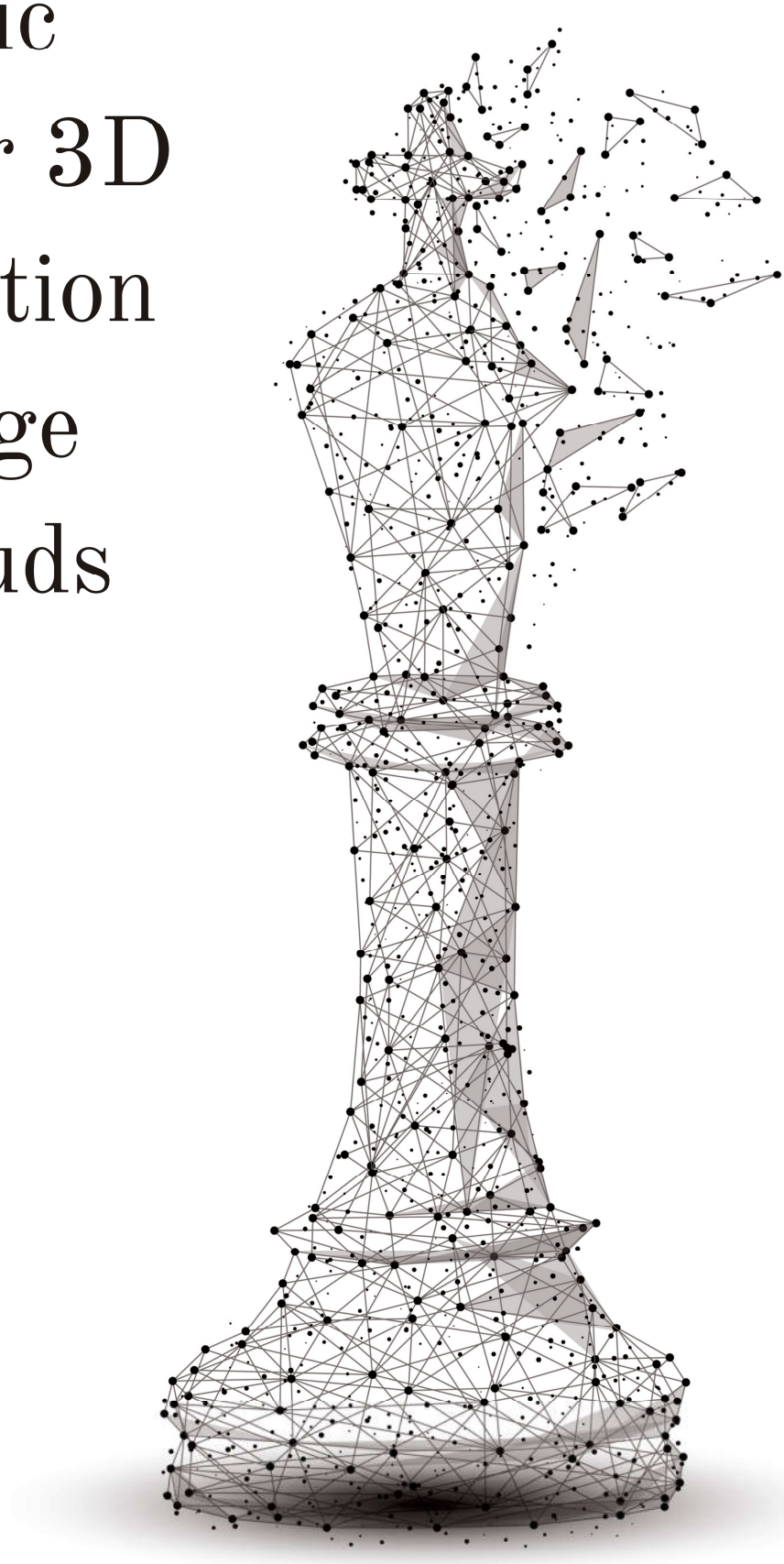
Geometric Methods for 3D Reconstruction from Large Point Clouds



BY TOLGA BIRDAL

Fakultät für Informatik
Computer Aided Medical Procedures & Augmented Reality / I16
TECHNISCHE UNIVERSITÄT MÜNCHEN

Geometric Methods for 3D Reconstruction from Large Point Clouds

TOLGA BIRDAL

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Prof. Dr.-Ing. Matthias Nießner
Prüfer der Dissertation:

1. Priv.-Doz. Dr. Slobodan Ilic
2. Prof. Dr.-Ing. Darius Burschka
3. Assistant Prof. Yasutaka Furukawa

Die Dissertation wurde am 12.10.2018 bei Technischen Universität München eingereicht und durch die Fakultät für Informatik am 07.12.2018 angenommen.

ABSTRACT

3D reconstruction involves the task of capturing the shape and appearance of objects or scenes in the form of 3D computer aided design (CAD) models, often through multiple measurements: individual 2D images or multiview 3D data. In this thesis, we explore both sparse and dense 3D reconstruction methods in scenarios where 3D cues are present and rough, prior CAD models are at hand before the operation. Thanks to the proliferation of 3D sensors and increased accessibility of 3D data, we can now remain true to the 3D nature of our physical world in such digitization processes and utilize direct 3D input, *point clouds*, which can also alleviate problems of capture modality and illumination conditions.

Both sparse and dense reconstruction problems arise frequently either in industrial machine vision where the production processes of parts and goods are to be inspected, or in restoration applications where crude digital models are desired to be improved. We start by explaining our contributions to sparse yet accurate reconstruction from non-overlapping multiview images. The experience developed here have also been used to acquire accurate ground truth aiding the assessment of the next stage, dense reconstruction. We tackle the latter by proposing a novel, multiview point cloud based 3D reconstruction pipeline in which it is possible to incorporate CAD proxies. This is accomplished by first aligning all the scans not by an $O(N^2)$ inter-scan matching but by a linear scan-to-model registration. Such alignment is made possible by novel object detection and pose estimation algorithms. Next, respecting the deviations of the real data from the CAD model, we perform a CAD-free multi-scan refinement further increasing the accuracy both qualitatively and quantitatively. We also propose novel methods to initialize such large scale optimization problems and to infer information about the reliability of solutions, known as the uncertainty.

As seen, the necessity to incorporate the CAD models as proxies to reconstruction comes along with many challenges to be addressed. In this thesis, the prominent sub-tasks include preparing CAD models towards the reconstruction task, object detection, estimation of full six degree of freedom (DoF) rigid pose and pose graph optimization, in which the roughly aligned scans are brought to the final alignment. We address all of those problems with rigor. Moreover, thanks to our photogrammetric ground truth acquisition strategies, we present thorough evaluation of all tasks, in real datasets besides synthetic ones.

ZUSAMMENFASSUNG

3D-Rekonstruktion beinhaltet die Aufgabe, Form und Erscheinungsbild von Objekten oder Szenen in Gestalt von 3D Computer Aided Design (CAD) Modellen zu erfassen, oft durch Mehrfachmessungen: verschiedene 2D-Bilder oder 3D Mehrfachansichten. In dieser Dissertation untersuchen wir sowohl spärliche als auch dichte 3D-Rekonstruktionsmethoden in Szenarien, in denen 3D-Hinweise vorhanden sind und grobe CAD-Modelle vor dem Einsatz zur Verfügung stehen. Dank der Verbreitung von 3D-Sensoren und der verbesserten Zugänglichkeit von 3D-Daten können wir nun der 3D-Natur unserer physischen Welt treu bleiben in solchen Digitalisierungsprozessen und direkt 3D-Eingabsdaten, *Punktwolken*, nutzen, was auch auch Probleme mit Aufnahmemodalität und Beleuchtungsbedingungen mindern kann.

