viewed from multiple images. The ground-truth camera pose for each image can be determined from these 2D–3D correspondences. In this work the pose is obtained using EPnP [Lepetit et al., 2009] followed by nonlinear PnP [Kneip and Furgale, 2014]. The latter is an indoor dataset collected with a structured-light RGBD camera. The ground-truth camera pose is supplied for each image.

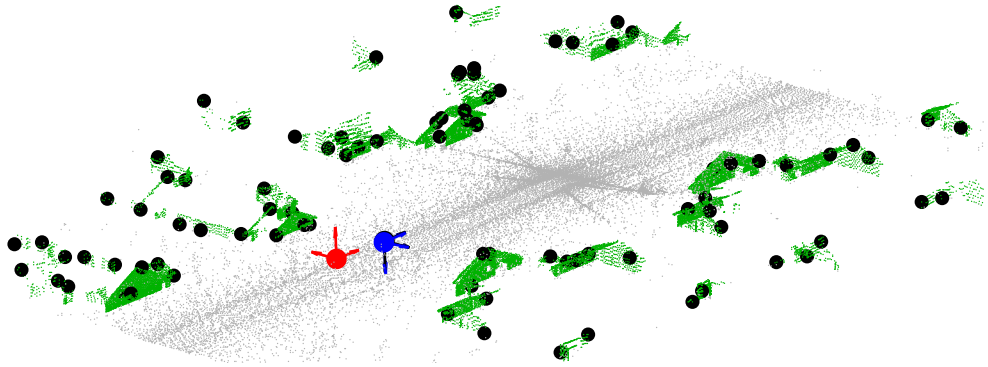
Finding the pose of a camera with respect to a point-set with only geometric (positional) information from a single image and without a good initialisation is an unsolved problem. The sub-problem of extracting points that correspond to known pixels in an

image is itself a challenging unsolved problem for 2D–3D registration pipelines. However, since GOPAC jointly solves for pose and correspondences, this problem can be relaxed to that of isolating regions of the point-set that appear in the image and vice versa. To do this, semantic labels of the images and point-set were used to select regions that were potentially observable in both modalities: building points for the outdoor dataset and furniture points for the indoor dataset. The number of selected pixels and points were then reduced to a manageable size using grid downsampling and k-means clustering, and the pixels were converted to bearing vectors using the camera calibration matrix. As a result, there is a good chance that each bearing vector has a 3D point inlier, despite not knowing the correspondences in advance.

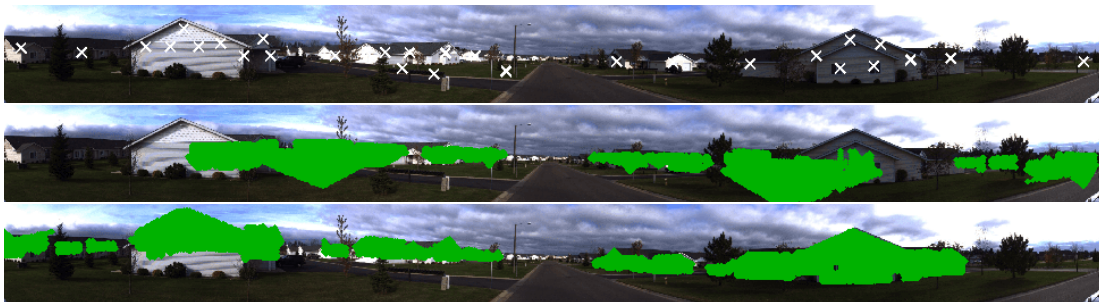
Outdoor Dataset

For the first set of experiments, two datasets were generated using this pre-processing technique for scene 1 and 5 of the Data61/2D3D dataset, shown in Figures 6.17 and 6.18, each consisting of a 3D point-set with 88 or 98 points respectively, a set of 11 images containing 30 2D features and a set of ground truth camera poses. The first dataset is a residential scene, while the second is mixed-use, with residential and industrial areas, and covers a smaller area with fewer buildings. The inlier threshold θ was set to 2°, the 2D outlier fraction guess ω_{2D} was set to 0.25 and the translation domain was set to $50 \times 5 \times 5$m, covering two lanes of the road since the camera was known to be mounted on a survey vehicle.

Qualitative results for the GOPAC and RANSAC algorithms are shown in Figures 6.17 and 6.18 and quantitative results in Tables 6.3 and 6.4, for scenes 1 and 5 respectively. GOPAC found the optimal number of inliers for all frames and the correct camera pose for the majority of frames, despite the naïvety of the 2D/3D point extraction process, surpassing the other methods. It is clear however that finding the optimal inlier set does not always correspond to finding the optimal pose for such a weak feature extraction procedure. The failure modes for GOPAC were 180° rotation flips, due to ambiguities arising from the low angular separation of points in the vertical direction. The difficulty of this ill-posed problem is illustrated by the performance of truncated GOPAC, which was not able to find all optima even after running for 30s, motivating the necessity for globally-optimal guided search. RANSAC achieved better results on the second scene than the first, since it contained fewer, denser 3D clusters. This is beneficial for RANSAC because it reduces the implicitly-searched pose space, whereas it is disadvantageous for GOPAC because it does not reduce the search space and can mean that the inlier set at the correct pose has a lower cardinality than the inlier set at some incorrect poses. Results are not shown for SoftPOSIT and BlindPnP because



(a) 3D point-set (grey and green), 3D features (black dots) and ground-truth (black), RANSAC (red) and GOPAC (blue) camera poses. The ground-truth and GOPAC camera poses coincide, whereas the RANSAC pose has a translation offset and a 180° rotation offset.

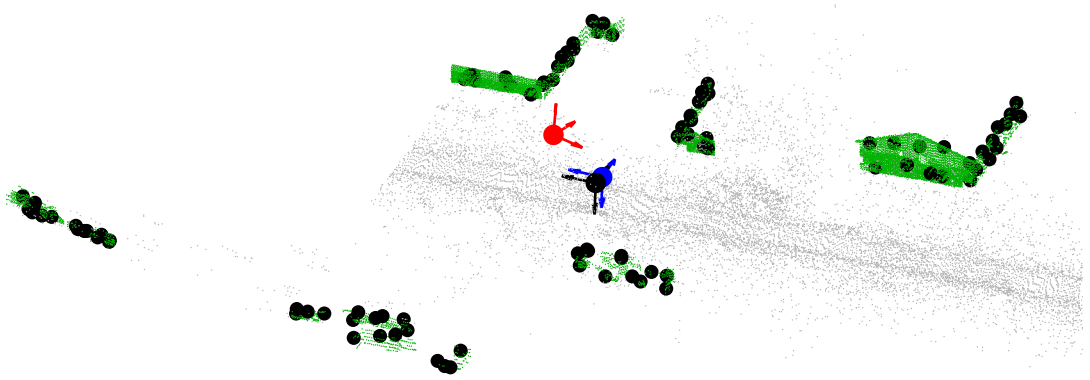


(b) Panoramic photograph and extracted 2D features (top), building points projected onto the image using the RANSAC camera pose (middle) and building points projected using the GOPAC camera pose (bottom).

Figure 6.17: Qualitative camera pose results for scene 1 of the Data61/2D3D dataset, showing the pose of the camera when capturing image 10 and the projection of 3D building points onto the image.

Table 6.3: Camera pose results for scene 1 of the Data61/2D3D dataset. Quartiles ($Q_2{}^{Q_3}_{Q_1}$) for translation error, rotation error and runtime and the mean inlier recall and success rates are reported. ⌊GOPAC⌋ denotes truncated GOPAC, where search is terminated after 30s, with no optimality guarantee. RANSAC_K denotes RANSAC with K million iterations.

Method	GOPAC	⌊GOPAC⌋	RANSAC_{20}	RANSAC_{280}
Translation Error (m)	$\mathbf{2.30}^{4.37}_{1.77}$	$3.10^{6.31}_{1.85}$	$20.3^{24.8}_{13.0}$	$28.5^{38.4}_{19.2}$
Rotation Error (°)	$\mathbf{2.08}^{3.15}_{1.75}$	$3.04^{137}_{1.92}$	$178^{179}_{90.2}$	179^{179}_{117}
Recall (Inliers)	**1.00**	0.97	0.75	0.81
Success Rate (Inliers)	**1.00**	0.45	0.00	0.00
Success Rate (Pose)	**0.82**	0.64	0.09	0.09
Runtime (s)	477^{496}_{311}	$\mathbf{34}^{34}_{33}$	$\mathbf{34}^{35}_{33}$	471^{480}_{453}



(a) 3D point-set (grey and green), 3D features (black dots) and ground-truth (black), RANSAC (red) and GOPAC (blue) camera poses. The ground-truth and GOPAC camera poses nearly coincide, whereas the RANSAC pose has a translation offset and a 180° rotation offset.

(b) Panoramic photograph and extracted 2D features (top), building points projected onto the image using the RANSAC camera pose (middle) and building points projected using the GOPAC camera pose (bottom).

Figure 6.18: Qualitative camera pose results for scene 5 of the Data61/2D3D dataset, showing the pose of the camera when capturing image 6 and the projection of 3D building points onto the image.

Table 6.4: Camera pose results for scene 5 of the Data61/2D3D dataset. Quartiles ($Q_2{}^{Q_3}_{Q_1}$) for translation error, rotation error and runtime and the mean inlier recall and success rates are reported. ⌊GOPAC⌋ denotes truncated GOPAC, where search is terminated after 30s, with no optimality guarantee. RANSAC_K denotes RANSAC with K million iterations.

Method	GOPAC	⌊GOPAC⌋	RANSAC_{20}	RANSAC_{240}
Translation Error (m)	$\mathbf{2.03}^{11.2}_{1.59}$	$2.72^{11.3}_{1.70}$	$31.5^{45.8}_{7.79}$	$4.11^{14.9}_{1.74}$
Rotation Error (°)	$\mathbf{3.28}^{179}_{2.14}$	$3.70^{179}_{1.95}$	179^{179}_{117}	$140^{179}_{3.52}$
Recall (Inliers)	**1.00**	0.98	0.78	0.86
Success Rate (Inliers)	**1.00**	0.55	0.00	0.09
Success Rate (Pose)	**0.55**	**0.55**	0.18	0.45
Runtime (s)	346^{409}_{229}	33^{33}_{33}	$\mathbf{29}^{32}_{29}$	347^{382}_{339}

Table 6.5: Comparing camera pose results for serial and parallel implementations of GOPAC for scenes 1 and 5 of the Data61/2D3D dataset. Quartiles ($Q_2{}^{Q_3}_{Q_1}$) for translation error, rotation error and runtime and the mean inlier recall and success rates are reported.

Implementation	Serial		Parallel: CPU		Parallel: GPU	
Angular Tolerance η	0	10^{-3}	0	10^{-3}	0	10^{-3}
Scene 1						
Translation Error (m)	$2.30^{4.37}_{1.72}$	$\mathbf{2.22}^{4.27}_{1.72}$	$2.30^{4.37}_{1.77}$	$2.29^{4.52}_{1.72}$	$\mathbf{2.22}^{4.57}_{1.72}$	$\mathbf{2.22}^{4.49}_{1.75}$
Rotation Error (°)	$2.18^{3.15}_{1.76}$	$\mathbf{2.08}^{3.15}_{1.80}$	$\mathbf{2.08}^{3.15}_{1.75}$	$2.09^{3.14}_{1.88}$	$2.10^{3.16}_{1.93}$	$2.09^{3.17}_{1.83}$
Recall (Inliers)	1.00	1.00	1.00	1.00	1.00	1.00
Success Rate (Inliers)	1.00	1.00	1.00	1.00	1.00	1.00
Success Rate (Pose)	0.82	0.82	0.82	0.82	0.82	0.82
Runtime (s)	614^{768}_{318}	352^{561}_{234}	477^{496}_{311}	323^{397}_{180}	8^{12}_{6}	$\mathbf{6}^{10}_{5}$
Scene 5						
Translation Error (m)	$2.03^{11.09}_{1.58}$	$\mathbf{1.80}^{11.3}_{1.58}$	$2.03^{11.2}_{1.59}$	$3.08^{11.1}_{1.66}$	$\mathbf{1.80}^{11.2}_{1.35}$	$\mathbf{1.80}^{11.6}_{1.22}$
Rotation Error (°)	$\mathbf{3.28}^{179}_{2.03}$	$\mathbf{3.28}^{179}_{2.13}$	$\mathbf{3.28}^{179}_{2.14}$	$4.24^{179}_{2.52}$	$4.30^{179}_{2.56}$	$4.25^{179}_{3.05}$
Recall (Inliers)	1.00	1.00	1.00	1.00	1.00	1.00
Success Rate (Inliers)	1.00	1.00	1.00	1.00	1.00	1.00
Success Rate (Pose)	0.55	0.55	0.55	0.55	0.55	0.55
Runtime (s)	307^{851}_{114}	205^{538}_{77}	346^{409}_{229}	222^{234}_{125}	$\mathbf{5}^{10}_{4}$	$\mathbf{5}^{9}_{4}$

they were unable to find the correct camera pose for any image in these datasets, even when supplied the ground truth pose as a prior, due to the weak ground truth correspondences and an inability to handle 3D points behind the camera. Moreover, they do not natively support panoramic imagery and required an artificially restricted field of view to function.

Quantitative results comparing the runtime of the serial and parallel (CPU and GPU) implementations of the GOPAC algorithm are shown in Tables 6.5 and Figure 6.19. The runtime of the GPU implementation is two orders of magnitude faster than the serial implementation without any loss of optimality or accuracy. In addition, the effect of relaxing the angular tolerance η from 0 (machine epsilon) to 10^{-3} radians is reported. Some reduction in runtime is observed, without any loss of optimality. However, if the angular tolerance is too large, the algorithm may discard branches containing the optimal pose. Therefore, η should be at least an order of magnitude smaller than θ.

The complete 2D qualitative results for scene 1, excluding image 10 shown in Figure 6.17, are given in Figures 6.20 and 6.21, including the two failure cases with respect to camera pose (albeit optimal with respect to the number of inliers). In both cases (images 6 and 9), a semantic segmentation error caused some of the extracted 2D fea-

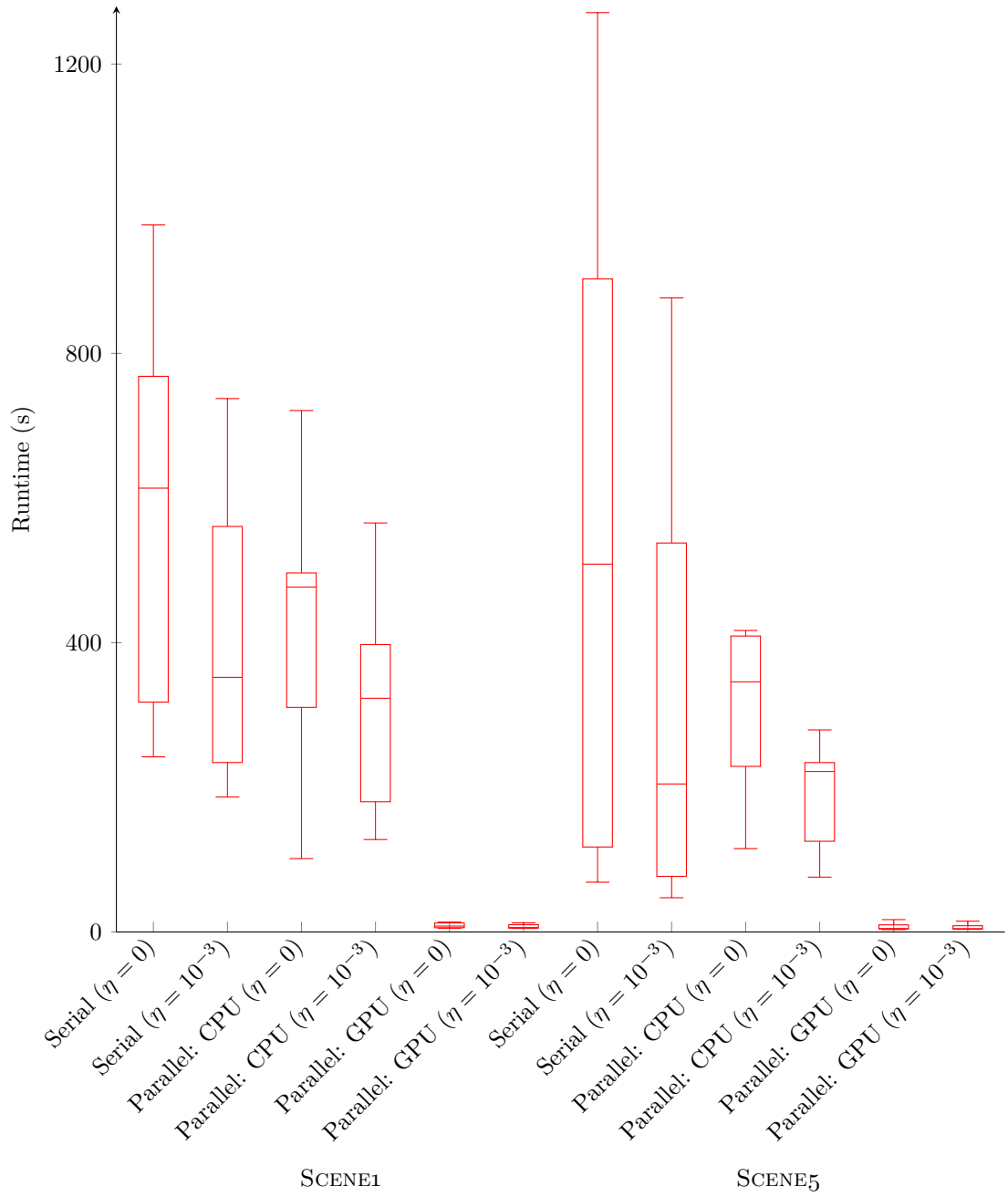
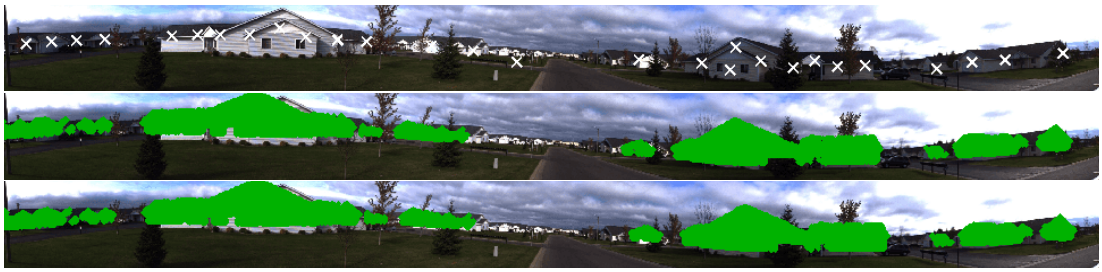


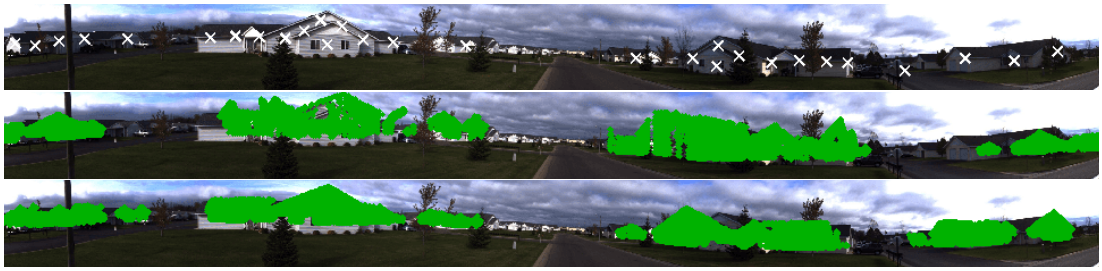
Figure 6.19: Comparing the runtime of the serial and parallel (CPU and GPU) implementations of GOPAC for scenes 1 and 5 of the Data61/2D3D dataset. The GPU implementation is two orders of magnitude faster than the serial implementation without any loss of optimality or accuracy. Relaxing the angular tolerance η from 0 (machine epsilon) to 10^{-3} radians also reduces the runtime.

tures to lie on non-building pixels, creating particularly undesirable 2D outliers. This is likely to have contributed to the algorithm finding the incorrect pose.