Sowohl spärliche als auch dichte Rekonstruktions-probleme treten häufig entweder bei der industriellen Bildverarbeitung, bei der die Produktionsprozesse von Teilen und Gütern untersucht werden sollen, als auch bei Restaurationsanwendungen, bei denen rohe digitale Modelle verbessert werden sollen, auf. Wir beginnen mit der Erklärung unserer Beiträge zur spärlichen, aber genauen Rekonstruktion aus nicht-überlappenden Bildern mehrerer Ansichten. Die hier aufgebaute Erfahrung wurde auch verwendet, um genaue Referenzdaten zu erhalten, welche der Beurteilung der nächsten Stufe, dichter Rekonstruktion, helfen. Zu dessen Bewältigung schlagen wir eine neuartige, auf Punktwolken aus mehreren Ansichten basierende 3D-Rekonstruktionspipeline, in die CAD-Proxies integriert werden können, vor. Dies wird erreicht, indem zuerst alle Scans nicht durch eine $O(N^2)$ Scan-zu-Scan-Anpassung, sondern durch eine lineare Scan-zu-Modell-Registrierung ausgerichtet werden. Eine solche Ausrichtung wird durch neuartige Algorithmen zur Objekterkennung und Posen-Schätzung ermöglicht. Als nächstes, unter Berücksichtigung der Abweichungen der realen Daten vom CAD-Modell, führen wir eine CAD-freie Verfeinerung durch, die die Genauigkeit sowohl qualitativ als auch quantitativ weiter erhöht. Wir schlagen auch neue Methoden vor, um solche groß angelegten Optimierungsprobleme zu initialisieren und Informationen über die Zuverlässigkeit von Lösungen abzuleiten, auch als Unsicherheit bekannt.

Die Notwendigkeit, die CAD-Modelle als Proxies in die Rekonstruktion zu integrieren, ist mit vielen Herausforderungen verbunden. In dieser Dissertation sind die Vorbereitung von CAD-Modellen auf die Rekonstruktionsaufgabe, die Objekt-Erkennung, die Schätzung aller sechs Freiheitsgrade (DoF) der Pose und die Pose-Graph-Optimierung, in denen die grob ausgerichteten Scans auf die endgültige Ausrichtung gebracht werden, die wichtigsten Teilaufgaben. Wir begegnen all diesen Problemen mit Strenge. Dank unserer photogrammetrischen Strategien zur Erfassung der Grundwahrheit präsentieren wir darüber hinaus eine gründliche Auswertung aller Teilaufgaben, sowohl in realen als auch in synthetischen Datensätzen.

DEDICATION AND ACKNOWLEDGEMENTS

With all my heart, I dedicate this thesis to my grandparents Saliha and Ramazan, who were highly influential in education life, and to my entire family. It was a courageous decision to move to Munich once again with a bigger goal. I couldn't have done this without them and the people surrounding me. Here is a toast to all who made this possible:

First and foremost, to my supervisor Dr. Slobodan Ilic and Siemens for having me and putting in all the necessary freedom as well as the financial support needed to run after my thoughts without headaches. The mindful environment created by them has probably been the most important ground for the prosperity of this work, if any ought to exist. In particular, I thank Claudio and Claudia for their positivity, optimism and unconditional help at all times, Frank for fruitful discussions, Patrick, Uwe and Helmuth for involving me in the real life Siemens engineering-cycles, to Rebecca for all the collaborations and inspiring conversations, and to Andres, with my intentions to share more in this given lifetime.

Further, to Prof. Darius Burschka, Prof. Yasutaka Furukawa and Prof. Matthias Nießner for serving on my PhD viva and taking the time to read the following 270 pages. To Dr. Peter Sturm for his delightful insights and positive collaboration. To Prof. Aytül Erçil for her lifelong presence and support. To Prof. Nassir Navab for always being there as the wise mentor. Along with him, I shall recall and thank all members of CAMP Chair, especially the computer vision group, including but not limited to Anees, Christian, David, Fabian, Fausto, Federico, Helisa, Hemal, Huseyin, Johanna, Iro, Keisuke, Leslie, Magda, Marco, Maximillian, Paul, Ralf, Sailesh, Salvatore the Jack, Shadi, Vasilis, Wadim, Yida, and numerous others for being sempre welcome to my spontaneous drop-bys. I was only lucky to engage in discussions with them.

Along the way, I was lucky to work with great younger minds acting as their thesis adviser. To them, for their dedication, teaching me in return and opening my mind. In particular, to Jenya and her warmhearted humbleness towards her exceptional talent; to Onur for his friendship and trust in going after my sometimes delusional ideas; to Alejandro, Fang and Ivana whose paths have crossed with mine at some point. To Adrian, with my wishes to stay eager, enthusiastic and always to be a free spirit. I must admit, you calling me *the most misunderstood person* will resonate in my ears for quite some time.

My collaborations in Munich have been enriched after my meeting with Rolf, a late-met soul mate if you will, and the Fantasmo team - in particular, Gordan, Jan, Ryan and Jameson. Here is to those amazing people. With that, I also cheer Mike of Autodesk for his friendship and care, and the rock-star founders of Carbon Robotics, Rosanna and Dan.

The roots of this thesis date back to the times of Gravi, my former start-up in Turkey. To all

people along with Emre, Emrah and Nihan who have ever stepped into our offices. Knowing you have certainly stimulated my eager for and belief in the presence of the good. Moreover, to Hakan of Mavi Ucak for being the reason of why I would do computer vision.

To the folks of computer vision and machine learning, especially to my conference pals, who made working in this field a tremendous adventure and a recurrent fun. Along with CVPR-Turkey, this list involves countless people some of which I shall mention. To Ozan, İlke, Amir, Sergül, Tuğçe, Ayşegül, Ahmet, Deniz, Nezihe, Alp, Alp Rıza, Ali, David, Öncel, Jesus, the INRIA WIllow Team. Anil. Let's have a big one for those anonymous reviewers who carried computer vision to its prestigious level today. May thy identities remain a mystery. I shall also reminisce the countless big shots of CVML communities, for leading the way.

Along the way, some, whose sanity I highly doubt, yet bravery I highly envy, have decided to walk with me or beside me. Here is big one to Benni, my co-thinker, who mercilessly planted the seeds of proper geometrical thinking in me and accompanied me at times where I felt nothing but despair. Then to Mira, for sharing my everyday latent struggles and putting up with my out of the blue comments. To Haowen, for his patience, endurance, Chineese gifts and making all ideas come to life. To Yongheng and Christiane, for their abilities of diving into uncharted territories. To one of my favorite geometers, Bertram, from whom I learned a great deal and took a lot of inspiration from.

No piece would be complete without mentioning the giants upon whose work we base our minor, incremental contributions. I have restlessly read and dug into the writings and memories of great mathematicians, artists, scientists and philosophers. Their words have fueled the very machinery of my mortal brain. I shall raise my glass to the ones who shed light upon the unknown, notably: Plato, Renatus *the Cartesian*, Heron and Euclid of Alexandria, Immanuel Kant, Bernhard Riemann, Henri Poincaré, Kurt Gödel, William Hamilton, Ludwig Wittgenstein, and Bertrand Russell. With this thesis, I hope to leave only a drop in the endless sea of knowledge, and pass the flag to all those who have newly arrived, and are only beginning a journey of their own: Fernando, Ivan, Mai, Jerome, Patrick, Roman, Sergey, Yumin and finally Mine. We shall meet again.

This thesis has been written on top of the coastal mountains of Turkey. Many people were generous and kind enough to host me there. Kudos to Çigdem, Lara, Şirin and İdris of the serene Karakaya Retreat. The peaceful wind and the tranquil view of the rocks during the sunset has greatly touched upon the words and figures of the upcoming pages. A toast to Latcho Camp and Latcho Band of Kabak. Thanks to their friendliness and the trance of the psychedelic ethnic tunes, even though I did not write a single word there, my thoughts were unleashed. A big one to Ali Nesin, locals of Şirince and their Mathematics Village for deeply inspiring thousands of mathematicians, geometers, philosophers and simply peripatetic thinkers, like myself. I hope to compile several upcoming manuscripts in all these places.

Without music, my time in Munich would be washed out and empty. I thank to all musicians of Munich who put soul and feelings above all to create art. I was blessed to play with some of them whose level I would never reach. This round goes to Vero, Philip, Stanko, Pung, Daniel and to all who we jammed together. A special cheers to Federico, for we managed to *keep calm, play and publish*. Also to Drumeo, for making my drum practices enjoyable. I thank all who lighten up the world with their colors of art: the street artists, painters, and students of

art. Without your mesmerizing work, I would not be able to inspire mine.

A bottoms up to the members of my Whisky Club, Stefan, Akrash, Mario, Hemal, Marco and Endi for making Saturdays a delight to wake up to (even though I could not). A weisswurst-beer breakfast has never tasted that good. I once again thank Hemal, *the salesman*, for introducing me to you guys. I hereby officially certify the nickname, *Sultan*.

Another round goes to Ceren, for keeping my mind stimulated and my imagination boundless at times where the world was just single-opinionated. To Yalın and Mustafa for their hospitality at İYTE, the high-tech hub in my beautiful hometown, İzmir. To Tolga, his wife Ece and their *epsilon* Ela for their friendship and making Europe feel crowded. Similarly, to new-dad Kaan, Neşe and their little one. To Turhan, for letting the time heal and putting our contact above all. It is hard for me to express in words the value of that. To James, with the hope that the magic wand of time rejoins us. To my big 'Sista' Meltem for her elderly life advice; to Pinar for being so close during my entire life, despite being thousands of kilometers away. To long distance friendships. Hence, to Suna, with the wishes that she, one day, does not settle for a single endeavor. To Zeynep, as life is only worthy with people who are always there, no matter what. To Osman and to the day we met in Lyon and had a romantic lunch, with my wishes that you would rejoin Michela soon and for good. This list is incomplete without my siblings, *the Telekom*, who made me feel the true brother and sister-hood. In no particular order, to Duygu, İlker, Demir, Öyküm, Sena, Melih, Efe, Candaş, Nur, Uğur, Özgür and Özlem.

To my mom, Hayrunnisa, and dad, Bayram, for their endless faith and trust in me. Thanks to their backing, Earth has always been a safe place. They have, without doubt put all the effort in my education a family could possibly have. To Ercan, Aylin and Irem and all the fun and the feeling of peace that they bring. To Nur, for being a true maternal friend whom I can always trust. To Eylem, and her visits whenever she has been around. To my vivacious cousins, Elif and Ezgi for making Munich basically home. A big toast to my big, crowded and chaotic Birdal-family for assuring that everyone is taken care of even when I am on the other side of the world. To Umut, my best friend, room mate and by now essential collaborator. His name would have by all means appeared at any stage of this dedication letter. I salute you with pride and honor. To Gul, for her never ending tolerance, joy (*cukcuk*) and for the fact that whatever she touches turns to gold.

This thesis has been written in major, unsung ordeal and vicissitudes. All along, there was one name to witness them all, stand with patience, and endure all the hardship that came along. To my dearest wife Cansu for her cheer, aid and company during this journey. I wholeheartedly wish that the smile on your face perpetuates forever.

To life, to live, and to the new beginnings! Cheers.

AUTHOR'S DECLARATION

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: ..... DATE: .....

Let all, albeit ignorant of geometry, enter here.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

List of Symbols and Notation

The next list describes several symbols that will be later used within the body of the document.

Abbreviations/Acronyms

CNN Convolutional Neural Network

EM Expectation Maximization

GPU Graphics Processing Unit

ICP Iterative Closest Point

IRLS Iteratively Reweighted Least Squares

MAP Maximum A-Posteriori

MCMC Markov Chain Monte Carlo

MLE Maximum Likelihood Estimation

ND N - Dimensional

PCA Principal Component Analysis

PPF Point Pair Features

SGD Stochastic Gradient Descent

Distributions

$\mathscr{B}(\mathbf{V}, \mathbf{\Lambda})$ Bingham distribution with concentration $\mathbf{\Lambda}$ and mode specified by $\mathbf{V}$

$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ Multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$

Number Sets

$\mathbb{C}$ Complex numbers

$\mathbb{DH}$ Dual quaternions

$\mathbb{D}$ Dual numbers

$\mathbb{H}$ Quaternions of Hamilton

$\mathbb{R}^d$ d-dimensional real numbers

Other Symbols

$\mathbb{S}^{d-1}$ d-dimensional sphere

$\mathcal{M}$ Manifold (Usually Riemannian)

$\mathcal{T}_{\mathbf{x}}\mathcal{M}$ Tangent space of manifold $\mathcal{M}$ at point $\mathbf{x}$

$SE(3)$ 3-dimensional special Euclidean group

$SO(3)$ 3-dimensional special orthogonal group

Variables

$\mathbf{p}$ Typically a point in $\mathbb{R}^3$

$\mathbf{Q}$ A conic section in 3D (quadratic form)

$\mathbf{q}$ A quaternion / a quadric

$\mathbf{R}$ A rotation matrix ($SO(3)$)

$\mathbf{t}$ A translation vector

$\mathbf{x}$ A vector in $\mathbb{R}^d$, usually denoting the unknown

CHAPTER 1

INTRODUCTION

"Then, my noble friend, geometry will draw the soul towards truth, and create the spirit of philosophy, and raise up that which is not unhappily allowed to fall down."

— Plato, *The Republic, VII, 52*

Computer vision, one of the notoriously difficult problems of engineering research, has given rise to some of the most prominent products of the late human intellect and of the curiosity to understand and represent our world. Its applications have been plentiful, from aiding visually impaired members of our society to automatizing grocery/fashion stores. It is now used in many advanced AI systems, as the enabler of artificial sight. "*If we want the machines to think, we should teach them to see*", quotes Fei Fei Li, the widely recognized computer vision scholar. The industrial counterpart of computer vision, *machine vision*, has been rather concerned with precision, enabling machines with exceptionally high tolerances, robustness, determinism, repeatability and stability. Such proliferation is perhaps not surprising, because cameras have already been found to provide the most reliable information and thus to be the most prosperous among all the sensors [182]. Until recently, many of the computer vision systems utilized 2D information, and whenever necessary inferred the 3D structure of the world from a collection of these projections. In fact, it is now commonly accepted

that computer vision has been born during the Ph.D. of Larry Roberts, who has tackled the task of extracting 3D geometrical information from 2D perspective views of blocks [134].