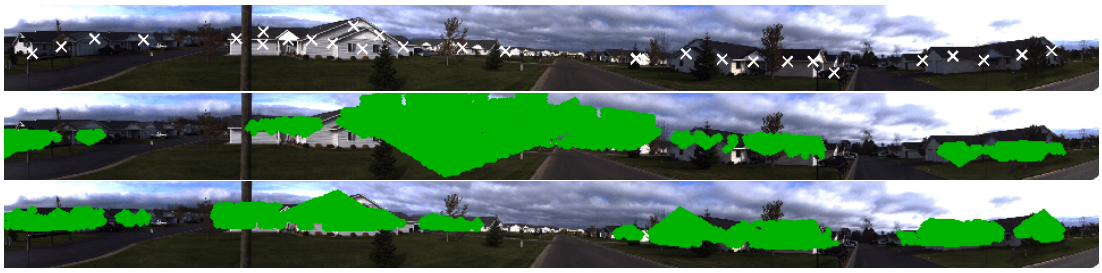
(a) Image 1



(b) Image 2



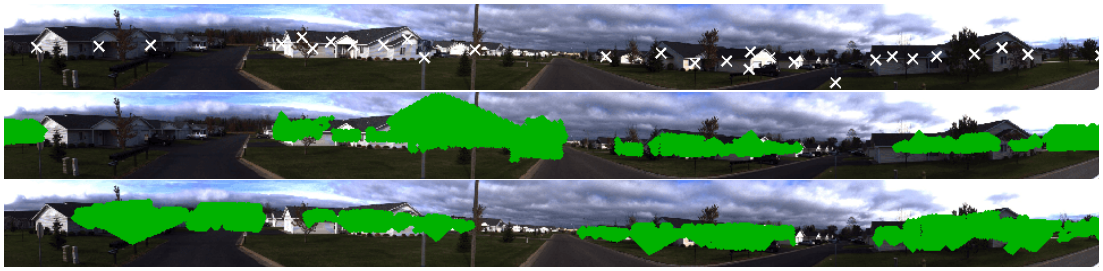
(c) Image 3



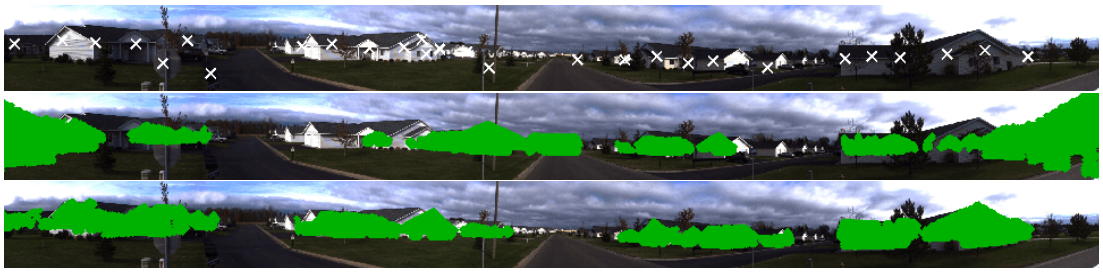
(d) Image 4

(e) Image 5

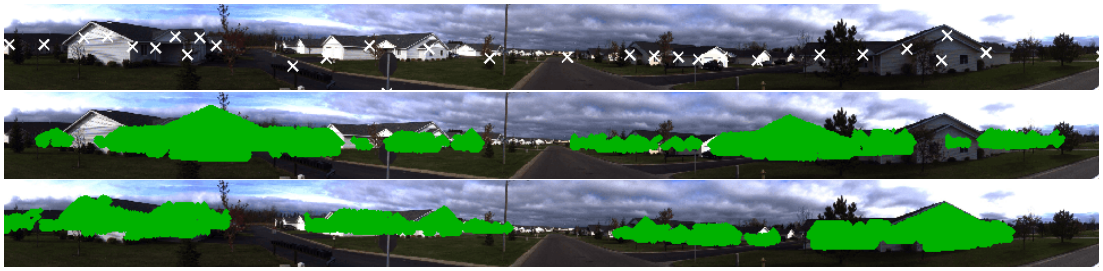
Figure 6.20: The first 5 images of scene 1 with 2D features (top) and 3D building points projected using the RANSAC (middle) and GOPAC (bottom) camera poses.



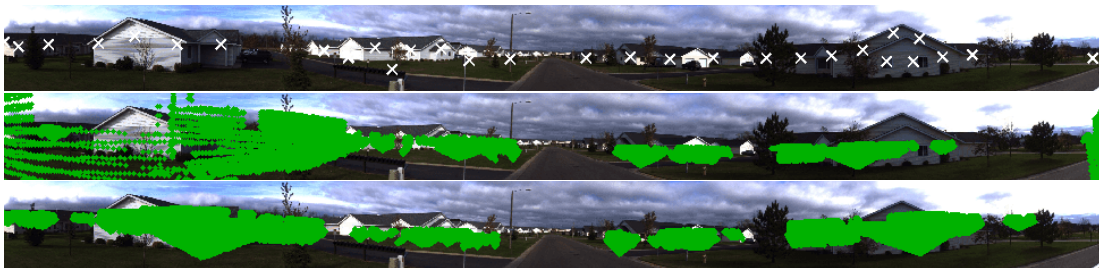
(a) Image 6



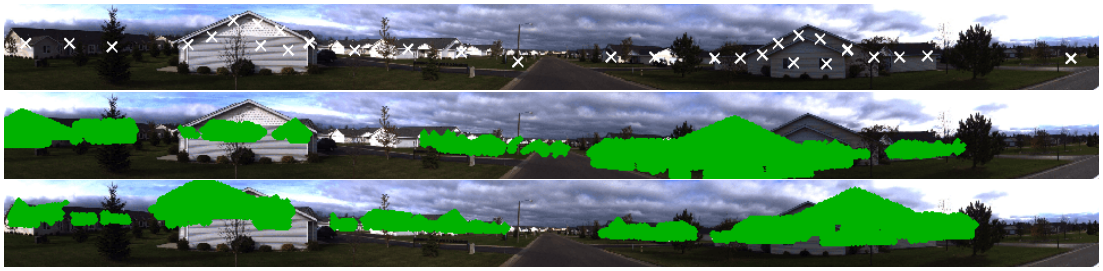
(b) Image 7



(c) Image 8



(d) Image 9



(e) Image 11

Figure 6.21: The remaining 5 images of scene 1 with 2D features (top) and 3D building points projected using the RANSAC (middle) and GOPAC (bottom) camera poses.

Table 6.6: Camera pose results for the quad-GPU implementation of GOPAC for the Data61/2D3D dataset. Quartiles ($Q_2{}^{Q_3}_{Q_1}$) for translation error, rotation error and runtime and the mean inlier recall and success rates are reported.

Scene	1	2	3	4	5
Number of 3D Points	514	572	721	314	259
Translation Error (m)	$1.11^{1.36}_{0.63}$	$0.97^{1.40}_{0.64}$	$1.06^{2.52}_{0.84}$	$1.57^{2.44}_{1.06}$	$1.12^{2.09}_{0.81}$
Rotation Error (°)	$0.70^{1.66}_{0.56}$	$1.45^{1.83}_{0.98}$	$1.51^{2.10}_{0.94}$	$1.36^{1.81}_{0.94}$	$1.15^{1.88}_{0.84}$
Recall (Inliers)	1.00	1.00	1.00	1.00	1.00
Success Rate (Inliers)	1.00	1.00	1.00	1.00	1.00
Success Rate (Pose)	1.00	1.00	1.00	1.00	1.00
Runtime (s)	15^{22}_{12}	27^{96}_{24}	11^{14}_{6}	$\mathbf{7}^{10}_{5}$	11^{17}_{9}
Scene	6	7	8	9	10
Number of 3D Points	234	245	439	819	899
Translation Error (m)	$1.12^{1.70}_{1.03}$	$\mathbf{0.34}^{0.59}_{0.26}$	$1.50^{4.40}_{0.86}$	$0.87^{1.00}_{0.69}$	$0.83^{1.59}_{0.36}$
Rotation Error (°)	$0.84^{1.14}_{0.69}$	$\mathbf{0.59}^{0.85}_{0.50}$	$1.40^{1.77}_{0.95}$	$0.83^{1.42}_{0.74}$	$1.45^{1.91}_{1.13}$
Recall (Inliers)	1.00	1.00	1.00	1.00	1.00
Success Rate (Inliers)	1.00	1.00	1.00	1.00	1.00
Success Rate (Pose)	1.00	1.00	0.91	1.00	1.00
Runtime (s)	25^{36}_{19}	$\mathbf{7}^{11}_{7}$	20^{46}_{14}	25^{32}_{17}	24^{43}_{15}

For the next set of experiments, the number of 2D and 3D features were increased to 50 2D and 500 3D features on average (2m^3 voxel downsampling). All 10 scenes from the Data61/2D3D dataset were processed, with 11 images per scene. The inlier threshold θ was set to 1°, the angular tolerance η was set to 10^{-3}, and the translation domain was set to $50 \times 5 \times 5$m. Quantitative results for the quad-GPU implementation of GOPAC are given in Table 6.6. The single pose failure case ($< 1\%$) was caused by a symmetry in the bearing vector set. In contrast, RANSAC was only able to correctly localise 13% of the images when run for 2 minutes per alignment, as shown in Table 6.7.

The effect of restricting the field-of-view to 90° was also tested by cropping the images. For scene 1, the optimal number of inliers was retrieved for every image and the correct pose was retrieved for 64% of the images. For this dataset, where the 3D features are far from the camera, many of the cropped regions do not contain discriminative features. A more sophisticated feature extraction procedure would be needed for better performance.

Indoor Data

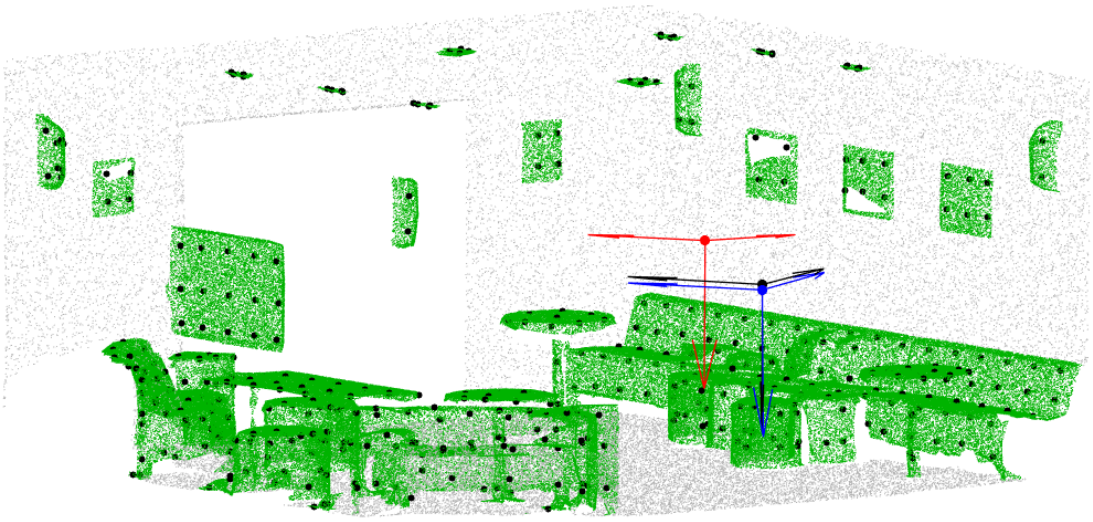
For these experiments, a dataset was generated from area 3 of the 2D-3D-S dataset, using the same pre-processing technique as the previous section with 0.3m^3 voxel down-

Table 6.7: RANSAC camera pose results for the Data61/2D3D dataset. Quartiles ($Q_2{}^{Q_3}_{Q_1}$) for translation error, rotation error and runtime and the mean inlier recall and success rates are reported.

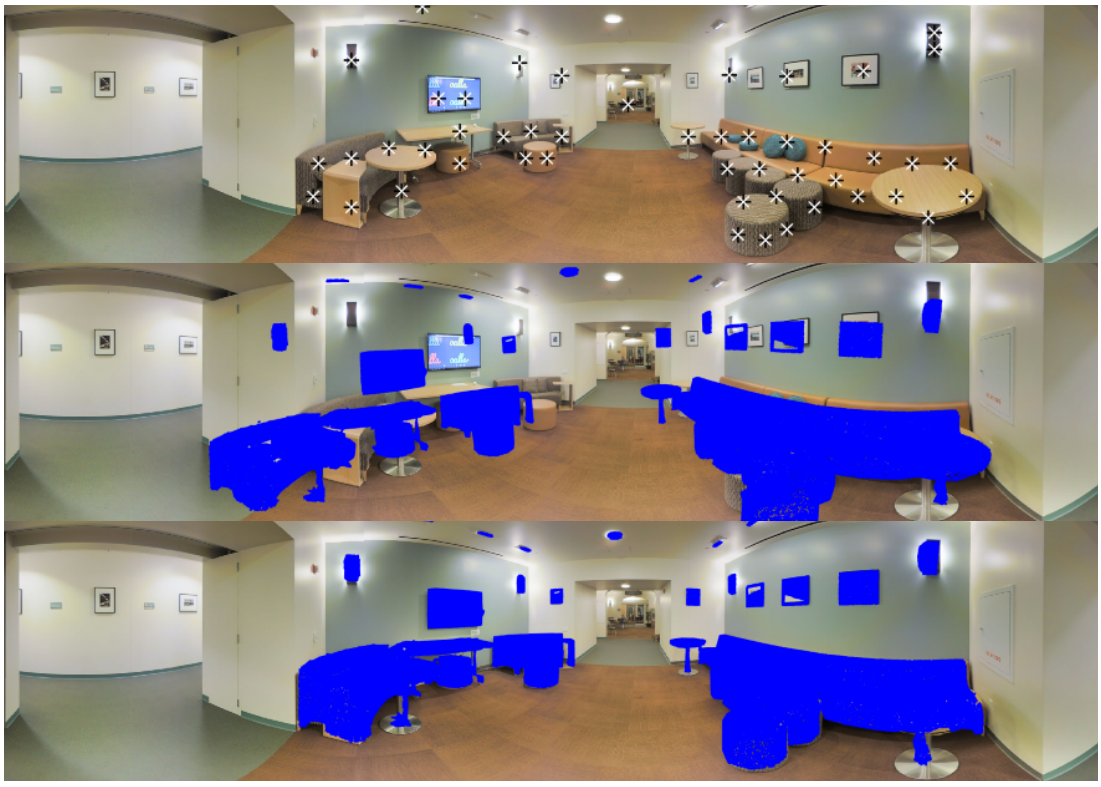
Scene	1	2	3	4	5
Number of 3D Points	514	572	721	314	259
Translation Error (m)	$11.6^{41.0}_{8.60}$	$24.2^{28.7}_{10.1}$	$86.2^{104}_{62.3}$	$76.1^{88.2}_{26.8}$	$36.2^{79.5}_{9.79}$
Rotation Error (°)	$83.4^{180}_{7.34}$	$54.7^{92.8}_{24.0}$	$176^{179}_{68.0}$	$165^{177}_{49.0}$	$173^{178}_{9.15}$
Recall (Inliers)	0.61	0.60	0.69	0.64	0.57
Success Rate (Inliers)	0.00	0.00	0.00	0.00	0.00
Success Rate (Pose)	0.09	0.18	0.00	0.00	0.18
Runtime (s)	117^{119}_{116}	120^{123}_{118}	119^{120}_{117}	120^{125}_{119}	118^{118}_{117}
Scene	6	7	8	9	10
Number of 3D Points	234	245	439	819	899
Translation Error (m)	$14.7^{26.3}_{7.74}$	$35.2^{78.2}_{5.91}$	$32.3^{45.8}_{18.0}$	$43.2^{53.2}_{29.8}$	$11.0^{19.0}_{7.69}$
Rotation Error (°)	$179^{180}_{9.72}$	$175^{178}_{4.71}$	$138^{179}_{44.2}$	$177^{179}_{59.1}$	$9.41^{177}_{4.54}$
Recall (Inliers)	0.59	0.51	0.62	0.49	0.61
Success Rate (Inliers)	0.00	0.00	0.00	0.00	0.00
Success Rate (Pose)	0.27	0.27	0.00	0.00	0.36
Runtime (s)	116^{117}_{114}	116^{116}_{115}	114^{114}_{113}	115^{117}_{114}	113^{113}_{112}

sampling. It consists of 15 rooms (lounges, offices, WCs and a conference room) and 27 sets of 50 bearing vectors, where the camera is at least 80cm from any item of furniture. The rooms were treated as separate point-sets to model visibility constraints, which assumes that the location of the camera is known to the room level. The inlier threshold θ was set to 2.5°, the angular tolerance η was set to 0.25°, and the translation domain was set to the room size. Results for the quad-GPU implementation of GOPAC and RANSAC are given in Figure 6.22 and Table 6.8.

A more sophisticated feature extraction technique was also tested, which selected corners of the walls and doors in 2D and 3D using the instance-level segmentations. For the central lounge and $\theta = 1°$, the median/maximum translation error was 0.03/0.05m, the rotation error was 0.52/0.76° and the runtime was 4/5s. This shows that additional pre-processing can greatly improve the pose accuracy and runtime of the method. However, this may require some domain-specific knowledge, since it makes a Manhattan world assumption, and is therefore less generally applicable than the downsampling pre-processing technique.



(a) 3D point-set (grey and green), 3D features (black dots) and ground-truth (black), RANSAC (red) and GOPAC (blue) camera poses.



(b) Panoramic photograph and extracted 2D features (top), furniture points projected onto the image using the RANSAC camera pose (middle) and furniture points projected using the GOPAC camera pose (bottom).

Figure 6.22: Qualitative camera pose results for lounge 1 of the Stanford 2D-3D-S dataset, showing the camera pose and the projection of the 3D furniture points onto the image.

Table 6.8: Camera pose results for the quad-GPU implementation of GOPAC and RANSAC for area 3 of the Stanford 2D-3D-S dataset. Quartiles ($Q_2{}_{Q_1}^{Q_3}$) for translation error, rotation error and runtime and the mean inlier recall and success rates are reported.