With the advance in 3D sensing, depth sensors, laser scanners and lidars have begun to let us directly perceive all the spatial dimensions of our environment. Yet, for several years, these sensors have remained on the shelves due to their high price and lack of application areas. The cost of 3D sensing has started to decrease with the release of the depth sensor Microsoft Kinect (`www.xbox.com/en-US/kinect`) in 2010, from tens of thousands of dollars to a couple of hundreds, rendering 3D sensing affordable. The next wave predicts a similar, maybe a more dramatic price drop on the lidars and laser scanners that provide accurate 3D information in the form of sparse 3D points [1]. Furthermore, thanks to the pervasiveness of autonomous driving, processing the information coming from these sensors is now a more demanded task than ever. Accurate 3D capture also opens up immense potential in machine vision, particularly Industry 4.0, the fourth industrial revolution. Quality inspection and 3D digitization of manufactured parts are now two of the most desired idiosyncrasies of 3D machine vision. 3D optical inspection, a sub-branch of precision engineering, can now aim to achieve non-contact micron level accuracy in coordinate measurement [124, 239]. The reduction of the cost combined with the proliferation of application domains, have now positioned 3D computer vision at the center of machine perception and autonomous systems.

Reconstruction from an image collection is well studied in the 2D domain and is often resolved via structure from motion pipelines involving a global bundle adjustment as a final stage [242, 291, 294]. Unfortunately, due to the lack of 3D local geometry information, carrying out a similar task in 3D is way more challenging, and no well accepted method exists. Moreover, due to the unstructured-ness of data and nature of representation, the point clouds resulting from 3D capture processes carry more geometric information and less reliable appearence information compared to high resolution images. Hence, many of the modern deep learning techniques that excel on images fail to perform well on such sparse, higher dimensional, unstructured and unorganized input [308]. For instance, as shown by Hodaň *et al.* [130], geometric

[1] A startup named Luminar (`www.luminartech.com`) has already begun the production and distribution of low cost 3D lidars `www.wired.com/story/luminar-lidar-self-driving-cars/`.

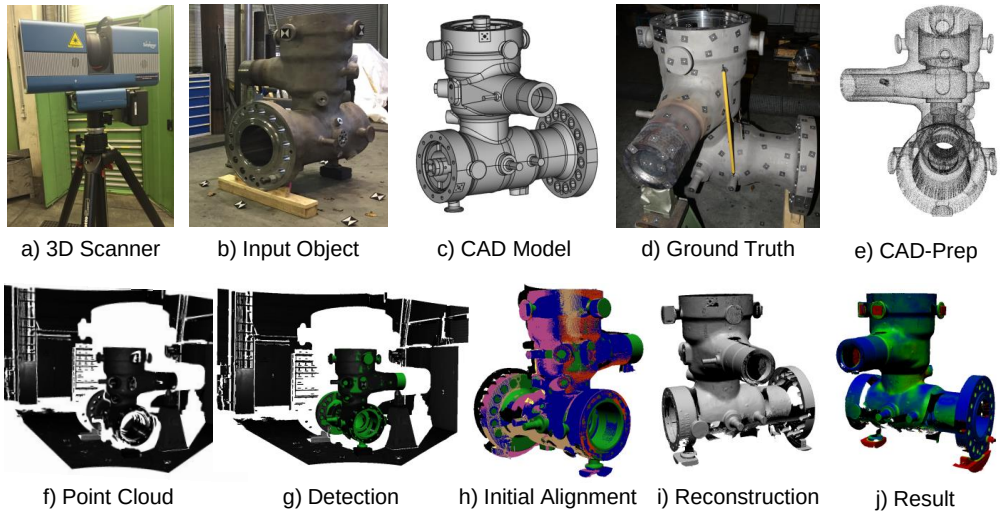


Figure 1.1: Building blocks of reconstruction using 3D data. A laser scanner **(a)** or similar 3D sensor is used to acquire 3D point clouds from the real world as shown in **(b)**. In many scenarios we also possess a CAD model of the object **(c)** to be reconstructed, typically a crude one. We develop methods for acquiring validation data by the use of fiducial tags and sparse coordinate triangulation **(d)**. The entire pipeline is composed of individual components, such as preprocessing of the CAD model **(e)** (and the scenes) serving the upcoming operations; object detection and pose estimation in 3D scans **(f, g)**; a rough initialization of the optimization procedure that governs the global refinement **(h)**; and a final 3D global alignment **(i)**. The final result typically follows a meshing operation and as shown in **(j)**, and can be used for various purposes such as storage, quality inspection, visualization or transmission. This thesis develops the necessary tools for realizing this pipeline in a robust and efficient manner.

methods are still the top performers at the task of 6DoF pose estimation.

In this thesis, we develop and mature an ensemble of geometric tools for processing such multi-scan point cloud data. Our algorithms unite under a novel system that achieves low complexity 3D reconstruction from multiple 3D scenes. In brief, our new framework fuses several low level building blocks such as object detection, pose estimation, model downsampling and pose graph optimization into a single pipeline that is able to utilize prior CAD data within the reconstruction process from unorganized scan collections. We interpret these blocks as a geometric toolset for point cloud processing and advance on the state of the art for all of them as summarized in Fig. 1.1.

1.1 Motivation

The limitations of 2D computer vision, such as lack of scale, and accurate modeling of textureless regions, can be easily tackled by inclusion of dense 3D data, acquired from 3D sensors. Yet, as an emerging field, robust and efficient utilization of the 3D information to solve a bunch of computer vision tasks is by far a non-saturated area. 3D data comes with its own difficulties. Occlusions are more severe, and sensors that capture a single viewpoint provide a sparser set of the geometry present in the entire 3D environment. Moreover, due to noise, unstructured-ness and boundary effects, 3D data obtained from a single view remains to be challenging for complete 3D understanding. These have led the researchers to turn to multi-scan capture and combine the individual data to a globally coherent 3D reconstruction [208], just like the 2D structure from motion. The resulting 3D models can then be utilized in many applications of computer vision, such as augmented reality, quality inspection, bin picking or autonomous navigation. Unfortunately, the process of reconstructing scenes from multiple 3D acquisitions involves many challenges including registration, fusion and overcoming computational and storage concerns. Some of these issues can be addressed by assuming that the order of scans are constrained and that the camera can be tracked through the capture sequence [74, 208, 253]. Yet, this is very restrictive from a user perspective, and is non-applicable to certain scenarios such as laser scanning of large objects/scenes.

The drawback of these aforementioned approaches and the lack of such a 3D vision system in the literature motivate us to bring in new solutions to this problem.

1.2 Objectives and Contributions

The overall objective of this thesis is to develop and excel a new 3D reconstruction pipeline that would allow a wide variety of applications from digitization in Industry 4.0 to robotic applications, commodity 3D scanning and augmented reality. The solution requires many algorithmic building blocks to be implemented, seamlessly glued together and improved. Each building block is bound to satisfy certain functional requirements in order to give rise to the most effective and robust system.

Possible applications Reconstructing objects and scenes is of interest to many domains. First, many of the augmented reality applications rely on correct identification of the object type and pose. While most AR applications use 2D data, the trend is shifting towards 3D, thanks to the advances in mixed reality and virtual reality wearables. Next, 3D printing industry craves for accurate models that are print-friendly. Naturally, practitioners of this domain often demand certified and precise solutions. Industry 4.0 is another initiative that tries to revolutionize how we manufacture goods. There, it is critical to have all the products quality inspected and digitized. In this context, digitization refers to gathering all properties and attributes of factory processes and products within a single smart body of systems that can later communicate, cooperate, control and inform about the progress and outcomes. Having a 3D CAD reconstruction is certainly one of the essentials of digitization.

On the technical side of the medallion, many higher-level computer vision algorithms can benefit from tools and ideas developed as part of this thesis. Semantic scene perception approaches now aim to bring a holistic perspective, where all relevant information about all the meaningful parts of the scene are to be inferred - often in the form of a scene graph. These techniques have recently turned to object-level processing thanks to large, annotated datasets and 3D object databases [188, 262]. Because our final pipeline shares a common backend with well studied methods like SLAM or SfM, our processing methods will in return advance the frontiers of these widespread research fields. Especially, because global optimization and object detection are frequently used in these core methods, 3D object priors and uncertainty-aware global pose graph initialization can be directly plugged into some of the existing pipelines [19, 234].

Desirable properties Due to the wide applicability and industrial constraints, our reconstruction algorithms have to be easy to operate, accurate, repeatable and reliable. For the reasons of usage simplicity and for the sake of generalizability, we neither assume a scanning order nor restrict the volume we operate in. Moreover, to overcome illumination effects, we avoid using color information. This also helps to generalize to a wider variety of 3D sensors. Depending on the application scenario, sparse coordinate measurement or dense capture is desired. Thus, our approaches have to address

both in a similar fashion. Since it is tedious to re-acquire laser scanner data[2], our systems should be capable of providing live feedback about the coverage of objects, quality of shots and the usability of input data. Thus, speed is a crucial aspect of the entire design. To avoid human involvement in the loop, our detectors are supposed to operate with almost-zero false positive rate, requiring a conservative hypotheses verification integrated. While initialization as well as online feedback can be delivered by approximate algorithms, we would like the final result to have as high fidelity as possible. Therefore, a rigorous refinement stage is necessary, enjoying initializations that are close enough and computed without harming the runtime.

In addition to the requirements above, we also have some operational constraints. First, CAD models coming from industry, or suited for computer graphics applications, are usually not friendly for processing methods involving computer vision. They are rather used in design phases, games or visualization tools. It is, hence, critical to convert these models to a standardized representation without information loss. Secondly, to assess the quality and report correct estimates of the accuracy, we should have good ground truth data and this thesis should certainly address such efforts.

Algorithms required to enable applications The issues of 3D reconstruction from 3D data necessitates smarter algorithms that are easier to integrate, use and at the same time, efficient and accurate. Luckily, many applications enjoy the existence of object priors, and one of our main objectives in this thesis is to engineer a pipeline around this prior such that the goals and requisites are met.

We first need to preprocess CAD models to make them usable in our pipeline. However, it is unlikely that all CAD models are presented in similar formats or properties. Therefore, special care is required handling parametric or mesh CAD forms and converting them to representations that are usable in computer vision solutions. Generally, this stage boils down to re-meshing of the models, removal of hidden geometry and downsampling in order to reduce the size for the sake of efficiency. The further stages are divided into two: sparse and dense reconstruction. For a sparse reconstruction, we would like to precisely localize a 2D camera with respect to the CAD instance. That is to be achieved by multiview registration methods. For the dense counterpart, only 3D data is exploited and to begin with, a false positive aware object

[2] Obtaining an accurate scan approximately takes 30min. with a state of the art scanner.

detection and pose estimation method is required. Naturally, hypothesized object poses should be verified for utmost correctness. It is often desirable to have a seamless integration between hypothesizing and verifying, avoiding any user intervention. In a dense reconstruction setting, it is typical to operate with multiview shots / scans / images. Hence, any pipeline addressing reconstruction should have capabilities of linking multiple poses within a fusion framework. As the CAD models available at hand will contain significant differences with respect to the real desired reconstruction, this fusion step should tolerate for discrepancies and misalignment in the pose estimation. Fusion is typically implemented as a global optimization of point sets in order to obtain a consistent reconstruction and camera localization across multiple shots. It is obvious that such an optimization is costly and requires good methods of initialization and we opt to devise an explicit solver for this.

Finally, it is critical to ensure that we can assess the quality of our algorithms. To this end, we require good techniques to collect ground truth data. We will do this by a fiducial marker based bundle adjustment procedure that is capable of handling overlapping scans. Ideas developed here will also be applied to sparse reconstruction, where only a set of keypoints are triangulated, but with high accuracy.

In this thesis, we first present our ground truth acquisition and sparse measurement methods. We then move to dense reconstruction and propose a novel pipeline. Next, we focus on all the individual requirements of the pipeline and propose solutions to overcome the current drawbacks.

Main Contributions To realize the required algorithms and to fulfill the objectives enlisted, we develop several novel methods that have the following contributions:

- A highly accurate sparse 3D point cloud reconstruction and calibration from targeted fiducial tags.
- Several algorithms for effectively downsampling point sets and CAD models.
- Efficient and highly accurate, sparse and dense reconstruction along with quality inspection of 3D parts using CAD models as proxies guiding the reconstruction.
- Novel object detectors that determine the pose of the objects in occluded and cluttered 3D scenes, as well as a method to localize parametric surfaces.

- Complimentary deep learning based approaches for rigid object and surface detection as well as pose estimation in point clouds.
- A new Bayesian optimization framework and its applications to uncertainty estimation for 3D pose graph optimization.

1.3 Publications

Authored

1. **Tolga Birdal**, Umut Şimşekli, Onur Eken & Slobodan Ilic:Bayesian Pose Graph Optimization via Bingham Distributions and Tempered Geodesic MCMC. In Advances in Neural Information Processing Systems (NIPS) 2018, Montréal, Quebec, Canada.
2. **Tolga Birdal**, Benjamin Busam, Nassir Navab, Slobodan Ilic & Peter Sturm: Generic Primitive Detection in Point Clouds Using Novel Minimal Quadric Fits. In ©IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI). *Tentative Accept with Revisions.*
3. **Tolga Birdal**, Benjamin Busam, Nassir Navab, Slobodan Ilic & Peter Sturm: A Minimalist Approach to Type-Agnostic Detection of Quadrics in Point Clouds. In ©IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2018, Salt Lake City, Utah, USA.
4. **Tolga Birdal**: Task Oriented 3D Sampling via Genetic Algorithms, ©IEEE Signal Processing and Communication Applications (SIU) 2018, Izmir, Turkey.
5. **Tolga Birdal** & Slobodan Ilic: CAD Priors for Accurate and Flexible 3D Reconstruction. In ©IEEE International Conference on Computer Vision (ICCV) 2017, Venice, Italy.
6. **Tolga Birdal** & Slobodan Ilic: A Point Sampling Algorithm for 3D Matching of Irregular Geometries, ©IEEE Intelligent Robots (IROS) 2017, Vancouver, Canada.
7. **Tolga Birdal**, Ievgeniia Dobryden & Slobodan Ilic: X-Tag: A Fiducial Tag for Flexible and Accurate Bundle Adjustment. In ©IEEE International Conference on 3D Vision (3DV) 2016, Stanford, USA.

8. **Tolga Birdal**, Emrah Bala, Tolga Eren & Slobodan Ilic: Online Inspection of 3D Parts via a Locally Overlapping Camera Network. In ©IEEE Winter Applications of Computer Vision (WACV) 2016, Lake Placid, USA.
9. **Tolga Birdal** & Slobodan Ilic: Point Pair Features Based Object Detection and Pose Estimation Revisited. In ©IEEE International Conference on 3D Vision (3DV) 2015, Lyon, France.
10. **Tolga Birdal**, Diana Mateus & Slobodan Ilic: Towards A Complete Framework For Deformable Surface Recovery Using RGBD Cameras. In ©IEEE Intelligent Robots (IROSW) 2012, Vilamoura, Portugal.