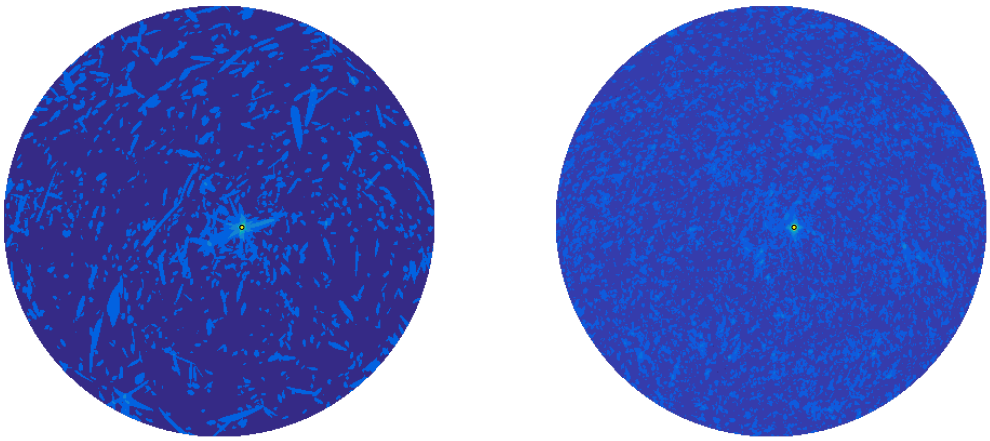
Room type	lounge		office		other	
Number of 3D Points	534		299		365	
Method	GOPAC	RANSAC	GOPAC	RANSAC	GOPAC	RANSAC
Translation error (m)	$\mathbf{0.07}^{0.14}_{0.05}$	$0.68^{1.86}_{0.34}$	$0.18^{0.34}_{0.10}$	$1.85^{3.21}_{0.28}$	$0.13^{0.19}_{0.10}$	$1.87^{2.15}_{0.61}$
Rotation error (°)	$\mathbf{1.74}^{3.23}_{1.17}$	$13.0^{52.5}_{4.94}$	$3.40^{4.64}_{2.22}$	$89.7^{171}_{11.8}$	$2.95^{3.43}_{2.60}$	$37.5^{66.4}_{28.4}$
Recall (inliers)	**1.00**	0.62	**1.00**	0.63	**1.00**	0.59
Success rate (inliers)	**1.00**	0.00	**1.00**	0.00	**1.00**	0.00
Success rate (pose)	**1.00**	0.20	0.80	0.10	**1.00**	0.14
Runtime (s)	$\mathbf{12}^{38}_{7}$	121^{123}_{121}	40^{168}_{14}	121^{122}_{120}	35^{50}_{13}	121^{122}_{120}

6.7 Discussion

As discussed in Section 3.4, there is often a mismatch between the task of optimising an objective function and the task of aligning sensor data. While a good objective function will attain its optimum at the true alignment, this is not often well-defined for sampled surfaces. However, robustness to outliers in the data is clearly a necessary condition for any objective function to be good in this sense.

Inlier set cardinality fulfils this robustness criterion by finding the largest set of consistent inlier correspondences, but the large number of outliers common to practical alignment tasks may still lead to alignment errors and ambiguities. Thus, finding the global optimum does not necessarily imply finding the ground-truth transformation. In particular, there may be false global optima created by noise and outliers or multiple global optima created by symmetries or near-symmetries in the data. Nonetheless, the results presented in Section 6.6 indicate that optimality with respect to the number of inliers closely corresponds to optimality with respect to the camera pose.

An important question is whether it is necessary to find the global optimum instead of a local optimum. The results presented in Section 6.6 answer this emphatically in the affirmative. This is unsurprising, since the objective functions are highly non-convex or non-concave, having a very large number of local optima even for a relatively small number of points, as shown in Figure 6.23. Local solvers, such as SoftPOSIT, are likely to become trapped at a local optimum near the pose prior with which the algorithm was initialised. Even when provided with a torus prior from which the true camera pose was randomly drawn, and therefore essentially running local solvers from many different poses, both SoftPOSIT and BlindPnP were unable to retrieve the correct pose in many cases. For example, SoftPOSIT and BlindPnP found incorrect poses in 50%



(a) 10 random 2D and 3D points (b) 30 random 2D and 3D points

Figure 6.23: Inlier set cardinality optima for a slice of the rotation domain passing through the optimal rotation (marked with a circle) and the Z-axis, for two alignment problems. The colours indicate the maximum number of inliers at each point in the rotation domain, evaluated by translation-only BB search, with lighter colours corresponding to greater inlier set cardinalities. Many local optima are evident, hence local optimisation in the neighbourhood of a pose prior is a bad strategy. Moreover, local optima are even more pervasive in the real alignment problem, for which rotation and translation are solved jointly.

and 38% respectively of experiments with 80 random points and 50% 3D outliers (see Table 6.1). Without the pose priors, such as in Section 6.6.2, they were not able to find the correct pose for any experiment. In contrast, RANSAC, being a global but non-optimal solver, was able to find the correct pose in some of these experiments, albeit with a significantly lower success rate than GOPAC. This shows that stochastic search over the correspondences was sometimes able to find a sufficient local optimum. However, the experiments reported in Figure 6.14 indicate that the likelihood of finding the correct pose decreases sharply as the number of points or outlier fraction increases, due to the combinatorial nature of the algorithm.

Another question is whether it is necessary to prove that the global optimum has been found. This was tested by implementing truncated GOPAC, which benefits from guided branch-and-bound search over the pose space without requiring the sometimes time-consuming proof of optimality. In this way, there can be a trade-off between the runtime and the likelihood of finding a good local optimum. Terminating the algorithm at 30s had no negative impact on the success rates for the synthetic data experiments in Section 6.6.1, however it did reduce the success rates for the real data experiments in Tables 6.3 and 6.4. It should also be noted that the correct alignment of data with near-symmetries is much more likely to be found by a globally-optimality solver. Hence, providing a guarantee of optimality may be considered a design choice, depending on the importance of reliability for the specific application.

For the challenging application of camera pose estimation using the large-scale outdoor datasets in Section 6.6.2, running the full globally-optimal GOPAC algorithm did provide a significant accuracy benefit. This can be attributed to the many alignment ambiguities and local optima created by the naïvety of the 2D/3D point extraction procedure. Downsampling and clustering the 2D pixels and 3D points is a very unsophisticated approach and is unlikely to create many true correspondences. Indeed, at the ground-truth pose, the angles between the bearing vectors and their rotationally-closest 3D points is very large. For example, only 17% of the bearing vectors have an inlier point at an angle of less than 1° and only 53% less than 2° for image 1 of scene 1 of the Data61/2D3D dataset. As a result, a large inlier threshold was required (2°), which increases the likelihood that an incorrect pose will have an equally large or larger number of inliers than the ground-truth pose.

While the naïve sampling and extraction procedure made the task much more challenging, GOPAC was still unexpectedly effective at finding the correct camera pose. This highlights the usefulness of a robust and optimal solver. However, using more sophisticated techniques in the feature extraction part of the 2D–3D alignment pipeline would tighten the correspondence between finding the optimal number of inliers and finding the correct camera pose. These could include using corner or edge detectors in 2D and 3D to find geometrically-meaningful points that are likely to have correspondences in the other modality. Regardless, using a robust and optimal solver like GOPAC relaxes the correspondence problem to one of extracting a set of 2D and 3D points that are likely to have some correspondences, without needing to know which of those points actually correspond.

Despite relaxing the correspondence problem, GOPAC has a significant limitation that was accentuated by these experiments. The algorithm has a time complexity of $\mathcal{O}(MN)$ and therefore cannot handle large numbers of points and bearing vectors without increasing the runtime substantially. As a result, it was impractical to use all points and pixels extracted by the semantic segmentation, as would be preferable. Instead, downsampling and clustering was required, which created the attendant ambiguity problems discussed previously. Moreover, GOPAC operates on discrete points, unlike the underlying task of aligning semantic regions in 2D and 3D. This motivates a surface-based approach, aligning regions on the unit sphere.

In Section 6.3, it was observed that using an asymmetric inlier measure could lead to sets of degenerate poses. Symmetric inlier measures offer some advantages, but introduce significant additional computation and may still be susceptible to these degenerate configurations. For example, measures that add or multiply the number of 2D and 3D inliers can still lead to degenerate poses, particularly if the cardinality of either the 2D or 3D data dominates the other. Furthermore, enforcing one-to-one cor-

respondences does not circumvent the second type of degeneracy, where the camera is brought close to dense regions of 3D points. In addition, many-to-one correspondences are sometimes appropriate, particularly for point-sets with low density regions.

An ideal objective function would be symmetric, cardinality-invariant, and reach its optimum at a pose that explains as much of the 2D and 3D information as possible. However, it is not clear how to optimally balance the two. One alternative would be to maximise the ratio of mutual information to joint entropy, that is, the intersection over union or Jaccard Index, on the sphere but within the image boundaries. The bearing vectors and 3D points induce inlier cones with half-aperture angle θ, which form circles on the sphere that can be used to calculate the Jaccard Index. Using the inlier cones for the alignment measure also partially circumvents the effect of variable sampling densities. However, not only is this symmetric measure very difficult to compute, the visibility constraints also vary with the pose, that is, the 3D points that were observable from that pose for a given field-of-view. As a result, it would be extremely challenging to find bounds for this function.

6.8 Transferring the Theoretical Framework

The theoretical framework developed in Section 6.4 can be transferred to other objective functions for 2D–3D registration. While the inlier set cardinality objective function has many advantages, being inherently robust to outliers, finding an exact optimiser and operating on raw sensor data, other objective functions may also have useful characteristics. One example is the L_2 distance between mixture models, which is inherently robust to outliers and operates on statistical densities generated from the raw sensor data. The densities model the underlying surfaces of the scene, which is beneficial because the fundamental 2D–3D registration problem is a surface alignment problem, not a discrete sample alignment problem.

In this section, theoretical insights from the cardinality maximisation approach are transferred to an L_2 distance minimisation approach. Firstly, the objective function will be introduced, with some discussion about the practical considerations that constrain its form. Next, the required bounding functions will be developed. This section is intended to outline the approach, so implementation details will not be considered.

6.8.1 L_2 Distance Between Mixture Models

Given the argument of Chapter 5, it is natural to consider whether a mixture model approach is suitable for the 2D–3D registration problem. While the L_2 distance between mixture models fulfils the criterion of being robust to outliers, tractability becomes a challenge in the 2D–3D case for two reasons.

Firstly, mixture models of directional data, such as those discussed in Section 3.3.6, lie on the unit sphere S^2, a more complex space than $\mathbb{R}^3$. As a result, most do not have a closed-form probability density function. An exception to this is the von Mises–Fisher Mixture Model (vMFMM), which has a closed-form probability density function (3.35), can admit arbitrarily accurate estimates of noisy sphere-projected surface densities, and can be computed efficiently from point-set or bearing vector set data. The trade-off is that the von Mises–Fisher distribution [Fisher, 1953] is isotropic and therefore less expressive than other probability distributions on the sphere, such as the non-isotropic Fisher-Bingham distribution [Kent, 1982].

Secondly, challenges arise from the structure of the problem, since 2D–3D registration involves positional data (3D points) as well as directional data (bearing vectors). While the directional data cannot be elevated to positional data unless the pixel depths are known, positional data can be projected onto the unit sphere for a fixed camera translation. Thus the L_2 distance between mixture models for 2D–3D registration must be defined on the sphere and a means of projecting the positional data onto the sphere must be established, both of which have tractability challenges.

As discussed in Section 3.3.5, a Gaussian Mixture Model (GMM) can be estimated from positional data to model the underlying surfaces of the scene. The projection of a Gaussian distribution in $\mathbb{R}^3$ to the unit sphere S^2 is given by the Projected Normal (PN) distribution [Mardia, 1972; Wang and Gelfand, 2013]. A Projected Normal Mixture Model (PNMM) is useful for modelling a 3D scene as observed by a camera and has a probability distribution function given by (3.45) for isotropic Gaussian components. However, the L_2 distance between a vMFMM (bearing vectors) and a PNMM (3D points) is not tractable, since it does not simplify to a closed form when integrated over the sphere and would therefore require time-consuming numerical integration.

Instead, a simplified projection from $\mathbb{R}^3$ to S^2 is considered, mapping an isotropic GMM to an over-parametrised vMFMM. Each component of the vMFMM is given by the parameter set $\{\hat{\boldsymbol{\mu}}_i, \kappa_i, \phi_i\}$ with mean direction $\hat{\boldsymbol{\mu}}_i \in S^2$, concentration $\kappa_i > 0$ and mixture weight $\phi_i \geqslant 0$ calculated according to

$$\hat{\boldsymbol{\mu}}_i = \frac{\boldsymbol{\mu}'_i}{\|\boldsymbol{\mu}'_i\|} \tag{6.86}$$

$$\kappa_i = \left(\frac{\|\boldsymbol{\mu}'_i\|}{\sigma'_i}\right)^2 + 1 \tag{6.87}$$

$$\phi_i = \phi'_i \tag{6.88}$$

where $\{\boldsymbol{\mu}'_i, \sigma'_i, \phi'_i\}$ is the parameter set of the GMM component with mean $\boldsymbol{\mu}'_i \in \mathbb{R}^3$, standard deviation $\sigma'_i > 0$ and mixture weight $\phi'_i \geqslant 0$. The primary requirement of

the projection is that the concentration κ increases with the square of the length of the Gaussian mean vector for each component, since the further away the Gaussian, the more certain its direction. Equation (6.87) can be derived by comparing the vMF and PN distributions. It is important that their probability density functions evaluate to similar values in the direction of the mean vector. Setting $\angle(\mathbf{f}, \hat{\boldsymbol{\mu}}) = 0$, the vMF probability density function (3.36) becomes

$$\mathrm{vMF}(\mathbf{f}|\hat{\boldsymbol{\mu}}, \kappa) = \frac{\kappa}{2\pi(1 - \exp(-2\kappa))} \approx \frac{\kappa}{2\pi} \tag{6.89}$$

and the PN probability density function (3.45) becomes

$$\mathrm{PN}(\mathbf{f}|\boldsymbol{\mu}', \sigma') = \frac{1}{(2\pi)^{\frac{3}{2}}} \exp\left(-\frac{1}{2}\rho^2\right)\left[\rho + \sqrt{2\pi}\Phi(\rho)\exp\left(\frac{1}{2}\rho^2\right)\left(1 + \rho^2\right)\right] \tag{6.90}$$

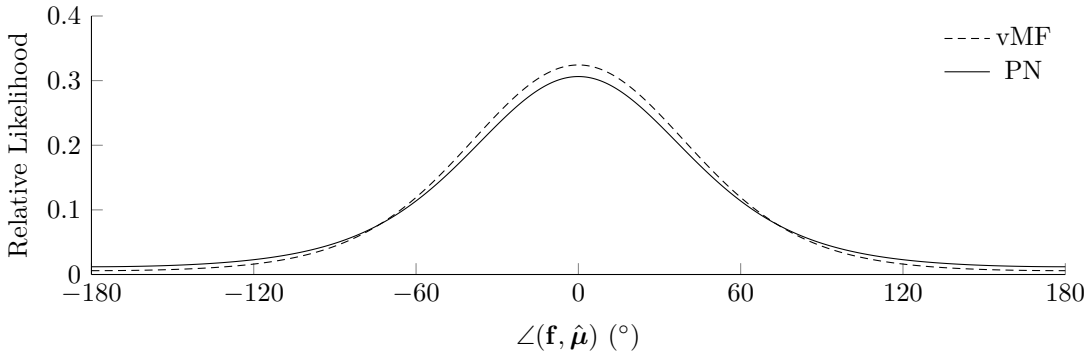
for $\rho = \|\boldsymbol{\mu}'\|/\sigma'$. As $\rho \to \infty$, it becomes

$$\mathrm{PN}(\mathbf{f}|\boldsymbol{\mu}', \sigma') = \frac{1}{2\pi}\left(\left(\frac{\|\boldsymbol{\mu}'\|}{\sigma'}\right)^2 + 1\right). \tag{6.91}$$

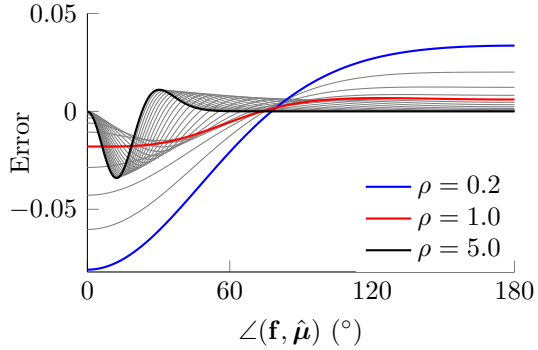
Equating (6.89) and (6.91) gives the equation for κ (6.87). Therefore the simplified projection is close to the true PN distribution at the mean vector when $\|\boldsymbol{\mu}'\| \gg \sigma'$.

To show that the projected vMFMM is close to the true PNMM, the discrepancy between the vMFMM and PNMM for each component was quantified across a range of values of $\rho = \|\boldsymbol{\mu}'\|/\sigma'$ and $\alpha = \angle(\mathbf{f}, \hat{\boldsymbol{\mu}})$. The two distributions are very similar, even for relatively low values of ρ, as shown in Figure 6.24(a) for $\rho = 1$. For example, the root mean square error across the entire angular range is less than 0.01 for all $\rho \geqslant 1$. Figure 6.24(b) shows the error in relative likelihood as the angle α between the mean vector and the evaluation vector increases and Figure 6.24(c) shows the root mean square error across the entire angular range as ρ increases.

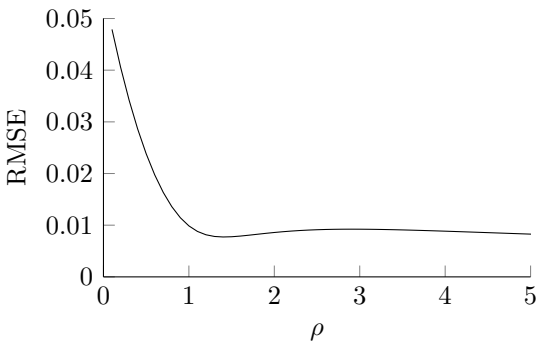
Moreover, this simplified projection reduces the objective function to the L_2 distance between vMFMMs, which is robust to outliers and can be calculated in closed-form. This function was used by Straub et al. [2017] for 3D–3D rotational alignment, however here it is used for 2D–3D rotational and translational alignment. Unlike the standard L_2 distance between vMFMMs, one of the vMFMMs is a projection from a GMM and is therefore function of camera translation. As a result, the objective function is a function of both camera rotation and translation.



(a) PN and vMF probability density functions with respect to $\angle(\mathbf{f}, \hat{\boldsymbol{\mu}})$ for $\rho = 1$



(b) Error with respect to $\angle(\mathbf{f}, \hat{\boldsymbol{\mu}})$



(c) RMS error with respect to ρ

Figure 6.24: Comparison of the true PN distribution and vMF approximation for projecting an isotropic Gaussian distribution onto the unit sphere. (a) The vMF and PN probability density functions are plotted with respect to the angle $\angle(\mathbf{f}, \hat{\boldsymbol{\mu}})$ between the mean direction $\hat{\boldsymbol{\mu}}$ and the evaluation direction $\mathbf{f}$ for $\rho = 1$. The distributions are very close, even for a relatively small $\rho = 1$. (b) The error PN − vMF is plotted with respect to the angle $\angle(\mathbf{f}, \hat{\boldsymbol{\mu}})$ for a range of values of $\rho \in [0.2, 5]$ at intervals of 0.2. For $\rho \geqslant 1$, the error is small and tends to 0 as the angle increases. (c) The Root Mean Square Error (RMSE) across the entire angular range is plotted with respect to ρ and is less than 0.01 for all $\rho \geqslant 1$.

The derivation of the L_2 distance between vMFMMs was given in Section 3.4.7. For 2D–3D sensor data alignment, the first term of the expanded L_2 distance is not invariant to translation and therefore cannot be dropped. Let $\boldsymbol{\theta}_1 = \{\boldsymbol{\mu}_{1i}, \sigma_{1i}, \phi_{1i}\}_{i=1}^{n_1}$ be the parameter set of an n_1-component GMM with isotropic covariances generated from the point-set $\mathcal{P}$, with means $\boldsymbol{\mu}_{1i}$, standard deviations $\sigma_{1i} \geqslant 0$, and mixture weights $\phi_{1i} \geqslant 0$, where $\sum_{i=1}^{n_1} \phi_{1i} = 1$. Also let $\boldsymbol{\theta}_2 = \{\hat{\boldsymbol{\mu}}_{2j}, \kappa_{2j}, \phi_{2j}\}_{j=1}^{n_2}$ be the parameter set of an n_2-component vMFMM generated from the bearing vector set $\mathcal{F}$ with mean directions $\hat{\boldsymbol{\mu}}_{2j} \in S^2$, concentrations $\kappa_{2j} > 0$, and mixture weights $\phi_{2j} \geqslant 0$, where $\sum_{j=1}^{n_2} \phi_{2j} = 1$. Then the objective is to find a rotation $\mathbf{R} \in SO(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$ that minimises the L_2 distance between the projected GMM and the vMFMM

$$d_{L_2}^* = \min_{\mathbf{R},\,\mathbf{t}} f(\mathbf{R}, \mathbf{t}) \tag{6.92}$$

$$f(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{n_1}\sum_{j=1}^{n_1} \frac{\phi_{1i}\phi_{1j} Z(K_{1i1j}(\mathbf{t}))}{Z(\kappa_{1i}(\mathbf{t}))Z(\kappa_{1j}(\mathbf{t}))} - 2\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \frac{\phi_{1i}\phi_{2j} Z(K_{1i2j}(\mathbf{R}, \mathbf{t}))}{Z(\kappa_{1i}(\mathbf{t}))Z(\kappa_{2j})} \tag{6.93}$$

where

$$Z(x) = \frac{\sinh(x)}{x} = \frac{\exp(x) - \exp(-x)}{2x}, \tag{6.94}$$

$$K_{1i1j}(\mathbf{t}) \triangleq \left\| \kappa_{1i}(\mathbf{t}) \frac{\boldsymbol{\mu}_{1i} - \mathbf{t}}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}\|} + \kappa_{1j}(\mathbf{t}) \frac{\boldsymbol{\mu}_{1j} - \mathbf{t}}{\|\boldsymbol{\mu}_{1j} - \mathbf{t}\|} \right\|, \tag{6.95}$$

$$K_{1i2j}(\mathbf{R}, \mathbf{t}) \triangleq \left\| \kappa_{1i}(\mathbf{t}) \mathbf{R} \frac{\boldsymbol{\mu}_{1i} - \mathbf{t}}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}\|} + \kappa_{2j} \hat{\boldsymbol{\mu}}_{2j} \right\|, \text{ and} \tag{6.96}$$

$$\kappa_{1i}(\mathbf{t}) = \left(\frac{\|\boldsymbol{\mu}_{1i} - \mathbf{t}\|}{\sigma_{1i}} \right)^2 + 1. \tag{6.97}$$

The function value $d_{L_2} = f(\mathbf{R}, \mathbf{t})$ is equal to the L_2 distance up to a constant factor ($1/4\pi$) and addition by a constant. Also, $Z(x)$ is a monotonically increasing function for $x \geqslant 0$ with $Z(x) \geqslant 1$.

While there are many differences between this objective function and the inlier set cardinality objective function, the one that has the largest effect on bounding the function is the dependence of κ on $\|\boldsymbol{\mu} - \mathbf{t}\|$. That is, the function is dependent on the distance to each translated Gaussian mean, unlike the inlier set cardinality function which is independent of the distance to each 3D point. This occurs because the L_2 distance between mixture models is a surface alignment method, whereas the inlier set cardinality is a point alignment method. Specifically, the (infinitesimal) surface area of a projected point does not scale with the square of the distance to the point, unlike the surface area of a projected Gaussian.

6.8.2 Bounding Functions

Similar arguments to those used in Section 6.4.2 can be used to bound the objective function (6.93) within a transformation domain $\mathcal{C}_r \times \mathcal{C}_t$. An upper bound can be found by evaluating the function at any transformation in the branch. The transformation at the centre of the rotation and translation cuboids is convenient and quick to evaluate.

Theorem 6.6. *(Upper bound) For the transformation domain $\mathcal{C}_r \times \mathcal{C}_t$ centred at $(\mathbf{r}_0, \mathbf{t}_0)$, an upper bound of the L_2 distance between the projected GMM and the vMFMM is*

$$\bar{d}_{L_2} \triangleq f(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}_0). \tag{6.98}$$

Proof. The validity of the upper bound follows from

$$f(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}_0) \geqslant \min_{\substack{\mathbf{r}\in\mathcal{C}_r \\ \mathbf{t}\in\mathcal{C}_t}} f(\mathbf{R}_{\mathbf{r}}, \mathbf{t}). \tag{6.99}$$

That is, the function value at a specific point within the domain is greater than or equal to the minimum within the domain. □

A lower bound on the objective function within a transformation domain $\mathcal{C}_r \times \mathcal{C}_t$ can be found using the bounds on the uncertainty angles ψ_r and ψ_t.

Theorem 6.7. *(Lower bound) For the transformation domain $\mathcal{C}_r \times \mathcal{C}_t$ centred at $(\mathbf{r}_0, \mathbf{t}_0)$, a lower bound of the L_2 distance between the projected GMM and the vMFMM is*

$$\underline{d}_{L_2} \triangleq \sum_{i=1}^{n_1}\sum_{j=1}^{n_1} \frac{\phi_{1i}\phi_{1j} Z(\underline{K}_{1i1j}(\mathcal{C}_t))}{Z(\bar{\kappa}_{1i}(\mathcal{C}_t))Z(\bar{\kappa}_{1j}(\mathcal{C}_t))} - 2\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \frac{\phi_{1i}\phi_{2j} Z\Big(\bar{K}_{1i2j}(\mathcal{C}_r, \mathcal{C}_t)\Big)}{Z(\underline{\kappa}_{1i}(\mathcal{C}_t))Z(\kappa_{2j})} \tag{6.100}$$

where

$$\begin{aligned}\underline{K}_{1i1j}(\mathcal{C}_t) \triangleq \max\Big\{0, K_{1i1j}(\mathbf{t}_0) &- \sqrt{\bar{\kappa}_{1i}^2(\mathcal{C}_t) + \kappa_{1i}^2(\mathbf{t}_0) - 2\bar{\kappa}_{1i}(\mathcal{C}_t)\kappa_{1i}(\mathbf{t}_0)\cos\psi_t(\boldsymbol{\mu}_{1i}, \mathcal{C}_t)} \\ &- \sqrt{\bar{\kappa}_{1j}^2(\mathcal{C}_t) + \kappa_{1j}^2(\mathbf{t}_0) - 2\bar{\kappa}_{1j}(\mathcal{C}_t)\kappa_{1j}(\mathbf{t}_0)\cos\psi_t(\boldsymbol{\mu}_{1j}, \mathcal{C}_t)}\Big\}\end{aligned} \tag{6.101}$$

$$\begin{aligned}\bar{K}_{1i2j}(\mathcal{C}_r, \mathcal{C}_t) \triangleq K_{1i2j}(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}_0) &+ \sqrt{\bar{\kappa}_{1i}^2(\mathcal{C}_t) + \kappa_{1i}^2(\mathbf{t}_0) - 2\bar{\kappa}_{1i}(\mathcal{C}_t)\kappa_{1i}(\mathbf{t}_0)\cos\psi_t(\boldsymbol{\mu}_{1i}, \mathcal{C}_t)} \\ &+ \kappa_{2j}\sqrt{2 - 2\cos\psi_r(\hat{\boldsymbol{\mu}}_{2j}, \mathcal{C}_r)}\end{aligned} \tag{6.102}$$

$$\bar{\kappa}_{1i}(\mathcal{C}_t) \triangleq \left(\frac{\max_{\mathbf{t}\in\mathcal{V}_t}\|\boldsymbol{\mu}_{1i} - \mathbf{t}\|}{\sigma_{1i}}\right)^2 + 1 \tag{6.103}$$

$$\underline{\kappa}_{1i}(\mathcal{C}_t) \triangleq \left(\frac{\min_{\mathbf{t}\in\mathcal{C}_t}\|\boldsymbol{\mu}_{1i} - \mathbf{t}\|}{\sigma_{1i}}\right)^2 + 1. \tag{6.104}$$

Proof. Observe that $\forall \mathbf{t} \in \mathcal{C}_t$,

$$K_{1i1j}(\mathbf{t}) = \left\| \kappa_{1i}(\mathbf{t}) \frac{\boldsymbol{\mu}_{1i} - \mathbf{t}}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}\|} + \kappa_{1j}(\mathbf{t}) \frac{\boldsymbol{\mu}_{1j} - \mathbf{t}}{\|\boldsymbol{\mu}_{1j} - \mathbf{t}\|} \right\| \tag{6.105}$$

$$\begin{aligned} &= \left\| \kappa_{1i}(\mathbf{t}_0) \frac{\boldsymbol{\mu}_{1i} - \mathbf{t}_0}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}_0\|} + \kappa_{1j}(\mathbf{t}_0) \frac{\boldsymbol{\mu}_{1j} - \mathbf{t}_0}{\|\boldsymbol{\mu}_{1j} - \mathbf{t}_0\|} \right. \\ &\quad - \left(\kappa_{1i}(\mathbf{t}_0) \frac{\boldsymbol{\mu}_{1i} - \mathbf{t}_0}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}_0\|} - \kappa_{1i}(\mathbf{t}) \frac{\boldsymbol{\mu}_{1i} - \mathbf{t}}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}\|} \right) \\ &\quad \left. - \left(\kappa_{1j}(\mathbf{t}_0) \frac{\boldsymbol{\mu}_{1j} - \mathbf{t}_0}{\|\boldsymbol{\mu}_{1j} - \mathbf{t}_0\|} - \kappa_{1j}(\mathbf{t}) \frac{\boldsymbol{\mu}_{1j} - \mathbf{t}}{\|\boldsymbol{\mu}_{1j} - \mathbf{t}\|} \right) \right\| \end{aligned} \tag{6.106}$$

$$\begin{aligned} &\geqslant \left\| \left\| \kappa_{1i}(\mathbf{t}_0) \frac{\boldsymbol{\mu}_{1i} - \mathbf{t}_0}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}_0\|} + \kappa_{1j}(\mathbf{t}_0) \frac{\boldsymbol{\mu}_{1j} - \mathbf{t}_0}{\|\boldsymbol{\mu}_{1j} - \mathbf{t}_0\|} \right\| \right. \\ &\quad - \left\| \left(\kappa_{1i}(\mathbf{t}_0) \frac{\boldsymbol{\mu}_{1i} - \mathbf{t}_0}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}_0\|} - \kappa_{1i}(\mathbf{t}) \frac{\boldsymbol{\mu}_{1i} - \mathbf{t}}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}\|} \right) \right. \\ &\quad \left. \left. + \left(\kappa_{1j}(\mathbf{t}_0) \frac{\boldsymbol{\mu}_{1j} - \mathbf{t}_0}{\|\boldsymbol{\mu}_{1j} - \mathbf{t}_0\|} - \kappa_{1j}(\mathbf{t}) \frac{\boldsymbol{\mu}_{1j} - \mathbf{t}}{\|\boldsymbol{\mu}_{1j} - \mathbf{t}\|} \right) \right\| \right\| \end{aligned} \tag{6.107}$$

$$\begin{aligned} &\geqslant \max \left\{ 0, \left\| \kappa_{1i}(\mathbf{t}_0) \frac{\boldsymbol{\mu}_{1i} - \mathbf{t}_0}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}_0\|} + \kappa_{1j}(\mathbf{t}_0) \frac{\boldsymbol{\mu}_{1j} - \mathbf{t}_0}{\|\boldsymbol{\mu}_{1j} - \mathbf{t}_0\|} \right\| \right. \\ &\quad - \left\| \kappa_{1i}(\mathbf{t}_0) \frac{\boldsymbol{\mu}_{1i} - \mathbf{t}_0}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}_0\|} - \kappa_{1i}(\mathbf{t}) \frac{\boldsymbol{\mu}_{1i} - \mathbf{t}}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}\|} \right\| \\ &\quad \left. - \left\| \kappa_{1j}(\mathbf{t}_0) \frac{\boldsymbol{\mu}_{1j} - \mathbf{t}_0}{\|\boldsymbol{\mu}_{1j} - \mathbf{t}_0\|} - \kappa_{1j}(\mathbf{t}) \frac{\boldsymbol{\mu}_{1j} - \mathbf{t}}{\|\boldsymbol{\mu}_{1j} - \mathbf{t}\|} \right\| \right\} \end{aligned} \tag{6.108}$$

$$\begin{aligned} &= \max \left\{ 0, K_{1i1j}(\mathbf{t}_0) \right. \\ &\quad - \sqrt{\kappa_{1i}^2(\mathbf{t}) + \kappa_{1i}^2(\mathbf{t}_0) - 2\kappa_{1i}(\mathbf{t})\kappa_{1i}(\mathbf{t}_0) \cos \angle \left(\frac{\boldsymbol{\mu}_{1i} - \mathbf{t}}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}\|}, \frac{\boldsymbol{\mu}_{1i} - \mathbf{t}_0}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}_0\|} \right)} \\ &\quad \left. - \sqrt{\kappa_{1j}^2(\mathbf{t}) + \kappa_{1j}^2(\mathbf{t}_0) - 2\kappa_{1j}(\mathbf{t})\kappa_{1j}(\mathbf{t}_0) \cos \angle \left(\frac{\boldsymbol{\mu}_{1j} - \mathbf{t}}{\|\boldsymbol{\mu}_{1j} - \mathbf{t}\|}, \frac{\boldsymbol{\mu}_{1j} - \mathbf{t}_0}{\|\boldsymbol{\mu}_{1j} - \mathbf{t}_0\|} \right)} \right\} \end{aligned} \tag{6.109}$$

$$\begin{aligned} &\geqslant \max \left\{ 0, K_{1i1j}(\mathbf{t}_0) \right. \\ &\quad - \sqrt{\bar{\kappa}_{1i}^2(\mathcal{C}_t) + \kappa_{1i}^2(\mathbf{t}_0) - 2\bar{\kappa}_{1i}(\mathcal{C}_t)\kappa_{1i}(\mathbf{t}_0) \cos \psi_t(\boldsymbol{\mu}_{1i}, \mathcal{C}_t)} \\ &\quad \left. - \sqrt{\bar{\kappa}_{1j}^2(\mathcal{C}_t) + \kappa_{1j}^2(\mathbf{t}_0) - 2\bar{\kappa}_{1j}(\mathcal{C}_t)\kappa_{1j}(\mathbf{t}_0) \cos \psi_t(\boldsymbol{\mu}_{1j}, \mathcal{C}_t)} \right\} \end{aligned} \tag{6.110}$$

where (6.107) and (6.108) follow from the (reverse) triangle inequality, (6.109) follows from the cosine rule, and (6.110) follows from Lemma 6.4.

Also observe that $\forall(\mathbf{r},\mathbf{t}) \in (\mathcal{C}_r \times \mathcal{C}_t)$,

$$K_{1i2j}(\mathbf{R}_\mathbf{r},\mathbf{t}) = \left\| \kappa_{1i}(\mathbf{t}) \frac{\boldsymbol{\mu}_{1i} - \mathbf{t}}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}\|} + \kappa_{2j} \mathbf{R}_\mathbf{r}^{-1} \hat{\boldsymbol{\mu}}_{2j} \right\| \tag{6.111}$$

$$\begin{aligned} &= \left\| \kappa_{1i}(\mathbf{t}_0) \frac{\boldsymbol{\mu}_{1i} - \mathbf{t}_0}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}_0\|} + \kappa_{2j} \mathbf{R}_{\mathbf{r}_0}^{-1} \hat{\boldsymbol{\mu}}_{2j} \right. \\ &\quad + \left(\kappa_{1i}(\mathbf{t}) \frac{\boldsymbol{\mu}_{1i} - \mathbf{t}}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}\|} - \kappa_{1i}(\mathbf{t}_0) \frac{\boldsymbol{\mu}_{1i} - \mathbf{t}_0}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}_0\|} \right) \\ &\quad \left. + \kappa_{2j} \left(\mathbf{R}_\mathbf{r}^{-1} \hat{\boldsymbol{\mu}}_{2j} - \mathbf{R}_{\mathbf{r}_0}^{-1} \hat{\boldsymbol{\mu}}_{2j} \right) \right\| \end{aligned} \tag{6.112}$$

$$\begin{aligned} &\leqslant \left\| \kappa_{1i}(\mathbf{t}_0) \frac{\boldsymbol{\mu}_{1i} - \mathbf{t}_0}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}_0\|} + \kappa_{2j} \mathbf{R}_{\mathbf{r}_0}^{-1} \hat{\boldsymbol{\mu}}_{2j} \right\| \\ &\quad + \left\| \kappa_{1i}(\mathbf{t}) \frac{\boldsymbol{\mu}_{1i} - \mathbf{t}}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}\|} - \kappa_{1i}(\mathbf{t}_0) \frac{\boldsymbol{\mu}_{1i} - \mathbf{t}_0}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}_0\|} \right\| \\ &\quad + \kappa_{2j} \left\| \mathbf{R}_\mathbf{r}^{-1} \hat{\boldsymbol{\mu}}_{2j} - \mathbf{R}_{\mathbf{r}_0}^{-1} \hat{\boldsymbol{\mu}}_{2j} \right\| \end{aligned} \tag{6.113}$$

$$\begin{aligned} &= K_{1i2j}(\mathbf{R}_{\mathbf{r}_0},\mathbf{t}_0) \\ &\quad + \sqrt{\kappa_{1i}^2(\mathbf{t}) + \kappa_{1i}^2(\mathbf{t}_0) - 2\kappa_{1i}(\mathbf{t})\kappa_{1i}(\mathbf{t}_0) \cos \angle \left(\frac{\boldsymbol{\mu}_{1i} - \mathbf{t}}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}\|}, \frac{\boldsymbol{\mu}_{1i} - \mathbf{t}_0}{\|\boldsymbol{\mu}_{1i} - \mathbf{t}_0\|} \right)} \\ &\quad + \kappa_{2j} \sqrt{2 - 2\cos \angle \left(\mathbf{R}_\mathbf{r}^{-1} \hat{\boldsymbol{\mu}}_{2j}, \mathbf{R}_{\mathbf{r}_0}^{-1} \hat{\boldsymbol{\mu}}_{2j} \right)} \end{aligned} \tag{6.114}$$

$$\begin{aligned} &\leqslant K_{1i2j}(\mathbf{R}_{\mathbf{r}_0},\mathbf{t}_0) \\ &\quad + \sqrt{\bar{\kappa}_{1i}^2(\mathcal{C}_t) + \kappa_{1i}^2(\mathbf{t}_0) - 2\bar{\kappa}_{1i}(\mathcal{C}_t)\kappa_{1i}(\mathbf{t}_0) \cos \psi_t(\boldsymbol{\mu}_{1i}, \mathcal{C}_t)} \\ &\quad + \kappa_{2j} \sqrt{2 - 2\cos \psi_r(\hat{\boldsymbol{\mu}}_{2j}, \mathcal{C}_r)} \end{aligned} \tag{6.115}$$

where (6.113) follows from the triangle inequality, (6.114) follows from the cosine rule, and (6.115) follows from Lemmas 6.2 and 6.4. □

These bounds can be implemented using the same 2D–3D registration framework developed in Sections 6.4 and 6.5 and the same rotation and translation uncertainty angles ψ_r and ψ_t. Many of the implementation details from Section 6.5 can be transferred to this approach, including nesting rotation search inside translation search.

However, one significant difference is that the L_2 distance approach can only achieve ϵ-suboptimality, not full optimality, since the bounding functions converge asymptotically as the branch sizes decrease but only coincide at infinitesimal branch sizes. Therefore, the user is required to set a small value ϵ such that the solution will be guaranteed to be within ϵ of the true global optimum.