Co-authored

1. Haowen Deng, **Tolga Birdal** & Slobodan Ilic: PPF-FoldNet: Unsupervised Learning of Rotation Invariant 3D Local Descriptors. In European Conference on Computer Vision (ECCV) 2018, Munich, Germany.
2. Adrian Haarbach, **Tolga Birdal** & Slobodan Ilic: Survey of Higher Order Rigid Body Motion Interpolation Methods for Keyframe Animation and Continuous-Time Trajectory Estimation. In ©IEEE International Conference on 3D Vision (3DV) 2018, Verona, Italy.
3. Haowen Deng, **Tolga Birdal** & Slobodan Ilic: PPFNet: Global Context Aware Local Features for Robust 3D Point Matching. In ©IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2018, Salt Lake City, Utah, USA.
4. Benjamin Busam, **Tolga Birdal** & Nassir Navab: Camera Pose Filtering with Local Regression Geodesics on the Riemannian Manifold of Dual Quaternions. In ©IEEE International Conference on Computer Vision (ICCVW) 2017, Venice, Italy.
5. Umut Şimşekli & **Tolga Birdal**: Unified Probabilistic Framework For Robust Decoding Of Linear Barcodes. In ©IEEE International Conference on Audio, Speech and Signal Processing (ICASSP) 2015, Brisbane, Australia.
6. Umut Şimşekli, **Tolga Birdal**, Emre Koc & Taylan Cemgil: A Factorization Based Recommender System for Online Services. In ©IEEE Signal Processing and Communication Applications (SIU) 2013, Cyprus.

Patents Parts of this thesis contain material published in the following patents:

1. **Tolga Birdal**: Computer-aided image processing method, WO2018137935A1.
2. **Tolga Birdal**, Ievgeniia Dobryden & Slobodan Ilic: Marking device, WO2018077535A1.

Awards The *reconstruction-via-detection* framework, proposed within this thesis and associated publications has been granted the prestigious **Young Professional Award 2016** by the European Machine Vision Association. Furthermore, the research and development for this thesis have received the **Ernst von Siemens Doctoral Scholarship**, given to a selected number of doctoral students in the body of Siemens.

1.4 Outline

We now briefly review the flow of this thesis and summarize the content of each chapter. Because multiple parts of this thesis were previously published, we provide both the related work and experimental evaluation per chapter. This also allows us to utilize multiple datasets and assessment methods.

Chapter 2 We will first begin by laying out the fundamentals, in which we explain the already established mathematical framework that is used as the basis of this thesis. This chapter will be composed of individual seemingly unrelated subsections, yet the methods devised thereafter will bridge the gaps. In particular, we will review point clouds, Riemennian and Projective geometry, pose representations and quadrics. We also touch upon point pair features, robust M-estimators and iterative closest point algorithm that are used in a bunch of methods proposed here.

Chapter 3 We first begin our attempts to collect good ground truth data, as well as methods of sparse reconstruction. In this context, Chapter 3 describes a new fiducial tag, **X-tag** that is amenable for applications in photogrammetry, camera calibration and sparse coordinate recovery through bundle adjustment. We also provide extensive qualitative and quantitative analysis on the low level vision problems X-tag can aid. The content is based upon our 3DV 2016 publication [34].

Chapter 4 Here, in Chapter 4, we propose a robust and online sparse reconstruction pipeline and accurate methods to generate validation data. In particular, a multi-view system composed of static, non-overlapping 2D cameras is developed to tackle the challenge of automated online inspection in production lines. The content is based upon our WACV 2016 publication [31].

Chapter 5 Moving onto dense reconstruction, this chapter maps out our *reconstruction-via-detection* framework, upon which the rest of the thesis will be built. Essentially, this pipeline uses the available CAD models of the industry as shape priors to ease and robustify the reconstruction from 3D data. Within this chapter, we will talk about the multi-view global alignment and how object detectors can contribute to reconstruction. The content is based upon our ICCV 2017 publication [36].

Chapter 6 We delve into the building blocks of the reconstruction pipeline, presented in Chapter 5. The first stage is the preparation of CAD models and downsampling of the scene for an efficient, yet still correct reconstruction. We review the recent re-meshing algorithms and specific to our point pair feature based object detectors, tailor a new CAD model sampling strategy. We show that this new algorithm can improve both the pose estimation accuracy and detection rate when compared to prior mesh decimation or sampling algorithms. The content is based upon our IROS 2017 and SIU 2018 publications [30, 37].

Chapter 7 This chapter explains our contributions to 3D object detection and pose estimation. Most of the detectors of this stage are based on point pair features [87] that are found among the top performers for pose estimation in cluttered scenes [130]. Though, we advance the state of the art in multiple directions: First, we present an

improved algorithm for 3D object detection that suffers less from quantization artifacts and object size. Next, we combine geometry and deep networks to learn 3D descriptors from point clouds. We outperform the state of the art by a large margin in these works. Next, we relax the assumption that a full rigid model is available and instead look for primitives/bases that are common in CAD models. We model these common bases as quadrics and propose a very efficient algorithm to spot primitives within point clouds. The content of this chapter is based upon a collection of papers [32, 35, 36, 80, 81].

Chapter 8 The final method chapter of this thesis is devoted to pose graph optimization and refinement that can be an essential post-processing stage of any 3D computer vision pipeline. We first elaborate on a scenario where the camera motion forms a trajectory and hence we can rely on the order of acquisition. There, we parameterize the camera pose via dual quaternions, and suggest a statistical, local camera trajectory smoothing. Next, we get rid of the mentioned assumption concerning the camera trajectory and take on the challenge of optimizing the full pose graph. Our contributions to this domain involve a new optimization framework that bridges the gap between Bayesian posterior sampling and optimization. This framework is capable of delivering uncertainty estimates in problems, where finding a local/global minimum is desired. The content is based upon our NIPS 2018 and ICCV 2018 publications [33, 50].

Chapter 9 and Appendix We conclude with a summary of the thesis and potential future research directions. In the appendix, we provide a derivation of our theoretical results regarding the pose graph optimization of Chapter 8.

A Note on the Notation Throughout the chapters, we will commonly re-use symbols. For instance, in quaternion-based pose estimation algorithms, $\mathbf{q}$ usually refers to a quaternion. Similarly, in sections describing the fitting of parametric forms, $\mathbf{q}$ will refer to a quadric. Thus, we will review the notation per chapter and define symbols where they are used, unless specified by the nomenclature.

CHAPTER 2

FUNDAMENTALS

"There is geometry in the humming of the strings."

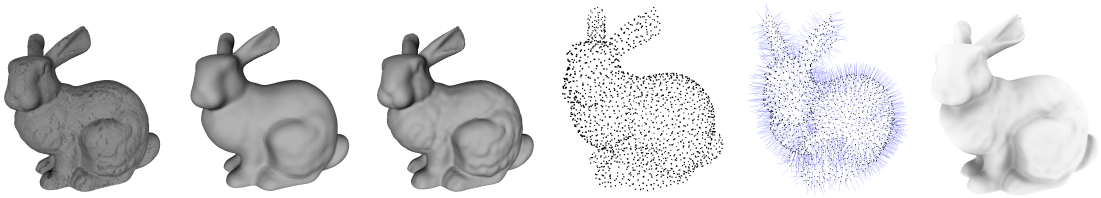
— Pythagoras

2.1 Point Clouds

We define a point cloud to be a d-dimensional point set composed of N points $\mathbf{X} \in \mathbb{R}^{d \times N}$. Usually the columns contain coordinates as in $\mathbf{x}_i = (x_i, y_i, z_i)^T \in \mathbf{X}$. Though, this could easily grow into more complex forms by introducing, for instance vertex normals: $\mathbf{x}_i = (x_i, y_i, z_i, n_i^x, n_i^y, n_i^z)^T$. Color components, subject to illumination changes or camera noise, will not be of particular interest in this thesis. Whenever possible, we omit the sensitive higher order terms such as curvature for the sake of robustness. A cloud composed of only 3D points is shown in 2.1(d), and its normals in 2.1(e).