6.9 Summary

This chapter developed a theoretical framework for robust and globally-optimal 2D–3D registration and demonstrated how this could be used to solve the simultaneous camera pose and correspondence problem using inlier set cardinality maximisation. The method applied the branch-and-bound paradigm to guarantee global optimality regardless of initialisation and used local optimisation to accelerate convergence. The pivotal contribution was the derivation of the objective function bounds using the geometry of $SE(3)$. The algorithm outperformed other local and global methods on challenging synthetic and real datasets, finding the global optimum reliably, with a GPU implementation greatly reducing runtime. These experiments provided evidence that a robust objective function and global optimality are critical for reliable 2D–3D alignment. Finally, another robust and globally-optimal approach, minimising the L_2 distance between mixture models, was outlined using the same framework developed for the cardinality maximisation algorithm. This demonstrated how theoretical insights from the first algorithm can be transferred to develop algorithms with other objective functions that have different properties.

Further investigation is warranted to develop a complete 2D–3D alignment pipeline. Extracting structural features that exist in both the image and point-set is itself a challenging problem. The insight that semantic segmentations in 2D and 3D can isolate regions that are potentially observable in both modalities merits further research, making use of recent developments in 2D–3D segmentation. Finally, several opportunities arise from the mixture model approach, including a novel local solver based on the objective function with coarse-to-fine annealing on the number of mixture components, trading off optimality for speed.

The following chapter will bring together the conclusions drawn in this and previous chapters on the necessity of robust objective functions and global optimality for geometric alignment tasks. It will also summarise the contributions made by this thesis and will discuss directions for ongoing and future research germinated by these investigations.

Chapter 7

Conclusions

This thesis has addressed the problem of geometric sensor data alignment, finding the rigid transformation that correctly aligns one set of sensor data with another without any prior knowledge about how the data correspond. In this investigation, the objective was to develop tractable algorithms for geometric sensor data alignment that were robust to outliers and not susceptible to spurious local optima. This was considered a valuable pursuit because outliers are highly prevalent in sensor data and alignment problems are highly non-convex, key challenges that have not been fully and jointly resolved by the literature. As a response to these challenges, this thesis presented the case that robust and optimal methods are necessary for geometric sensor data alignment without correspondences to handle outliers and non-convexity.

The approach taken in this thesis was to first analyse the limitations of the prior art in geometric sensor data alignment, with particular attention to the literature that proposed solutions to the nD–nD and 2D–3D alignment problems studied in this thesis. This analysis identified outliers and non-convexity as the key challenges inherent to the alignment problem. The analysis further identified that most alignment methods failed to consider both factors or treated one of the factors as a *post hoc* requirement, not a core feature. For example, several authors developed algorithms that were not robust to outliers, for which techniques to robustify the approach, such as trimming, were applied retrospectively. Other authors developed algorithms that were susceptible to local optima, for which stochastic global optimisation techniques, such as a random-start strategy, were applied after the fact. Following this analysis, the approach taken was to consider the challenges presented by outliers and non-convexity from the outset when developing geometric alignment algorithms. That is, building robustness to outliers directly into the algorithms through intrinsically robust objective functions, and reducing susceptibility to local optima from the initial choice of optimisation technique.

In this chapter, the primary and secondary contributions of the investigation are outlined, limitations of the approaches taken are discussed, and future work stemming from the research is considered.

7.1 Contributions

The major contributions of this thesis were:

1. A novel positional sensor data representation, the Support Vector–parametrised Gaussian Mixture (SVGM), that is sparsely-parametrised, discriminative and efficient to compute. As a sparse parametrisation of the data that adapts to local surface complexity, it is time efficient, having fewer components to align, and memory efficient, compressing the data without sacrificing model fidelity. As a discriminative model, it has better viewpoint invariance than generative models since it does not model sampling artefacts, such as varying point density and occlusion, to the same extent.
2. A novel local optimisation algorithm, Support Vector Registration (SVR), for aligning positional sensor data under the robust L_2 distance between densities. In comparison to other local optimisation algorithms, SVR manifested a greater robustness to outliers and sampling artefacts, a wider region of convergence, and a superior time complexity. In comparison to existing global optimisation algorithms, SVR had a better trade-off between accuracy and speed, except for very large motions.
3. A novel global optimisation algorithm, Globally-Optimal Gaussian Mixture Alignment (GOGMA), for optimally aligning 3D positional sensor data under the robust L_2 distance between densities. GOGMA was the first optimal solution proposed for 3D–3D geometric alignment with an inherently robust objective function. The pivotal contribution was the derivation of novel bounds on the objective function using the geometry of $SE(3)$. These were used within a parallel branch-and-bound framework to guarantee global optimality regardless of initialisation. In comparison to other 3D–3D alignment algorithms, GOGMA performed more robustly on challenging datasets due to its guaranteed optimality and outlier robustness, without unduly increasing the runtime.
4. A novel global optimisation algorithm, Globally-Optimal Pose And Correspondences (GOPAC), for optimally aligning 2D directional and 3D positional sensor data under the robust inlier set cardinality objective function. GOPAC was the first optimal solution proposed for 2D–3D alignment with an inherently robust objective function. The pivotal contribution was the derivation of novel bounds on the objective function using the geometry of $SE(3)$. These were used within a nested branch-and-bound framework to guarantee global optimality regardless of initialisation. In comparison to other 2D–3D alignment algorithms, GOPAC performed more robustly on challenging datasets due to its guaranteed optimality and outlier robustness, with a GPU implementation greatly reducing runtime.

5. A novel global optimisation algorithm for optimally aligning 2D directional and 3D positional sensor data under the robust L_2 distance between densities. The pivotal contributions incorporated into this surface alignment algorithm were the derivations of a novel projection of Gaussian mixture models onto the unit sphere and novel bounds on the closed-form objective function.

Secondary contributions of this thesis included:

1. An analysis of discriminative and generative models for sensor data, showing that discriminative models have useful properties for alignment problems, such as a robustness to sampling artefacts including varying point density and occlusion.
2. A novel and time-efficient algorithm, GMMerge, for merging aligned mixture models without retaining redundant components or weighting the intersection regions disproportionately. GMMerge is useful for reconstruction and mapping applications, particularly those that have time and memory limitations.
3. A tight and novel bound on the rotation uncertainty distance that can be directly transferred to other 3D–3D geometric alignment algorithms that use branch-and-bound [Yang et al., 2016] to improve the quality of their bounds and the runtime of their implementations.
4. Tight and novel bounds on the rotation and translation uncertainty angles that can be directly transferred to other branch-and-bound geometric alignment algorithms [Brown et al., 2015; Yang et al., 2016; Parra Bustos et al., 2016] to improve the quality of their bounds and the runtime of their implementations.
5. A tighter objective function bound that considers the interaction between the elements of both datasets (see Section 6.4.2). The same insight can be applied to tighten the bounds of other branch-and-bound geometric alignment algorithms [Brown et al., 2015; Yang et al., 2016; Campbell and Petersson, 2016; Parra Bustos et al., 2016].
6. Insights into how branch-and-bound methods for alignment problems can be made more efficient, including the use of sophisticated data structures, projections, and pre-computation.
7. A parallel branch-and-bound framework that implemented an adaptive branching strategy, significantly reducing redundant branching and computation. The strategy orders the the dimensions to subdivide by their angular uncertainty.
8. An analysis of the novel spherical projection of a Gaussian mixture model, including the discrepancy between the projection and the true distribution, the projected normal mixture model.

7.2 Limitations of the Approach

The proposed algorithms have several limitations that restrict their applicability. The most significant limitations include:

1. **Time Complexity and Runtime:** The primary limitation of the algorithms developed in this thesis is their theoretical time complexity and their runtime in practice. All of the algorithms have a quadratic time complexity of $\mathcal{O}(MN)$ with respect to the input size, where M and N are the number of Gaussian components or the number of points. As a result, scenes and objects cannot be modelled to a high resolution without increasing the runtime significantly. This in turn increases the ambiguity of the alignment problem, since the optimal alignment of low resolution models may not correspond to the correct alignment of the underlying surfaces. More importantly, the runtime of the branch-and-bound algorithms can be quite high in practice, due to the size of the search space, and is data-dependent. This excludes real-time applications and those that require a consistent runtime.
2. **Alignment Objective Functions:** Another limitation is attributable to the mismatch between the task of optimising an objective function and the task of aligning sensor data. While the robust objective functions used in this thesis typically attain their optimum at the true alignment, this is not always the case. False optima can result from the sampling process itself, as well as symmetries or near-symmetries in the data. This reflects the underlying nature of the problem as that of aligning surfaces, which are only approximated by the discrete samples.
3. **Degenerate Cases:** A limitation of the GOPAC algorithm is the degenerate poses that can result from optimising the asymmetric objective function. While degenerate configurations are unavoidable for 2D–3D alignment due to the nature of the problem, they can be exacerbated by the choice of objective function. However, objective functions with fewer degenerate cases can be non-trivial to compute and bound.
4. **Transformation Classes:** The scope of this investigation was limited to rigid transformations, to the exclusion of affine, projective, piecewise-rigid and non-rigid transformations. As a result, the algorithms proposed in this thesis cannot be applied directly to non-rigid problems, such as face and body alignment.

7.3 Ongoing and Future Work

There are several theoretical and technical areas of this research that warrant further analysis and investigation:

1. **Data Representations:** For the mixture model approaches, estimating full covariance matrices from the data would increase the representational power of the model while requiring fewer components. For example, more spherical Gaussians are required to model flat surfaces than would be required by anisotropic Gaussians. Once estimated, an SVM could be trained with the full covariances [Abe, 2005] to produce a discriminative model. For the SVR algorithm, incorporating the full covariances into the objective function and optimisation process is not problematic, as shown in Section 3.4.6. However, tractability becomes a challenge for mixture models on the sphere, which do not have a closed-form in the anisotropic case, and for the optimal methods, which would require new bounds on the Mahalanobis distance to remain valid. Finally, approximate algorithms could be applied to reduce the training time of the SVGM data representation [Joachims, 1999; Tsang et al., 2005].
2. **Data Structures:** Further research could usefully explore how data structures could be applied to improve the time efficiency of the algorithms. The SVR and GOGMA algorithms evaluate a discrete Gauss transform, the sum of n_1 Gaussians at n_2 points in D dimensions, as part of the L_2 distance computation. This imposes a time complexity of $\mathcal{O}(n_1 n_2)$ and dominates the complexity behaviour of the algorithms. The time complexity can be reduced to $\mathcal{O}(n_1 + n_2)$ by using an approximation such as the fast Gauss transform [Greengard and Strain, 1991] or improved fast Gauss transform [Yang et al., 2003] and an associated data structure with memory requirements proportional to $n_1 + n_2$. A similar data structure could be applied for estimating the L_2 distance between mixture models on the sphere. Alternatively, a data structure could be designed by analogy to the distance transform, storing the set of K least-attenuated Gaussians at each point in $\mathbb{R}^3$ and reducing the time complexity to $\mathcal{O}(K n_1)$. This makes use of the observation that Gaussians far from any given point have very little influence on the function value at that point, due to the rapid attenuation of Gaussians. In this formulation, the second mixture could have an arbitrary number of components without affecting the runtime, suitable for large-scale maps.
3. **Parallel Processing:** A natural progression of this work is to analyse how advanced parallel processing techniques could be applied to the existing parallel implementations. For example, implementing a dynamic branching factor would reduce redundant computation and allow for more parallelism at the same memory requirements. Furthermore, using dynamic parallelism would make it possible to implement the nested branch-and-bound structure on the GPU. This has several advantages, including better memory and computational efficiency and the ability to precompute uncertainty angles and transformations. Finally, runtime

benefits could also be realised by implementing local optimisation on the GPU, reducing the amount of serial processing required.

4. **Optimisation Techniques:** Another possible area of future research would be to investigate the application of more sophisticated optimisation techniques to the problem, particularly for the local optimisation approaches used in SVR and GOGMA. For example, the number of iterations required could be reduced by utilising analytically-expressed Hessian matrices in the optimiser. In addition, techniques for expanding the search domain, such as random-start optimisation or particle filtering, could be applied.
5. **Transformation Class:** Extending the class of transformations handled to the non-rigid case would be a useful addition. For SVR, this would be trivial, since a direct application of the approach of Jian and Vemuri [2011] would be sufficient. For the other algorithms, this would not currently be tractable, since the dimensionality of the problem is already very high for a branch-and-bound approach.
6. **2D–3D Alignment Pipeline:** Further investigation is warranted to develop a complete 2D–3D alignment pipeline. In particular, the insight that semantic segmentations can be used to extract features that are observable in both modalities merits further research. Exploiting recent developments in multi-modal classification is a natural way to incorporate appearance information into geometric alignment problems.

Bibliography

Abe, S., 2005. Training of support vector machines with Mahalanobis kernels. In *Proceedings of the 15th International Conference on Artificial Neural Networks: Formal Models and Their Applications – Part II* (Warsaw, Poland, Sep. 2005), 571–576. Springer. 124, 225

Abramowitz, M. and Stegun, I. A., 1964. Handbook of mathematical functions with formulas, graphs, and mathematical tables. *National Bureau of Standards, Applied Mathematics Series*, 55 (Jun. 1964). 69

Ai, S.; Jia, L.; Zhuang, C.; and Ding, H., 2017. A registration method for 3D point clouds with convolutional neural network. In *Proceedings of the 2017 International Conference on Intelligent Robotics and Applications* (Wuhan, China, Aug. 2017), 377–387. Springer. 29

Aiger, D.; Mitra, N. J.; and Cohen-Or, D., 2008. 4-points congruent sets for robust pairwise surface registration. *ACM Transactions on Graphics*, 27, 3 (Aug. 2008), 85:1–85:10. doi: 10.1145/1360612.1360684. 6, 8, 22, 27, 76, 90, 106, 129

Aizerman, A.; Braverman, E.; and Rozoner, L., 1964. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25 (1964), 821–837. 108

Akutsu, T.; Tamaki, H.; and Tokuyama, T., 1998. Distribution of distances and triangles in a point set and algorithms for computing the largest common point sets. *Discrete and Computational Geometry*, 20, 3 (Oct. 1998), 307–331. doi: 10.1007/PL00009388. 77

Albarelli, A.; Rodola, E.; and Torsello, A., 2010. A game-theoretic approach to fine surface registration without initial motion estimation. In *Proceedings of the 2010 Conference on Computer Vision and Pattern Recognition* (San Francisco, CA, USA, Jun. 2010), 430–437. IEEE. doi: 10.1109/CVPR.2010.5540183. 20

Antoniak, C. E., 1974. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 2 (Nov. 1974), 1152–1174. 65, 130

Armeni, I.; Sax, A.; Zamir, A. R.; and Savarese, S., 2017. Joint 2D-3D-semantic data for indoor scene understanding. *ArXiv e-prints*, (Feb. 2017). URL http://arxiv.org/abs/1702.01105. 198

Arun, K. S.; Huang, T. S.; and Blostein, S. D., 1987. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9, 5 (Sep. 1987), 698–700. doi: 10.1109/TPAMI.1987.4767965. 19, 72

Ask, E.; Enqvist, O.; and Kahl, F., 2013. Optimal geometric fitting under the truncated L_2-norm. In *Proceedings of the 2013 Conference on Computer Vision and Pattern Recognition* (Portland, OR, USA, Jun. 2013), 1722–1729. IEEE. doi: 10.1109/CVPR.2013.225. 20, 21, 34, 35, 166

Aubry, M.; Maturana, D.; Efros, A. A.; Russell, B. C.; and Sivic, J., 2014. Seeing 3D chairs: exemplar part-based 2D-3D alignment using a large dataset of CAD models. In *Proceedings of the 2014 Conference on Computer Vision and Pattern Recognition* (Columbus, OH, USA, Jun. 2014), 3762–3769. IEEE. doi: 10.1109/CVPR.2014.487. 6, 162

Aubry, M.; Schlickewei, U.; and Cremers, D., 2011. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Proceedings of the 2011 International Conference on Computer Vision Workshops* (Barcelona, Spain, Nov. 2011), 1626–1633. IEEE. doi: 10.1109/ICCVW.2011.6130444. 18

Ayache, N. and Faugeras, O. D., 1986. HYPER: A new approach for the recognition and positioning of two-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8, 1 (Jan. 1986), 44–54. doi: 10.1109/TPAMI.1986.4767751. 39

Ballard, D. H., 1981. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13, 2 (Sep. 1981), 111–122. doi: 10.1016/0031-3203(81)90009-1. 27

Banerjee, A.; Dhillon, I. S.; Ghosh, J.; and Sra, S., 2005. Clustering on the unit hypersphere using von Mises–Fisher distributions. *Journal of Machine Learning Research*, 6 (Dec. 2005), 1345–1382. 68

Basu, A.; Harris, I. R.; Hjort, N. L.; and Jones, M., 1998. Robust and efficient estimation by minimising a density power divergence. *Biometrika*, 85, 3 (Sep. 1998), 549–559. 80, 82

Bazin, J.-C.; Li, H.; Kweon, I. S.; Demonceaux, C.; Vasseur, P.; and Ikeuchi, K., 2013. A branch-and-bound approach to correspondence and grouping problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35, 7 (Jul. 2013), 1565–1576. doi: 10.1109/TPAMI.2012.264. 6, 30, 36, 42

Bazin, J.-C.; Seo, Y.; and Pollefeys, M., 2012. Globally optimal consensus set maximization through rotation search. In *Proceedings of the 2012 Asian Conference on Computer Vision* (Daejeon, Korea, 2012), 539–551. Springer. doi: 10.1007/978-3-642-37444-9_42. 20, 22, 129

Beis, J. S. and Lowe, D. G., 1999. Indexing without invariants in 3D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21, 10 (1999), 1000–1015. 40

Belongie, S.; Malik, J.; and Puzicha, J., 2002. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24, 4 (2002), 509–522. 6, 16, 103, 126

Besl, P. J. and McKay, N. D., 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14, 2 (1992), 239–256. 8, 9, 22, 23, 71, 72, 102, 103, 117, 119, 126, 147, 152, 166

Beveridge, J. R. and Riseman, E. M., 1995. Optimal geometric model matching under full 3D perspective. *Computer Vision and Image Understanding*, 61, 3 (1995), 351–364. 37, 39

Biber, P. and Strasser, W., 2003. The Normal Distributions Transform: A new approach to laser scan matching. In *Proceedings of the 2003 International Conference on Intelligent Robots and Systems*, vol. 3 (Las Vegas, NV, USA, 2003), 2743–2748. IEEE. 25

Blais, G. and Levine, M. D., 1995. Registering multiview range data to create 3D computer objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17, 8 (1995), 820–824. 6, 16, 28, 102, 126, 129

Bookstein, F. L., 1989. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11, 6 (1989), 567–585. 24

Boscaini, D.; Masci, J.; Melzi, S.; Bronstein, M. M.; Castellani, U.; and Vandergheynst, P., 2015. Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. *Computer Graphics Forum*, 34, 5 (Jul. 2015), 13–23. doi: 10.1111/cgf.12693. 18

Boscaini, D.; Masci, J.; Rodolà, E.; and Bronstein, M., 2016a. Learning shape correspondence with anisotropic convolutional neural networks. In *Proceedings of the 2016 International Conference on Neural Information Processing Systems* (Barcelona, Spain, Dec. 2016), 3189–3197. Curran Associates Inc. 18

Boscaini, D.; Masci, J.; Rodolà, E.; Bronstein, M. M.; and Cremers, D., 2016b. Anisotropic diffusion descriptors. *Computer Graphics Forum*, 35, 2 (May 2016), 431–441. doi: 10.1111/cgf.12844. 18

Bosse, M.; Zlot, R.; and Flick, P., 2012. Zebedee: Design of a spring-mounted 3-D range sensor with application to mobile mapping. *IEEE Transactions on Robotics*, 28, 5 (Oct. 2012), 1104–1119. 61

Boyer, E.; Bronstein, A. M.; Bronstein, M. M.; Bustos, B.; Darom, T.; Horaud, R.; Hotz, I.; Keller, Y.; Keustermans, J.; Kovnatsky, A.; Litman, R.; Reininghaus, J.; Sipiran, I.; Smeets, D.; Suetens, P.; Vandermeulen, D.; Zaharescu, A.; and Zobel, V., 2011. SHREC 2011: Robust feature detection and description benchmark. In *Proceedings of the 4th Eurographics Conference on 3D Object Retrieval* (Llandudno, UK, 2011), 71–78. Eurographics Association, Aire-la-Ville, Switzerland. doi: 10.2312/3DOR/3DOR11/071-078. 17