2.1.1 Computing Model Normals

Given a point clouds, it always possible to compute the normal space by performing a local plane approximation [132] and recording the collection of tangent spaces. In other words, a surface normal of a point of interest $\mathbf{x} \in \mathbb{R}^3$ is computed by the principal



(a) (b) (c) (d) (e) (f)

Figure 2.1: **(a)** The original mesh. **(b)** 1^{st} order MLS smoothing. **(c)** 2^{nd} order approximation. **(d)** Poisson Disk Sampling. **(e)** Normals of sampled cloud. **(f)** Occlusion map.

component analysis of the covariance matrix $\mathbf{C} \in \mathbb{R}^{3\times3}$ obtained by incorporating the neighboring points $\mathbf{x}_i \in \Omega$ within the vicinity:

$$\mathbf{C} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \tag{2.1}$$

where $\bar{\mathbf{x}}$ denotes the mean, or the center of the local patch. The equation of the plane is then computed from the eigenvectors of $\mathbf{C}$. Due to the sign ambiguity in eigenvector analysis, the direction of the resulting normal is unknown. Thus, we use the convention where each surface normal is made to point towards the camera by ensuring that the dot product between the viewpoint vector and surface normal is acute:

$$-\mathbf{x}\cdot\mathbf{n}_\mathbf{x} < \frac{\pi}{2} \tag{2.2}$$

For more accurate / less noisy acquisitions, the neighborhoods and local structures can be better represented by higher order patches than planes. For instance, a higher fidelity approximation is obtained using 2^{nd} order terms, where the convexity and concavity are also modeled. Even though computing 2^{nd} order approximations are costly for online phase, it is safe to use them in the offline stage, for instance when CAD models available. Thus one seeks to find the parameters of a second order polynomial, approximating the height field of the neighboring points, given a local reference frame [6]. Formally, given a point $\mathbf{x} \in \mathbb{R}^3$, MLS operates by fitting a surface of order m in a local K-neighborhood $\{\mathbf{x}_k\}$ and projecting the point on this surface. Fitting is essentially a standard weighted least squares estimation (WLS) of the polynomial surface parameters. The closer the neighbors are, the higher the contribution is. This

is controlled by the weighting function $w(\mathbf{x}_i) = \exp(-\|\mathbf{x}-\mathbf{x}_k\|/2\sigma^2_{mls})$. The point $\mathbf{x}_i$ is then projected on the second order surface. This process is repeated for all points resulting in a smoothed point set with well defined normals. There are also works picking σ_{mls} adaptively, but we will not summarize those, as this falls out of the scope of this thesis. We refer the reader to [6] and show the effect of the MLS computation in Fig. 2.1(c) against the planar approximation shown in Fig. 2.1(b) concerning the mesh in Fig. 2.1(a).

2.2 Riemannian Geometry

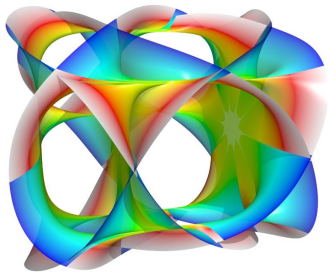


Figure 2.2: An illustration of Calabi-Yau manifolds commonly used by string theorists. *Eugenio Calabi and Shing-tung Yau*, Source: `goo.gl/VpHBYA`.

With every simple act of thinking, something permanent, substantial, enters our soul. This substantial somewhat appears to us as a unit but (in so far as it is the expression of something extended in space and time) it seems to contain an inner manifoldness; I therefore name it "mind-mass." All thinking is, accordingly, formation of new mind masses.

C. J. Keyser, "On the Psychology of Metaphysics. Being the Philosophical Fragments of Bernhard Riemann." The Monist (1900)

An infinitesimally small observation of the verse would not yield a global understanding on the whole space. In other words, Bernhard Riemann has argued, in the middle of the nineteenth century, that shape of the space itself invalidates the Euclidean geometry, and devised, based on other form of dot products in the tangent space, a non-Euclidean perspective that is smooth and is free of the Euclid's assumptions of flat space: Near the earth, the universe looks roughly like three dimensional Euclidean space; yet, near very heavy stars and black holes, the space is curved and bent. In particular, Riemann's geometry completely rejects the validity of Euclid's fifth postulate (through a point not on a given line there is only one line parallel to the given line) and modifies the second postulate (a straight line of finite length can be extended continuously without bounds). In this chapter, we will briefly introduce

the basic notion on Riemannian geometry that will ease the understanding of the following chapters.

Definition 2.2.1 (Manifold)**.** An n-dimensional manifold $\mathscr{M}$ (n-manifold) is a second countable Hausdorff space, where each point has a neighbourhood that is homeomorphic to the open n-dimensional Euclidean disc. While more rigorous definitions exist in topology, we will stick to that simplistic view for brevity.

Definition 2.2.2 (Tangency)**.** A vector $\mathbf{v}$ is said to be *tangent* to a point $\mathbf{x} \in \mathscr{M}$ if $\mathbf{x}^T\mathbf{v} = 0$.

Definition 2.2.3 (Tangent Space)**.** A tangent space is the set $\mathscr{T}_x$ of tangent such vectors:

$$\mathscr{T}_x = \{\mathbf{v} \in \mathbb{R}^d : \mathbf{x}^T\mathbf{v} = 0\}. \tag{2.3}$$

Definition 2.2.4 (Riemannian metric)**.** On a smooth manifold $\mathscr{M}$, a Riemannian metric $\mathbf{G} \in \mathscr{T}_0^2(\mathscr{M})$ is a tensor $\mathbf{G} = g_{ij}dx^i \otimes dx^j$, where $g_{ij} = g(\partial_i, \partial_j)$. Therefore $g(U^i\partial_i, V^j\partial_j) = g_{ij}U^iV^j$.

Definition 2.2.5 (Riemannian manifold)**.** An m-dimensional *Riemannian manifold* $\mathscr{M}$, endowed with a *Riemannian metric* $\mathbf{G}$, is defined to be a differentiable, smooth curved space, equipped with the inner product $\langle \mathbf{u}, \mathbf{v} \rangle_x = \mathbf{u}^T\mathbf{G}\mathbf{v}$ in the tangent space $\mathscr{T}_x\mathscr{M}$, embedded in an ambient higher-dimensional Euclidean space $\mathbb{R}^n$.

Below we provide some examples of such manifolds in $\mathbb{R}^d$:

- Euclidean geometry: $\mathscr{E}^d = \{\mathbf{x} \in \mathbb{R}^d\}$, $\quad g_E = \delta_{ij}dx^i dx^j = \sum_{i=1}^n (dx^i)^2$.
- Polar coordinates in domain $y > 0$: To obtain the Riemannian metric, we substitute the polar representation of $\mathbb{R}^2$, $x = r\cos\phi$, $y = r\sin\phi$:

  $$dx = \cos\psi\, dr - r\sin\psi\, d\psi, \quad dy = \sin\psi\, dr + r\cos\psi\, d\psi \tag{2.4}$$

  and therefore:

  $$g_s = (dx)^2 + (dy)^2 = dr^2 + r^2(d\psi)^2 \tag{2.5}$$

- Sphere in $\mathbb{R}^3$: $\mathbb{S}^2 = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| = 1\} \subset R^3$: Similar operation yields $g = r^2 d\theta^2 + a^2\sin^2\theta\, d\psi^2$, where $x = r\sin\theta\cos\psi$ and $y = r\sin^\theta\sin\psi z = r\cos\theta$.