Brachmann, E.; Krull, A.; Nowozin, S.; Shotton, J.; Michel, F.; Gumhold, S.; and Rother, C., 2017. DSAC – Differentiable RANSAC for camera localization. In *Proceedings of the 2017 Conference on Computer Vision and Pattern Recognition* (Honolulu, Hawaii, USA, Jul. 2017), 2492–2500. doi: 10.1109/CVPR.2017.267. 29, 33, 40, 41

Bregman, L. M., 1967. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7, 3 (1967), 200–217. 80

Breuel, T. M., 1992. Fast recognition using adaptive subdivisions of transformation space. In *Proceedings of the 1992 Conference on Computer Vision and Pattern Recognition* (Champaign, IL, USA, Jun. 1992), 445–451. IEEE. 29, 39

Breuel, T. M., 2003. Implementation techniques for geometric branch-and-bound matching methods. *Computer Vision and Image Understanding*, 90, 3 (Jun. 2003), 258–294. doi: 10.1016/S1077-3142(03)00026-2. 8, 9, 29, 30, 37, 41, 42, 76, 166, 167

Bronstein, M. M. and Kokkinos, I., 2010. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *Proceedings of the 2010 Conference on Computer Vision and Pattern Recognition* (San Francisco, CA, USA, Jun. 2010), 1704–1711. IEEE. 18

Brown, M.; Windridge, D.; and Guillemaut, J.-Y., 2015. Globally optimal 2D-3D registration from points or lines without correspondences. In *Proceedings of the 2015 International Conference on Computer Vision* (Santiago, Chile, Dec. 2015), 2111–2119. 8, 10, 37, 42, 43, 75, 163, 164, 165, 167, 173, 176, 177, 180, 191, 192, 223

Bruneau, P.; Gelgon, M.; and Picarougne, F., 2010. Parsimonious reduction of Gaussian mixture models with a variational-Bayes approach. *Pattern Recognition*, 43, 3 (Mar. 2010), 850–858. 115

Bülow, H. and Birk, A., 2013. Spectral 6DOF registration of noisy 3D range data with partial overlap. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35, 4 (Apr. 2013), 954–969. 28

Burns, J. B.; Weiss, R. S.; and Riseman, E. M., 1993. View variation of point-set and line-segment features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15, 1 (Jan. 1993), 51–68. doi: 10.1109/34.184774. 40

Byrd, R. H.; Lu, P.; Nocedal, J.; and Zhu, C., 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16, 5 (1995), 1190–1208. 131

Campbell, D. and Petersson, L., 2015. An adaptive data representation for robust point-set registration and merging. In *Proceedings of the 2015 International Conference on Computer Vision* (Santiago, Chile, Dec. 2015), 4292–4300. IEEE. doi: 10.1109/ICCV.2015.488. 25, 65, 126, 127, 128, 130, 131, 148

Campbell, D. and Petersson, L., 2016. GOGMA: Globally-Optimal Gaussian Mixture Alignment. In *Proceedings of the 2016 Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV, USA, Jun. 2016), 5685–5694. IEEE. doi: 10.1109/CVPR.2016.613. 8, 10, 31, 42, 43, 97, 167, 223

Campbell, D.; Petersson, L.; Kneip, L.; and Li, H., 2017. Globally-optimal inlier set maximisation for simultaneous camera pose and feature correspondence. In *Proceedings of the 2017 International Conference on Computer Vision* (Venice, Italy, Oct. 2017), 1–10. IEEE. doi: 10.1109/ICCV.2017.10. 8, 10, 42

Campbell, D.; Petersson, L.; Kneip, L.; and Li, H., 2018. Globally-optimal inlier set maximisation for camera pose and correspondence estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (Jun. 2018), preprint. doi: 10.1109/TPAMI.2018.2848650.

Canny, J., 1986. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8, 6 (Nov. 1986), 679–698. 33

Carr, J. C.; Beatson, R. K.; Cherrie, J. B.; Mitchell, T. J.; Fright, W. R.; McCallum, B. C.; and Evans, T. R., 2001. Reconstruction and representation of 3D objects with radial basis functions. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, 67–76. ACM. doi: 10.1145/383259.383266. 122

Cass, T. A., 1997. Polynomial-time geometric matching for object recognition. *International Journal of Computer Vision*, 21, 1 (Jan. 1997), 37–61. 37, 39

Castellani, U. and Bartoli, A., 2012. 3D shape registration. In *3D Imaging, Analysis and Applications* (Eds. N. Pears; Y. Liu; and P. Bunting), 221–264. Springer, London. ISBN 978-1-4471-4063-4. doi: 10.1007/978-1-4471-4063-4_6. 23

Chang, C.-C. and Lin, C.-J., 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2 (Apr. 2011), 27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. 109, 121

Chen, C.-S.; Hung, Y.-P.; and Cheng, J.-B., 1999. RANSAC-based DARCES: a new approach to fast automatic registration of partially overlapping range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21, 11 (1999), 1229–1234. 27

Chen, Y. and Medioni, G., 1992. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10, 3 (1992), 145–155. 23, 71, 72

Chetverikov, D.; Stepanov, D.; and Krsek, P., 2005. Robust Euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm. *Image and Vision Computing*, 23, 3 (2005), 299–309. 24, 75, 106, 128

Chew, L. P.; Goodrich, M. T.; Huttenlocher, D. P.; Kedem, K.; Kleinberg, J. M.; and Kravets, D., 1997. Geometric pattern matching under euclidean motion. *Computational Geometry*, 7, 1-2 (1997), 113–124. 77

Chin, T.-J.; Heng Kee, Y.; Eriksson, A.; and Neumann, F., 2016. Guaranteed outlier removal with mixed integer linear programs. In *Proceedings of the 2016 Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV, USA, Jun. 2016), 5858–5866. doi: 10.1109/CVPR.2016.631. 35, 166

Chui, H. and Rangarajan, A., 2000a. A feature registration framework using mixture models. In *Proceedings of the 2000 Workshop on Mathematical Methods in Biomedical Image Analysis* (Hilton Head Island, SC, USA, Jun. 2000), 190–197. IEEE. doi: 10.1109/MMBIA.2000.852377. 80, 102, 103, 126, 130

Chui, H. and Rangarajan, A., 2000b. A new algorithm for non-rigid point matching. In *Proceedings of the 2000 Conference on Computer Vision and Pattern Recognition*, vol. 2 (Hilton Head Island, SC, USA, Jun. 2000), 44–51. IEEE. 80, 102, 126

Chui, H. and Rangarajan, A., 2003. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89, 2 (2003), 114–141. Point-sets available at http://cise.ufl.edu/~anand/students/chui/rpm/TPS-RPM.zip. 9, 24, 106, 117, 128

Chum, O. and Matas, J., 2008. Optimal randomized RANSAC. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30, 8 (2008), 1472–1482. 20, 34, 166

Chung, D. H.; Yun, I. D.; and Lee, S. U., 1998. Registration of multiple-range views using the reverse-calibration technique. *Pattern Recognition*, 31, 4 (1998), 457–464. 28

Coleman, T. F. and Li, Y., 1996. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6, 2 (1996), 418–445. 114

Comaniciu, D., 2003. An algorithm for data-driven bandwidth selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25, 2 (2003), 281–288. 65, 111, 130

Cortes, C. and Vapnik, V., 1995. Support-vector networks. *Machine Learning*, 20, 3 (1995), 273–297. 108

Curless, B. and Levoy, M., 2014. Dragon, Stanford Computer Graphics Laboratory. Point-set available at http://graphics.stanford.edu/data/3Dscanrep/. 119, 148, 149

Darom, T. and Keller, Y., 2012. Scale-invariant features for 3-D mesh models. *IEEE Transactions on Image Processing*, 21, 5 (May 2012), 2758–2769. doi: 10.1109/TIP.2012.2183142. 18

David, P.; DeMenthon, D.; Duraiswami, R.; and Samet, H., 2002. SoftPOSIT: simultaneous pose and correspondence determination. In *Proceedings of the 2002 European Conference on Computer Vision* (Copenhagen, Denmark, May 2002), 698–714. Springer. 37, 38, 89

David, P.; Dementhon, D.; Duraiswami, R.; and Samet, H., 2004. SoftPOSIT: simultaneous pose and correspondence determination. *International Journal of Computer Vision*, 59, 3 (2004), 259–284. 8, 9, 39, 163, 165, 167, 180, 192

Dellaert, F.; Seitz, S. M.; Thorpe, C. E.; and Thrun, S., 2000. Structure from motion without correspondence. In *Proceedings of the 2000 Conference on Computer Vision and Pattern Recognition*, vol. 2 (Hilton Head Island, SC, USA, Jun. 2000), 557–564. IEEE. 36, 166

Dementhon, D. F. and Davis, L. S., 1995. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15, 1-2 (1995), 123–141. 38, 167

Dempster, A. P.; Laird, N. M.; and Rubin, D. B., 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39, 1 (Jan. 1977), 1–38. 24, 65, 111, 130, 149

Deselaers, T.; Heigold, G.; and Ney, H., 2010. Object classification by fusing SVMs and Gaussian mixtures. *Pattern Recognition*, 43, 7 (2010), 2476–2484. 65, 109, 115, 130

Detry, R.; Pugeault, N.; and Piater, J. H., 2009. A probabilistic framework for 3D visual object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31, 10 (2009), 1790–1803. 65, 111, 130

Devroye, L., 1987. *A course in density estimation*. Progress in Probability and Statistics. Birkhäuser Boston Inc. ISBN 9780817633653. 65, 81

Dhillon, I. S. and Modha, D. S., 2001. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42, 1 (2001), 143–175. 68

Dorai, C.; Weng, J.; and Jain, A. K., 1997. Optimal registration of object views using range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 10 (1997), 1131–1138. 28

Efrat, A.; Itai, A.; and Katz, M. J., 2001. Geometry helps in bottleneck matching and related problems. *Algorithmica*, 31, 1 (2001), 1–28. 77

Elbaz, G.; Avraham, T.; and Fischer, A., 2017. 3D point cloud registration for localization using a deep neural network auto-encoder. In *Proceedings of the 2017 Conference on Computer Vision and Pattern Recognition* (Honolulu, Hawaii, USA, Jul. 2017), 2472–2481. IEEE. 29

Engel, J.; Schöps, T.; and Cremers, D., 2014. LSD-SLAM: large-scale direct monocular SLAM. In *Proceedings of the 2014 European Conference on Computer Vision* (Zurich, Switzerland, Sep. 2014), 834–849. Springer. 35

Enqvist, O.; Ask, E.; Kahl, F.; and Åström, K., 2012. Robust fitting for multiple view geometry. In *Proceedings of the 2012 European Conference on Computer Vision* (Florence, Italy, Oct. 2012), 738–751. Springer. 20, 21, 34, 35, 166

Enqvist, O.; Ask, E.; Kahl, F.; and Åström, K., 2015. Tractable algorithms for robust model estimation. *International Journal of Computer Vision*, 112, 1 (2015), 115–129. 6, 20, 21, 34, 35, 166

Enqvist, O.; Josephson, K.; and Kahl, F., 2009. Optimal correspondences from pairwise constraints. In *Proceedings of the 2009 International Conference on Computer Vision* (Kyoto, Japan, Sep. 2009), 1295–1302. IEEE. 8, 20, 21

Enqvist, O. and Kahl, F., 2008. Robust optimal pose estimation. In *Proceedings of the 2008 European Conference on Computer Vision* (Marseille, France, Oct. 2008), 141–153. Springer. 34, 36, 166

Fan, R.-E.; Chen, P.-H.; and Lin, C.-J., 2005. Working set selection using second order information for training support vector machines. *The Journal of Machine Learning Research*, 6 (2005), 1889–1918. 109

Figueiredo, M. A. and Jain, A. K., 2002. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 24, 3 (Mar. 2002), 381–396. 151

Fischler, M. A. and Bolles, R. C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24, 6 (1981), 381–395. 6, 8, 9, 10, 20, 26, 34, 36, 39, 78, 89, 162, 163, 166, 192

Fisher, N. I., 1995. *Statistical Analysis of Circular Data*. Cambridge University Press. ISBN 9780521568906. 68

Fisher, R., 1953. Dispersion on a sphere. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 217, 295–305. Royal Society. doi: 10.1098/rspa.1953.0064. 67, 213

Fitzgibbon, A. W., 2003. Robust registration of 2D and 3D point sets. *Image and Vision Computing*, 21, 13 (Dec. 2003), 1145–1153. 8, 9, 22, 23, 24, 106, 128

Fredriksson, J.; Larsson, V.; Olsson, C.; and Kahl, F., 2016. Optimal relative pose with unknown correspondences. In *Proceedings of the 2016 Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV, USA, Jun. 2016), 1728–1736. IEEE. 36, 166

Frome, A.; Huber, D.; Kolluri, R.; Bülow, T.; and Malik, J., 2004. Recognizing objects in range data using regional point descriptors. In *Proceedings of the 2004 European Conference on Computer Vision* (Prague, Czech Republic, May 2004), 224–237. Springer. 17, 33

Gallego, G. and Yezzi, A., 2015. A compact formula for the derivative of a 3-D rotation in exponential coordinates. *Journal of Mathematical Imaging and Vision*, 51, 3 (2015), 378–384. 184

Gao, X.-S.; Hou, X.-R.; Tang, J.; and Cheng, H.-F., 2003. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25, 8 (Aug. 2003), 930–943. 33

Geiger, A.; Moosmann, F.; Car, O.; and Schuster, B., 2012. Automatic camera and range sensor calibration using a single shot. In *Proceedings of the 2012 International Conference on Robotics and Automation* (Saint Paul, MN, USA, May 2012), 3936–3943. IEEE. 6, 16, 103, 126

Gelfand, N.; Mitra, N. J.; Guibas, L. J.; and Pottmann, H., 2005. Robust global registration. In *Proceedings of the 2005 Eurographics Symposium on Geometry Processing*, vol. 255 (Vienna, Austria, Jul. 2005), 197–206. Eurographics Association. 6, 20, 21, 129

Glaunes, J.; Trouvé, A.; and Younes, L., 2004. Diffeomorphic matching of distributions: A new approach for unlabelled point-sets and sub-manifolds matching. In *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition*, vol. 2 (Washington, DC, USA, Jun. 2004), 712–718. IEEE. 25

Glocker, B.; Izadi, S.; Shotton, J.; and Criminisi, A., 2013. Real-time RGB-D camera relocalization. In *Proceedings of the 2013 International Symposium on Mixed*

and Augmented Reality (Adelaide, SA, Australia, Oct. 2013), 173–179. IEEE. doi: 10.1109/ISMAR.2013.6671777. 40

Gold, S. and Rangarajan, A., 1996. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18, 4 (Apr. 1996), 377–388. 38, 167

Gopal, S. and Yang, Y., 2014. Von Mises-Fisher clustering models. In *Proceedings of the 31st International Conference on Machine Learning*, vol. 32 of *Proceedings of Machine Learning Research* (Beijing, China, 22–24 Jun 2014), 154–162. PMLR. 68

Granger, S. and Pennec, X., 2002. Multi-scale EM-ICP: A fast and robust approach for surface registration. In *Proceedings of the 2002 European Conference on Computer Vision*, vol. 2353 (Copenhagen, Denmark, May 2002), 418–432. 24, 106

Greengard, L. and Strain, J., 1991. The fast Gauss transform. *SIAM Journal on Scientific and Statistical Computing*, 12, 1 (1991), 79–94. 113, 123, 225

Grimson, W. E. L., 1990. *Object Recognition by Computer: The Role of Geometric Constraints*. MIT Press, Cambridge, MA, USA. ISBN 0-262-07130-4. 10, 38, 39, 89, 163, 167

Grimson, W. E. L., 1991. The combinatorics of heuristic search termination for object recognition in cluttered environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13, 9 (Sep. 1991), 920–935. doi: 10.1109/34.93810. 39

Grunert, J. A., 1841. Das pothenotische problem in erweiterter gestalt nebst über seine anwendungen in der geodäsie. *Grunerts Archiv für Mathematik und Physik*, 1 (1841), 238–248. 33

Guennebaud, G.; Jacob, B.; et al., 2010. Eigen v3. http://eigen.tuxfamily.org. 193

Hamilton, W. R., 1844. LXXVIII. on quaternions; or on a new system of imaginaries in algebra. *Philosophical Magazine Series 3*, 25, 169 (1844), 489–495. 52

Hampel, F. R.; Ronchetti, E. M.; Rousseeuw, P. J.; and Stahel, W. A., 1986. *Robust statistics: the approach based on influence functions*. Wiley. ISBN 9780471829218. 80, 82

Hamza, A. B. and Krim, H., 2003. Jensen–Rényi divergence measure: Theoretical and computational perspectives. In *Proceedings of the 2003 International Symposium on Information Theory* (Yokohama, Japan, Jun. 2003), 257–257. IEEE. 80

Haralick, B. M.; Lee, C.-N.; Ottenberg, K.; and Nölle, M., 1994. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13, 3 (1994), 331–356. 33, 162, 166

Harris, C. and Stephens, M., 1988. A combined corner and edge detector. In *Proceedings of the Fourth Alvey Vision Conference*, vol. 15 (Manchester, UK, Aug. 1988), 147–151. 17, 33

Hartley, R. and Kahl, F., 2007. Optimal algorithms in multiview geometry. In *Proceedings of the 2007 Asian Conference on Computer Vision* (Tokyo, Japan, Nov. 2007), 13–34. Springer. 7

Hartley, R.; Trumpf, J.; Dai, Y.; and Li, H., 2013. Rotation averaging. *International Journal of Computer Vision*, 103, 3 (Jul. 2013), 267–305. 56, 57

Hartley, R. and Zisserman, A., 2003. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edn. ISBN 0521540518. 34, 51

Hartley, R. I. and Kahl, F., 2009. Global optimization through rotation space search. *International Journal of Computer Vision*, 82, 1 (Apr. 2009), 64–79. 6, 20, 22, 30, 34, 36, 42, 52, 55, 57, 97, 133, 134, 167, 171, 176

Heller, J.; Havlena, M.; and Pajdla, T., 2012. A branch-and-bound algorithm for globally optimal hand-eye calibration. In *Proceedings of the 2012 Conference on Computer Vision and Pattern Recognition* (Providence, RI, USA, Jun. 2012), 1608–1615. IEEE. 6

Hesch, J. A. and Roumeliotis, S. I., 2011. A direct least-squares (DLS) method for PnP. In *Proceedings of the 2011 International Conference on Computer Vision* (Barcelona, Spain, Nov. 2011), 383–390. IEEE. 33, 34, 162, 166

Horaud, R. and Dornaika, F., 1995. Hand-eye calibration. *The International Journal of Robotics Research*, 14, 3 (1995), 195–210. 6

Horaud, R.; Forbes, F.; Yguel, M.; Dewaele, G.; and Zhang, J., 2011. Rigid and articulated point registration with expectation conditional maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33, 3 (2011), 587–602. 24

Horn, B. K., 1987. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, 4, 4 (April 1987), 629–642. 8, 18, 19, 20, 22, 23, 72

Horn, B. K. P., 1984. Extended Gaussian images. *Proceedings of the IEEE*, 72, 12 (Dec. 1984), 1671–1686. doi: 10.1109/PROC.1984.13073. 28

Huber, D. F. and Hebert, M., 2003. Fully automatic registration of multiple 3D data sets. *Image and Vision Computing*, 21, 7 (2003), 637–650. 6, 16, 102, 126

Huber, P. J., 1981. *Robust Statistics*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons. ISBN 9780471418054. 74

Huttenlocher, D. P., 1991. Fast affine point matching: An output-sensitive method. In *Proceedings of the 1991 Conference on Computer Vision and Pattern Recognition* (Lahaina, Maui, Hawaii, USA, Jun. 1991), 263–268. IEEE. doi: 10.1109/CVPR.1991.139699. 27

Huttenlocher, D. P. and Ullman, S., 1990. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5, 2 (1990), 195–212. 6, 26, 37, 162

Irani, S. and Raghavan, P., 1999. Combinatorial and experimental results for randomized point matching algorithms. *Computational Geometry*, 12, 1-2 (1999), 17–31. 22, 26, 27

Jacobs, D. W., 1997. Matching 3-D models to 2-D images. *International Journal of Computer Vision*, 21, 1 (1997), 123–153. 37

Jian, B. and Vemuri, B. C., 2011. Robust point set registration using Gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33, 8 (2011), 1633–1645. 9, 11, 22, 24, 25, 65, 80, 102, 103, 104, 106, 107, 111, 112, 113, 117, 119, 126, 127, 128, 130, 149, 226

Joachims, T., 1999. Making large-scale support vector machine learning practical. In *Advances in Kernel Methods: Support Vector Learning* (Eds. B. Schölkopf; C. J. Burges; and A. J. Smola), 169–184. MIT Press, Cambridge, MA, USA. 109, 123, 225

Johnson, A. E. and Hebert, M., 1999. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21, 5 (1999), 433–449. 6, 16, 17, 19, 20, 33, 102, 126, 129

Jones, M. C.; Hjort, N. L.; Harris, I. R.; and Basu, A., 2001. A comparison of related density-based minimum divergence estimators. *Biometrika*, 88, 3 (2001), 865–873. 80, 82

Jurie, F., 1999. Solution of the simultaneous pose and correspondence problem using Gaussian error model. *Computer Vision and Image Understanding*, 73, 3 (1999), 357–373. 37, 41, 42, 167

Ke, Q. and Kanade, T., 2007. Quasiconvex optimization for robust geometric reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29, 10 (2007). 35

Kendall, A. and Cipolla, R., 2017. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the 2017 Conference on Computer Vision and Pattern Recognition* (Honolulu, Hawaii, USA, Jul. 2017), 6555–6564. doi: 10.1109/CVPR.2017.694. 40, 41

Kendall, A.; Grimes, M.; and Cipolla, R., 2015. PoseNet: A convolutional network for real-time 6-DOF camera relocalization. In *Proceedings of the 2015 International Conference on Computer Vision* (Santiago, Chile, Dec. 2015), 2938–2946. doi: 10.1109/ICCV.2015.336. 33, 40

Kent, J. T., 1982. The Fisher–Bingham distribution on the sphere. *Journal of the Royal Statistical Society. Series B (Methodological)*, (1982), 71–80. 67, 69, 213

Kiefer, J., 1953. Sequential minimax search for a maximum. *Proceedings of the American Mathematical Society*, 4, 3 (1953), 502–506. 184

Kim, V. G.; Lipman, Y.; and Funkhouser, T., 2011. Blended intrinsic maps. *ACM Transactions on Graphics*, 30, 4 (Jul. 2011), 79:1–79:12. doi: 10.1145/2010324.1964974. 18

Klein, G. and Murray, D., 2007. Parallel tracking and mapping for small ar workspaces. In *Proceedings of the 2007 International Symposium on Mixed and Augmented Reality* (Nara, Japan, Nov. 2007), 225–234. IEEE. 35

Kneip, L. and Furgale, P., 2014. OpenGV: A unified and generalized approach to real-time calibrated geometric vision. In *Proceedings of the 2014 International Conference on Robotics and Automation* (Hong Kong, China, Jun. 2014), 1–8. IEEE. 34, 180, 192, 193, 198

Kneip, L.; Li, H.; and Seo, Y., 2014. UPnP: An optimal O(n) solution to the absolute pose problem with universal applicability. In *Proceedings of the 2014 European Conference on Computer Vision* (Zurich, Switzerland, Sep. 2014), 127–142. Springer. 33, 34

Kneip, L.; Scaramuzza, D.; and Siegwart, R., 2011. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Proceedings of the 2011 Conference on Computer Vision and Pattern Recognition* (Colorado Springs, CO, USA, Jun. 2011), 2969–2976. IEEE. 33, 34, 162, 166, 192

Kneip, L.; Yi, Z.; and Li, H., 2015. SDICP: Semi-dense tracking based on iterative closest points. In *Proceedings of the 2015 British Machine Vision Conference* (Swansea, UK, Sep. 2015), 100.1–100.12. BMVA Press. doi: 10.5244/C.29.100. 6, 162

Kokkinos, I.; Bronstein, M. M.; Litman, R.; and Bronstein, A. M., 2012. Intrinsic shape context descriptors for deformable shapes. In *Proceedings of the 2012 Conference on Computer Vision and Pattern Recognition* (Providence, RI, USA, Jun. 2012), 159–166. IEEE. 18

Kullback, S. and Leibler, R. A., 1951. On information and sufficiency. *The Annals of Mathematical Statistics*, 22, 1 (1951), 79–86. 80

Lamdan, Y. and Wolfson, H. J., 1988. Geometric hashing: A general and efficient model-based recognition scheme. In *Proceedings of the 1988 International Conference on Computer Vision* (Tampa, Florida, USA, Dec. 1988), 238–249. IEEE. doi: 10.1109/CCV.1988.589995. 40

Land, A. H. and Doig, A. G., 1960. An automatic method of solving discrete programming problems. *Econometrica: Journal of the Econometric Society*, (1960), 497–520. 9, 29, 41, 90, 131, 167, 169

Lawler, E. L. and Wood, D. E., 1966. Branch-and-bound methods: A survey. *Operations Research*, 14, 4 (1966), 699–719. 90

Leonard, J. J. and Durrant-Whyte, H. F., 1991. Simultaneous map building and localization for an autonomous mobile robot. In *Proceedings of the 1991 International Workshop on Intelligent Robots and Systems* (Osaka, Japan, Nov. 1991), 1442–1447. IEEE. doi: 10.1109/IROS.1991.174711. 6

Lepetit, V.; Moreno-Noguer, F.; and Fua, P., 2009. EPnP: An accurate O(n) solution to the PnP problem. *International Journal of Computer Vision*, 81, 2 (2009), 155–166. 8, 33, 34, 162, 166, 198

Levenberg, K., 1944. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2, 2 (1944), 164–168. 34

Li, H., 2007. A practical algorithm for L_∞ triangulation with outliers. In *Proceedings of the 2007 Conference on Computer Vision and Pattern Recognition* (Minneapolis, MN, USA, Jun. 2007), 1–8. IEEE. 35

Li, H., 2009. Consensus set maximization with guaranteed global optimality for robust geometry estimation. In *Proceedings of the 2009 International Conference on Computer Vision* (Kyoto, Japan, Sep. 2009), 1074–1080. IEEE. 21, 34, 78, 166

Li, H. and Hartley, R., 2007. The 3D-3D registration problem revisited. *Proceedings of the 2007 International Conference on Computer Vision*, (Oct. 2007), 1–8. 8, 10, 30, 42, 52, 57, 73, 92, 106, 129, 132, 167, 170

Li, Y.; Snavely, N.; Huttenlocher, D.; and Fua, P., 2012. Worldwide pose estimation using 3D point clouds. In *Proceedings of the 2012 European Conference on Computer Vision* (Florence, Italy, Oct. 2012), 15–29. Springer-Verlag. 35, 166

Li, Y.; Snavely, N.; and Huttenlocher, D. P., 2010. Location recognition using prioritized feature matching. In *Proceedings of the 2010 European Conference on Computer Vision* (Crete, Greece, Sep. 2010), 791–804. Springer. 35, 166

Lin, J., 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37, 1 (1991), 145–151. 80

Lin, W.-Y.; Cheong, L.-F.; Tan, P.; Dong, G.; and Liu, S., 2012. Simultaneous camera pose and correspondence estimation with motion coherence. *International Journal of Computer Vision*, 96, 2 (2012), 145–161. doi: 10.1007/s11263-011-0456-9. 36, 166

Litman, R. and Bronstein, A. M., 2014. Learning spectral descriptors for deformable shape correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36, 1 (2014), 171–180. 18

Litman, R.; Bronstein, A. M.; and Bronstein, M. M., 2011. Diffusion-geometric maximally stable component detection in deformable shapes. *Computers and Graphics*, 35, 3 (2011), 549–560. 17

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60, 2 (2004), 91–110. 17, 33, 163

Maes, C.; Fabry, T.; Keustermans, J.; Smeets, D.; Suetens, P.; and Vandermeulen, D., 2010. Feature detection on 3D face surfaces for pose normalisation and recognition. In *Proceedings of the 2010 International Conference on Biometrics:*

Theory Applications and Systems (Washington, DC, USA, Sep. 2010), 1–6. IEEE. 17, 33

Magnusson, M., 2011. AASS-Loop, Robotic 3D Scan Repository. Point-set available at http://kos.informatik.uni-osnabrueck.de/3Dscans/aass-loop.zip. Örebro University. 119

Magnusson, M.; Lilienthal, A.; and Duckett, T., 2007. Scan registration for autonomous mining vehicles using 3D-NDT. *Journal of Field Robotics*, 24, 10 (2007), 803–827. 6, 25, 66, 80, 106, 119, 128, 130

Magnusson, M.; Nuchter, A.; Lorken, C.; Lilienthal, A. J.; and Hertzberg, J., 2009. Evaluation of 3D registration reliability and speed – a comparison of ICP and NDT. In *Proceedings of the 2009 International Conference on Robotics and Automation* (Kyoto, Japan, Sep. 2009), 3907–3912. IEEE. 25

Makadia, A.; Geyer, C.; and Daniilidis, K., 2007. Correspondence-free structure from motion. *International Journal of Computer Vision*, 75, 3 (2007), 311–327. 36, 166

Makadia, A.; Patterson, A.; and Daniilidis, K., 2006. Fully automatic registration of 3D point clouds. In *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition*, vol. 1 (New York, NY, USA, Jun. 2006), 1297–1304. IEEE. 28

Makela, T.; Clarysse, P.; Sipila, O.; Pauna, N.; Pham, Q. C.; Katila, T.; and Magnin, I. E., 2002. A review of cardiac image registration methods. *IEEE Transactions on Medical Imaging*, 21, 9 (2002), 1011–1021. 6, 16, 103, 126

Marchand, E.; Uchiyama, H.; and Spindler, F., 2016. Pose estimation for augmented reality: a hands-on survey. *IEEE Transactions on Visualization and Computer Graphics*, 22, 12 (2016), 2633–2651. 6, 162

Mardia, K., 1972. *Statistics of Directional Data*. Probability and Mathematical Statistics. Academic Press. ISBN 9780124711501. 67, 69, 213

Masci, J.; Boscaini, D.; Bronstein, M.; and Vandergheynst, P., 2015. Geodesic convolutional neural networks on Riemannian manifolds. In *Proceedings of the 2015 International Conference on Computer Vision Workshop* (Santiago, Chile, Dec. 2015), 832–840. 18

Matas, J.; Chum, O.; Urban, M.; and Pajdla, T., 2004. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22, 10 (2004), 761–767. 33

Mellado, N.; Aiger, D.; and Mitra, N. J., 2014. Super 4PCS fast global pointcloud registration via smart indexing. *Computer Graphics Forum*, 33, 5 (2014), 205–215. doi: 10.1111/cgf.12446. 27, 90, 106, 119, 123, 129

Mezzadri, F., 2007. How to generate random matrices from the classical compact groups. *Notices of the American Mathematical Society*, 54, 5 (2007), 592–604. 194

Minguez, J.; Lamiraux, F.; and Montesano, L., 2005. Metric-based scan matching algorithms for mobile robot displacement estimation. In *Proceedings of the 2005 International Conference on Robotics and Automation*, vol. 4 (Barcelona, Spain, Apr. 2005), 35–57. IEEE. 24

Moré, J., 1978. The Levenberg-Marquardt algorithm: Implementation and theory. *Numerical Analysis*, (1978), 105–116. 9, 24, 106, 128

Moreno-Noguer, F.; Lepetit, V.; and Fua, P., 2008. Pose priors for simultaneously solving alignment and correspondence. In *Proceedings of the 2008 European Conference on Computer Vision* (Marseille, France, Oct. 2008), 405–418. Springer. 9, 37, 39, 89, 163, 165, 167, 192, 193, 194

Mount, D. M.; Netanyahu, N. S.; and Le Moigne, J., 1999. Efficient algorithms for robust feature matching. *Pattern Recognition*, 32, 1 (1999), 17–38. 29, 30

Mundy, J. L., 2006. Object recognition in the geometric era: A retrospective. In *Toward Category-Level Object Recognition* (Eds. J. Ponce; M. Hebert; C. Schmid; and A. Zisserman), vol. 4170 of *Lecture Notes in Computer Science*, 3–28. Springer, Berlin, Heidelberg. ISBN 978-3-540-68795-5. doi: 10.1007/11957959_1. 6, 162

Mur-Artal, R.; Montiel, J. M. M.; and Tardos, J. D., 2015. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31, 5 (2015), 1147–1163. 35

Myronenko, A. and Song, X., 2010. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32, 12 (2010), 2262–2275. 9, 22, 24, 80, 102, 106, 117, 119, 126, 128, 147, 152

Namin, S. T.; Najafi, M.; Salzmann, M.; and Petersson, L., 2015. A multi-modal graphical model for scene analysis. In *Proceedings of the 2015 Winter Conference on Applications Computer Vision* (Waikoloa, HI, USA, Jan. 2015), 1006–1013. IEEE. 198

Newcombe, R. A.; Lovegrove, S. J.; and Davison, A. J., 2011. DTAM: Dense Tracking And Mapping in real-time. In *Proceedings of the 2011 International Conference on Computer Vision* (Barcelona, Spain, Nov. 2011), 2320–2327. IEEE. 35

Nöll, T.; Pagani, A.; and Stricker, D., 2011. Markerless Camera Pose Estimation – An Overview. In *Visualization of Large and Unstructured Data Sets - Applications in Geospatial Planning, Modeling and Engineering*, vol. 19 of *OpenAccess Series in Informatics (OASIcs)*, 45–54. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany. 6, 162

Nüchter, A.; Lingemann, K.; Hertzberg, J.; and Surmann, H., 2007. 6D SLAM–3D mapping outdoor environments. *Journal of Field Robotics*, 24, 8-9 (2007), 699–722. 6, 16, 23, 103, 126

Olson, C. F., 1997. Efficient pose clustering using a randomized algorithm. *International Journal of Computer Vision*, 23, 2 (1997), 131–147. 27, 39

Olson, C. F., 2001. A general method for geometric feature matching and model extraction. *International Journal of Computer Vision*, 45, 1 (2001), 39–54. 6, 37, 162

Olsson, C.; Enqvist, O.; and Kahl, F., 2008. A polynomial-time bound for matching and registration with outliers. In *Proceedings of the 2008 Conference on Computer Vision and Pattern Recognition* (Anchorage, Alaska, USA, Jun. 2008), 1–8. IEEE. 20, 21, 35

Olsson, C.; Eriksson, A.; and Hartley, R., 2010. Outlier removal using duality. In *Proceedings of the 2010 Conference on Computer Vision and Pattern Recognition* (San Francisco, CA, USA, Jun. 2010), 1450–1457. IEEE. 35, 166

Olsson, C.; Kahl, F.; and Oskarsson, M., 2006. Optimal estimation of perspective camera pose. In *Proceedings of the 2006 International Conference on Pattern Recognition*, vol. 2 (Hong Kong, Aug. 2006), 5–8. IEEE. 30, 33, 34, 42

Olsson, C.; Kahl, F.; and Oskarsson, M., 2009. Branch-and-bound methods for Euclidean registration problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31, 5 (2009), 783–794. 30, 42, 129, 167

Papadimitriou, C. H. and Steiglitz, K., 1982. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA. ISBN 0-13-152462-3. 73

Papazov, C. and Burschka, D., 2011. Stochastic global optimization for robust point set registration. *Computer Vision and Image Understanding*, 115, 12 (2011), 1598–1609. 28, 129

Parra Bustos, A. J. and Chin, T.-J., 2015. Guaranteed outlier removal for rotation search. In *Proceedings of the 2015 IEEE International Conference on Computer Vision* (Santiago, Chile, Dec. 2015), 2165–2173. IEEE. 35

Parra Bustos, A. J.; Chin, T.-J.; Eriksson, A.; Li, H.; and Suter, D., 2016. Fast rotation search with stereographic projections for 3D registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38, 11 (Nov. 2016), 2227–2240. doi: 10.1109/TPAMI.2016.2517636. 6, 8, 10, 31, 76, 223

Parra Bustos, A. J.; Chin, T.-J.; and Suter, D., 2014. Fast rotation search with stereographic projections for 3D registration. In *Proceedings of the 2014 Conference on Computer Vision and Pattern Recognition* (Columbus, OH, USA, Jun. 2014), 3930–3937. IEEE. 30, 31, 129

Paudel, D. P.; Habed, A.; Demonceaux, C.; and Vasseur, P., 2015a. LMI-based 2D-3D registration: From uncalibrated images to Euclidean scene. In *Proceedings of the 2015 Conference on Computer Vision and Pattern Recognition* (Boston, Massachusetts, USA, Jun. 2015), 4494–4502. doi: 10.1109/CVPR.2015.7299079. 37, 167

Paudel, D. P.; Habed, A.; Demonceaux, C.; and Vasseur, P., 2015b. Robust and optimal sum-of-squares-based point-to-plane registration of image sets and structured scenes. In *Proceedings of the 2015 International Conference on Computer Vision* (Santiago, Chile, Dec. 2015), 2048–2056. 37, 167

Penate-Sanchez, A.; Andrade-Cetto, J.; and Moreno-Noguer, F., 2013. Exhaustive linearization for robust camera pose and focal length estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35, 10 (2013), 2387–2400. 33, 34

Petersen, K. B. and Pedersen, M. S., 2012. The matrix cookbook. URL http://www2.imm.dtu.dk/pubdb/p.php?3274. 67

Pfeuffer, F.; Stiglmayr, M.; and Klamroth, K., 2012. Discrete and geometric branch and bound algorithms for medical image registration. *Annals of Operations Research*, 196, 1 (2012), 737–765. 30

Platt, J. C., 1999. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods: Support Vector Learning* (Eds. B. Schölkopf; C. J. Burges; and A. J. Smola), 185–208. MIT Press, Cambridge, MA, USA. 109

Pomerleau, F.; Colas, F.; Siegwart, R.; and Magnenat, S., 2013. Comparing ICP variants on real-world data sets. *Autonomous Robots*, 34, 3 (2013), 133–148. 6, 16, 23, 103, 106, 126, 128

Pomerleau, F.; Liu, M.; Colas, F.; and Siegwart, R., 2012. Challenging data sets for point cloud registration algorithms. *The International Journal of Robotics Research*, 31, 14 (Dec. 2012), 1705–1711. 151, 153, 155

Pons-Moll, G.; Taylor, J.; Shotton, J.; Hertzmann, A.; and Fitzgibbon, A., 2015. Metric regression forests for correspondence estimation. *International Journal of Computer Vision*, 113, 3 (2015), 163–175. 18

Pukkila, T. M. and Rao, C. R., 1988. Pattern recognition based on scale invariant discriminant functions. *Information Sciences*, 45, 3 (1988), 379–389. 69

Quan, L. and Lan, Z., 1999. Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21, 8 (1999), 774–780. 34

Raposo, C. and Barreto, J. P., 2017. Using 2 point+normal sets for fast registration of point clouds with small overlap. In *Proceedings of the 2017 International Conference on Robotics and Automation* (Singapore, May 2017), 5652–5658. 27, 90

Remondino, F., 2011. Heritage recording and 3D modeling with photogrammetry and 3D scanning. *Remote Sensing*, 3, 6 (2011), 1104–1138. doi: 10.3390/rs3061104. 6

Robertson, C. and Fisher, R. B., 2002. Parallel evolutionary registration of range data. *Computer Vision and Image Understanding*, 87, 1 (2002), 39–50. 28, 129

Rodolà, E.; Rota Bulo, S.; Windheuser, T.; Vestner, M.; and Cremers, D., 2014. Dense non-rigid shape correspondence using random forests. In *Proceedings of the 2014 Conference on Computer Vision and Pattern Recognition* (Columbus, OH, USA, Jun. 2014), 4177–4184. IEEE. 18

Ruland, T.; Pajdla, T.; and Krüger, L., 2012. Globally optimal hand-eye calibration. In *Proceedings of the 2012 Conference on Computer Vision and Pattern Recognition* (Providence, RI, USA, Jun. 2012), 1035–1042. IEEE. 6

Rusinkiewicz, S. and Levoy, M., 2001. Efficient variants of the ICP algorithm. In *Proceedings of the 2001 International Conference on 3D Digital Imaging and Modeling* (Quebec City, Canada, May 2001), 145–152. IEEE. 23, 74, 106, 128

Rusu, R. B.; Blodow, N.; and Beetz, M., 2009. Fast Point Feature Histograms (FPFH) for 3D registration. In *Proceedings of the 2009 International Conference on Robotics and Automation* (Kyoto, Japan, Sep. 2009), 3212–3217. IEEE. 17, 21, 33, 89, 106, 129

Rusu, R. B.; Blodow, N.; Marton, Z. C.; and Beetz, M., 2008a. Aligning point cloud views using persistent feature histograms. In *Proceedings of the 2008 International Conference on Intelligent Robots and Systems* (Nice, France, Sep. 2008), 3384–3391. IEEE. 19, 20, 21

Rusu, R. B. and Cousins, S., 2011. 3D is here: Point Cloud Library (PCL). In *Proceedings of the 2011 International Conference on Robotics and Automation* (Shanghai, China, May 2011). IEEE. 62

Rusu, R. B.; Marton, Z. C.; Blodow, N.; and Beetz, M., 2008b. Learning informative point classes for the acquisition of object model maps. In *Proceedings of the 2008 International Conference on Control, Automation, Robotics and Vision* (Hanoi, Vietnam, Dec. 2008). IEEE. 17, 33

Sanderson, C. and Curtin, R., 2016. Armadillo: a template-based C++ library for linear algebra. *Journal of Open Source Software*, 1 (2016). 193

Sandhu, R.; Dambreville, S.; and Tannenbaum, A., 2010. Point set registration via particle filtering and stochastic dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32, 8 (2010), 1459–1473. 28, 80, 89, 106, 129

Sattler, T.; Leibe, B.; and Kobbelt, L., 2011. Fast image-based localization using direct 2D-to-3D matching. In *Proceedings of the 2011 International Conference on Computer Vision* (Barcelona, Spain, Nov. 2011), 667–674. IEEE. 35, 166

Sattler, T.; Leibe, B.; and Kobbelt, L., 2012. Improving image-based localization by active correspondence search. In *Proceedings of the 2012 European Conference on Computer Vision* (Florence, Italy, Oct. 2012), 752–765. Springer-Verlag. 35, 166

Sattler, T.; Leibe, B.; and Kobbelt, L., 2017. Efficient effective prioritized matching for large-scale image-based localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 9 (Sep. 2017), 1744–1756. doi: 10.1109/TPAMI.2016.2611662. 8, 33, 35

Schmidt, J. and Niemann, H., 2001. Using quaternions for parametrizing 3-D rotations in unconstrained nonlinear optimization. In *Proceedings of the Vision Modeling and Visualization Conference*, vol. 1 (Stuttgart, Germany, Nov. 2001), 399–406. IEEE. 87, 131

Schölkopf, B.; Platt, J. C.; Shawe-Taylor, J.; Smola, A. J.; and Williamson, R. C., 2001. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13, 7 (2001), 1443–1471. 108, 109

Schölkopf, B.; Sung, K.-K.; Burges, C. J. C.; Girosi, F.; Niyogi, P.; Poggio, T.; and Vapnik, V., 1997. Comparing support vector machines with Gaussian kernels to radial basis function classifiers. *IEEE Transactions on Signal Processing*, 45, 11 (1997), 2758–2765. 110

Scott, D. W., 2001. Parametric statistical modeling by minimum integrated square error. *Technometrics*, 43, 3 (2001), 274–285. 81, 82, 112, 130

Scott, D. W. and Szewczyk, W. F., 2001. From kernels to mixtures. *Technometrics*, 43, 3 (2001), 323–335. 80, 111

Seo, Y.; Choi, Y.-J.; and Lee, S. W., 2009. A branch-and-bound algorithm for globally optimal calibration of a camera-and-rotation-sensor system. In *Proceedings of the 2009 International Conference on Computer Vision* (Kyoto, Japan, Sep. 2009), 1173–1178. IEEE. 6

Shotton, J.; Glocker, B.; Zach, C.; Izadi, S.; Criminisi, A.; and Fitzgibbon, A., 2013. Scene coordinate regression forests for camera relocalization in RGB-D images. In *Proceedings of the 2013 Conference on Computer Vision and Pattern Recognition* (Portland, OR, USA, Jun. 2013), 2930–2937. IEEE. doi: 10.1109/CVPR.2013.377. 29, 33, 40

Silva, L.; Bellon, O. R. P.; and Boyer, K. L., 2005. Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27, 5 (2005), 762–776. 28, 89, 106, 129

Sim, K. and Hartley, R., 2006. Removing outliers using the L_∞ norm. In *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition*, vol. 1 (New York, NY, USA, Jun. 2006), 485–494. IEEE. 35, 166

Sipiran, I. and Bustos, B., 2010. A robust 3D interest points detector based on Harris operator. In *Proceedings of the 2010 Eurographics Conference on 3D Object Retrieval* (Florence, Italy, Oct. 2010), 7–14. Eurographics Association. 17, 33

Smith, R. C. and Cheeseman, P., 1986. On the representation and estimation of spatial uncertainty. *The International Journal of Robotics Research*, 5, 4 (1986), 56–68. 6

Steder, B.; Rusu, R. B.; Konolige, K.; and Burgard, W., 2011. Point feature extraction on 3D range scans taking into account object boundaries. In *Proceedings of the 2011 International Conference on Robotics and Automation* (Shanghai, China, May 2011), 2601–2608. IEEE. doi: 10.1109/ICRA.2011.5980187. 17, 33

Steinke, F.; Schölkopf, B.; and Blanz, V., 2005. Support Vector Machines for 3D shape processing. *Computer Graphics Forum*, 24, 3 (2005), 285–294. 108, 122

Stewart, C. V., 1999. Robust parameter estimation in computer vision. *SIAM review*, 41, 3 (1999), 513–537. 80, 82

Stockman, G., 1987. Object recognition and localization via pose clustering. *Computer Vision, Graphics, and Image Processing*, 40, 3 (1987), 361–387. 27, 39

Stoyanov, T. D.; Magnusson, M.; Andreasson, H.; and Lilienthal, A., 2012. Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations. *The International Journal of Robotics Research*, (2012). 25, 106, 119, 121, 124, 128

Straub, J.; Campbell, T.; How, J. P.; and Fisher, J. W., 2015. Small-variance nonparametric clustering on the hypersphere. In *Proceedings of the 2015 Conference on Computer Vision and Pattern Recognition* (Boston, Massachusetts, USA, Jun. 2015), 334–342. IEEE. 68

Straub, J.; Campbell, T.; How, J. P.; and Fisher III, J. W., 2017. Efficient global point cloud alignment using Bayesian nonparametric mixtures. In *Proceedings of the 2017 Conference on Computer Vision and Pattern Recognition* (Honolulu, Hawaii, USA, Jul. 2017), 2403–2412. IEEE. doi: 10.1109/CVPR.2017.258. 8, 31, 52, 65, 85, 92, 129, 130, 214

Sun, B.; Kong, W.; Zhang, L.; and Zhang, J., 2014. Fourier analysis techniques applied in data registration: A survey. In *Proceedings of the 2014 International Conference on Multisensor Fusion and Information Integration for Intelligent Systems* (Beijing, China, Sep. 2014), 1–5. IEEE. doi: 10.1109/MFI.2014.6997740. 28

Sun, J.; Ovsjanikov, M.; and Guibas, L., 2009. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, 28, 5 (2009), 1383–1392. doi: 10.1111/j.1467-8659.2009.01515.x. 17

Sutherland, I., 1963. Sketchpad, a man-machine graphical communication system. Technical report, Massachusetts Institute of Technology. 34

Svärm, L.; Enqvist, O.; Kahl, F.; and Oskarsson, M., 2016. City-scale localization for cameras with known vertical direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 7 (2016), 1455–1461. 6, 8, 34, 35, 166

Svärm, L.; Enqvist, O.; Oskarsson, M.; and Kahl, F., 2014. Accurate localization and pose estimation for large 3D models. In *Proceedings of the 2014 Conference on Computer Vision and Pattern Recognition* (Columbus, OH, USA, Jun. 2014), 532–539. IEEE. 6, 34, 35, 166

Tam, G. K.; Cheng, Z.-Q.; Lai, Y.-K.; Langbein, F. C.; Liu, Y.; Marshall, D.; Martin, R. R.; Sun, X.-F.; and Rosin, P. L., 2013. Registration of 3D point clouds and meshes: A survey from rigid to nonrigid. *IEEE Transactions on Visualization and Computer Graphics*, 19, 7 (2013), 1199–1217. 23, 106, 128

Taylor, J.; Shotton, J.; Sharp, T.; and Fitzgibbon, A., 2012. The Vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *Proceedings of the 2012 Conference on Computer Vision and Pattern Recognition* (Providence, RI, USA, Jun. 2012), 103–110. IEEE. 18

Tombari, F.; Salti, S.; and Di Stefano, L., 2010. Unique signatures of histograms for local surface description. In *Proceedings of the 2010 European Conference on Computer Vision* (Crete, Greece, Sep. 2010), 356–369. Springer. 17, 33

Tombari, F.; Salti, S.; and Di Stefano, L., 2013. Performance evaluation of 3D keypoint detectors. *International Journal of Computer Vision*, 102, 1–3 (Mar. 2013), 198–220. doi: 10.1007/s11263-012-0545-4. 17, 19, 33

Torr, P. H. and Zisserman, A., 2000. MLESAC: a new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78, 1 (2000), 138–156. 20

Tsang, I. W.; Kwok, J. T.; and Cheung, P.-M., 2005. Core vector machines: Fast SVM training on very large data sets. *Journal of Machine Learning Research*, 6 (2005), 363–392. 109, 123, 225

Tsin, Y. and Kanade, T., 2004. A correlation-based approach to robust point set registration. In *Proceedings of the 2004 European Conference on Computer Vision* (Prague, Czech Republic, May 2004), 558–569. Springer. Point-set available at http://www.cs.cmu.edu/~ytsin/KCReg/KCReg.zip. 9, 24, 25, 80, 102, 103, 106, 117, 126, 128, 130

Turk, G. and Levoy, M., 2014. Stanford Bunny, Stanford Computer Graphics Laboratory. http://graphics.stanford.edu/data/3Dscanrep/. 148

Van Erven, T. and Harremos, P., 2014. Rényi divergence and kullback–leibler divergence. *IEEE Transactions on Information Theory*, 60, 7 (2014), 3797–3820. doi: 10.1109/TIT.2014.2320500. 80

Van Kaick, O.; Zhang, H.; Hamarneh, G.; and Cohen-Or, D., 2011. A survey on shape correspondence. *Computer Graphics Forum*, 30, 6 (2011), 1681–1707. 18, 22, 46

Van Nguyen, H. and Porikli, F., 2013. Support Vector Shape: A classifier-based shape representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35, 4 (2013), 970–982. 104, 107, 108

Vestner, M.; Litman, R.; Rodolà, E.; Bronstein, A.; and Cremers, D., 2017. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In *Proceedings of the 2017 Conference on Computer Vision and Pattern Recognition* (Honolulu, Hawaii, USA, Jul. 2017), 6681–6690. doi: 10.1109/CVPR.2017.707. 18

VXL, 2014. VXL 1.14.0: C++ Libraries for Computer Vision. http://vxl.sourceforge.net/. 148

Wachowiak, M. P.; Smolíková, R.; Zheng, Y.; Zurada, J. M.; and Elmaghraby, A. S., 2004. An approach to multimodal biomedical image registration utilizing particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8, 3 (2004), 289–301. 28, 129

Wand, M. P. and Jones, M. C., 1995. *Kernel Smoothing*, vol. 60 of *Monographs on Statistics and Applied Probability*. Chapman & Hill / CRC Press. 111

Wang, F. and Gelfand, A. E., 2013. Directional data analysis under the general projected normal distribution. *Statistical Methodology*, 10, 1 (2013), 113–127. 69, 213

Wang, F.; Syeda-Mahmood, T.; Vemuri, B. C.; Beymer, D.; and Rangarajan, A., 2009. Closed-form Jensen–Renyi divergence for mixture of Gaussians and applications to group-wise shape registration. In *Proceedings of the 2009 International Conference on Medical Image Computing and Computer-Assisted Intervention* (London, UK, Sep. 2009), 648–655. Springer. 80

Wang, F.; Vemuri, B. C.; Rangarajan, A.; and Eisenschenk, S. J., 2008. Simultaneous nonrigid registration of multiple point sets and atlas construction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30, 11 (Nov. 2008), 2011–2022. doi: 10.1109/TPAMI.2007.70829. 80

Watson, G., 1983. *Statistics on Spheres*, vol. 6 of *University of Arkansas Lecture Notes in the Mathematical Sciences*. Wiley. ISBN 9780471888666. 67, 69

Wilks, S. S., 1932. Certain generalizations in the analysis of variance. *Biometrika*, 24, 3-4 (1932), 471–494. 109

Windham, M. P., 1995. Robustifying model fitting. *Journal of the Royal Statistical Society. Series B (Methodological)*, (1995), 599–609. 80, 82

Windheuser, T.; Vestner, M.; Rodola, E.; Triebel, R.; and Cremers, D., 2014. Optimal intrinsic descriptors for non-rigid shape analysis. In *Proceedings of the 2014 British Machine Vision Conference*. BMVA Press. doi: 10.5244/C.28.44. 18

Wolfson, H. J. and Rigoutsos, I., 1997. Geometric hashing: An overview. *IEEE Computational Science and Engineering*, 4, 4 (1997), 10–21. 27

Woodford, O. J.; Pham, M.-T.; Maki, A.; Perbet, F.; and Stenger, B., 2014. Demisting the Hough transform for 3D shape recognition and registration. *International Journal of Computer Vision*, 106, 3 (2014), 332–341. 19, 20, 129

Wulf, O., 2011. Hannover2, Robotic 3D Scan Repository. Point-set available at http://kos.informatik.uni-osnabrueck.de/3Dscans/hannover2.tgz. Leibniz University. 119

Wunsch, P. and Hirzinger, G., 1996. Registration of CAD-models to images by iterative inverse perspective matching. In *Proceedings of the 1996 International Conference on Pattern Recognition*, vol. 1 (Vienna, Austria, Aug. 1996), 78–83. IEEE. 37

Xiong, H.; Szedmak, S.; and Piater, J., 2013a. A study of point cloud registration with probability product kernel functions. In *Proceedings of the 2013 International Conference on 3D Vision* (Seattle, Washington, USA, Jul. 2013), 207–214. IEEE. 65, 111

Xiong, H.; Szedmark, S.; and Piater, J., 2013b. Efficient, general point cloud registration with kernel feature maps. In *Proceedings of the 2013 International Conference on Computer and Robot Vision* (Regina, Saskatchewan, Canada, May 2013), 83–90. IEEE. 28

Yang, C.; Duraiswami, R.; Gumerov, N. A.; and Davis, L., 2003. Improved fast Gauss transform and efficient kernel density estimation. In *Proceedings of the 2003 International Conference on Computer Vision* (Nice, France, 2003), 664–671. IEEE. 123, 124, 225

Yang, J.; Dai, Y.; Li, H.; Gardner, H.; and Jia, Y., 2013a. Single-shot extrinsic calibration of a generically configured RGB-D camera rig from scene constraints. In *Proceedings of the 2013 International Symposium on Mixed and Augmented Reality* (Adelaide, Australia, Oct. 2013), 181–188. 6, 16, 103, 126

Yang, J.; Li, H.; Campbell, D.; and Jia, Y., 2016. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38, 11 (Nov. 2016), 2241–2254. doi: 10.1109/TPAMI.2015.2513405. 8, 10, 22, 31, 42, 43, 72, 75, 97, 106, 127, 129, 135, 141, 142, 147, 158, 164, 167, 179, 180, 223

Yang, J.; Li, H.; and Jia, Y., 2013b. GoICP: Solving 3D registration efficiently and globally optimally. In *Proceedings of the 2013 International Conference on Computer Vision* (Sydney, Australia, Dec. 2013), 1457–1464. IEEE. 31, 42, 119, 123, 127, 129, 151, 152

Yershova, A.; Jain, S.; Lavalle, S. M.; and Mitchell, J. C., 2010. Generating uniform incremental grids on SO(3) using the Hopf fibration. *The International Journal of Robotics Research*, 29, 7 (Jun. 2010), 801–812. doi: 10.1177/0278364909352700. 147

Yu, J.; Eriksson, A.; Chin, T.-J.; and Suter, D., 2011. An adversarial optimization approach to efficient outlier removal. In *Proceedings of the 2011 International Conference on Computer Vision* (Barcelona, Spain, Nov. 2011), 399–406. IEEE. 35, 166

Yuille, A. L. and Grzywacz, N. M., 1989. A mathematical analysis of the motion coherence theory. *International Journal of Computer Vision*, 3, 2 (1989), 155–175. 24

Zaharescu, A.; Boyer, E.; Varanasi, K.; and Horaud, R., 2009. Surface feature detection and description with applications to mesh matching. In *Proceedings of the 2009 Conference on Computer Vision and Pattern Recognition* (Miami, FL, USA, Jun. 2009), 373–380. IEEE. 17

Zeisl, B.; Sattler, T.; and Pollefeys, M., 2015. Camera pose voting for large-scale image-based localization. In *Proceedings of the 2015 International Conference on Computer Vision* (Santiago, Chile, Dec. 2015), 2704–2712. IEEE. 35

Zhang, Z., 1994. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13, 2 (1994), 119–152. 23, 24, 102, 103, 106, 126, 128

Zhao, W.; Nister, D.; and Hsu, S., 2005. Alignment of continuous video onto 3D point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27, 8 (2005), 1305–1318. 6, 16, 103, 126

Zhong, Y., 2009. Intrinsic shape signatures: A shape descriptor for 3D object recognition. In *Proceedings of the 2009 International Conference on Computer Vision Workshops* (Kyoto, Japan, Sep. 2009), 689–696. IEEE. 17, 33

Zhou, Q.-Y.; Park, J.; and Koltun, V., 2016. Fast global registration. In *Proceedings of the 2016 European Conference on Computer Vision* (Amsterdam, The Netherlands, Oct. 2016), 766–782. Springer. 19

Zia, M. Z.; Nardi, L.; Jack, A.; Vespa, E.; Bodin, B.; Kelly, P. H.; and Davison, A. J., 2016. Comparative design space exploration of dense and semi-dense SLAM. In *Proceedings of the 2016 International Conference on Robotics and Automation* (Stockholm, Sweden, May 2016), 1292–1299. IEEE. doi: 10.1109/ICRA.2016.7487261. 6