- Quaternions, dual quaternions, covariance matrices, in general Lie groups and many other fundamental spaces live on Riemannian manifolds.

Definition 2.2.6 (Length)**.** Let γ denote a C^1 continuous curve on $\mathscr{M}$. The length of γ, Ł(γ) is given by:

$$L(\gamma) = \int_{\mathscr{M}} \sqrt{\left(g(\dot{\gamma}(t), \dot{\gamma}(t))\right)}\, dt \tag{2.6}$$

Definition 2.2.7 (Metric Space)**.** Let $(\mathscr{M}, \mathbf{G})$ be a path-connected Riemannian manifold. For two points $\mathbf{x}, \mathbf{y} \in \mathscr{M}$, denote the set of C^1-curves by C_{xy}. Let the curve $\gamma : [0,1] \to \mathscr{M}$ s.t. $\gamma(0) = \mathbf{x}$ and $\gamma(1) = \mathbf{y}$ and define $d : \mathscr{M} \times \mathscr{M} \to \mathbb{R}_0^+$:

$$d(\mathbf{x}, \mathbf{y}) = \inf\{L(\gamma) : \gamma \in C_{pq}\}. \tag{2.7}$$

Then $(\mathscr{M}, d)$ is said to be a **metric space** and therefore for all $\mathbf{x}, \mathbf{y}$ we have:

- $d(\mathbf{x}, \mathbf{y}) \geq 0$
- $d(\mathbf{x}, \mathbf{y}) = 0$ if and only if $\mathbf{x} = \mathbf{y}$
- $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$
- $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{r}) + d(\mathbf{r}, \mathbf{y})$ for any $\mathbf{r} \in \mathscr{M}$.

Definition 2.2.8 (Geodesic)**.** We define the geodesic on the manifold to be a constant speed, length minimizing curve between $\mathbf{x}, \mathbf{y} \in \mathscr{M}$, $\gamma : [0,1] \to \mathscr{M}$, with $\gamma(0) = \mathbf{x}$ and $\gamma(1) = \mathbf{y}$:

$$\frac{d\langle \dot{\gamma}(t), \dot{\gamma}(t) \rangle}{dt} = 0. \tag{2.8}$$

Definition 2.2.9 (Parallel Transport)**.** Oftentimes operation on manifolds require translating a vector $\mathbf{v}$ parallel to itself along a differentiable curve. This is analogous to the notion where two vectors in Euclidean space are transformed to the common origin by simple shifts and analyzed. For manifolds under consideration in this work, the parallel transport along the geodesic γ maps target vector $\boldsymbol{\gamma}'_x$ to $\boldsymbol{\gamma}'_y$ s.t. $\boldsymbol{\nabla}_{\gamma'} \boldsymbol{\gamma}' = 0$. The notion is illustrated in Fig. 2.3.

Definition 2.2.10 (Lie Groups and Exponential Maps)**.** A Lie group can be viewed as a differentiable Riemannian manifold. The Lie algebra to the Lie group is the tangent space at the identity of the group. Thus it gives a linearization of the Lie group near

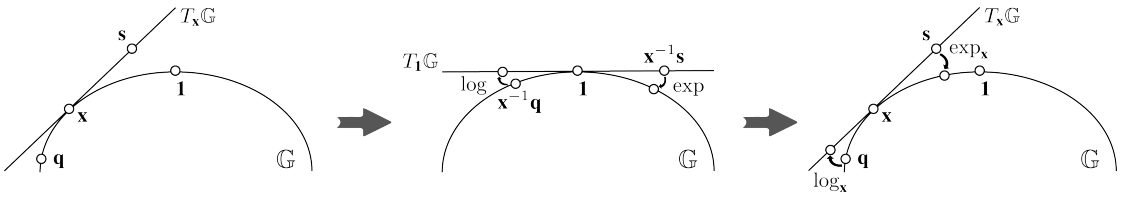


Figure 2.3: Parallel transport for the calculation of $\exp_{\mathbf{x}}$ and $\log_{\mathbf{x}}$. The exponential and logarithm maps at the support point $\mathbf{x}$ are indirectly calculated via the explicit maps at the identity.

the identity. In the generic sense, the map from the tangent space $\mathscr{X}_{\mathbf{x}}\mathbb{G}$ at $\mathbf{x}$ to the Lie group $\mathbb{G}$ is called a *retraction*. A length preserving retraction is *the exponential map* $\exp_{\mathbf{x}} : T_{\mathbf{x}}\mathbb{G} \to \mathbb{G}$, which is locally defined and maps a vector in the tangent space to a point on the manifold following the geodesic on $\mathbb{G}$ through $\mathbf{x}$. Its inverse is called the logarithm map $\log_{\mathbf{x}} : \mathbb{G} \to T_{\mathbf{x}}\mathbb{G}$. The mapping $\exp_{\mathbf{x}}$ at point $\mathbf{x} \in \mathbb{G}$ can be computed by parallel transport [100] as illustrated in Fig. 2.3. With the exponential map $\exp_{\mathbf{1}} =: \exp$ at the identity $\mathbf{1} \in \mathbb{G}$ and the logarithm map it holds

$$\begin{aligned} \exp_{\mathbf{x}}(\mathbf{s}) &= \mathbf{x}\exp\left(\mathbf{x}^{-1}\mathbf{s}\right), \\ \log_{\mathbf{x}}\left(\mathbf{q}\right) &= \mathbf{x}\log\left(\mathbf{x}^{-1}\mathbf{q}\right). \end{aligned} \tag{2.9}$$

2.3 Rigid Transformations

Definition 2.3.1 (Pose)**.** A pose of a rigid object is its distinguishable static state [46].

Definition 2.3.2 (Pose Space)**.** The set of possible poses compose a pose space.

In 3D, a 6DoF pose is described by a rotational and a translational component such that when applied to a 3D point leads to a *rigid* transformation. Now, we will analyze the possible representations of poses used throughout the rest of this thesis.

2.3.1 Special Orthogonal Group $SO(3)$

The linear rotation matrices belong to the *special orthogonal group*:

$$SO(3) = \{\mathbf{R} \in \mathbb{R}^{3\times 3} : \mathbf{R}^T\mathbf{R} = \mathbf{I} \text{ and } det(\mathbf{R}) = 1\} \tag{2.10}$$

The group acts linearly on a 3D point $\mathbf{x}$ and a Euclidean rotation is described by $\mathbf{x}' = \mathbf{R}\mathbf{x}$. To maximize consistency, we always use left multiplication.

2.3.2 Quaternions

In 1840s, during a walk from work to home, William Rowan Hamilton's *quaternions* were born, extending the complex numbers with three imaginary units $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ [119].

Definition 2.3.3 (Quaternion)**.** A **quaternion** $\mathbf{q}$ is an element of the Hamiltonian algebra $\mathbb{H}$ in the form:

$$\mathbb{H} = \{\mathbf{q} = q_1\mathbf{1} + q_2\mathbf{i} + q_3\mathbf{j} + q_4\mathbf{k} \in \mathbb{R}^4 : \mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -\mathbf{1}\}. \tag{2.11}$$

We also write $\mathbf{q} := [a, \mathbf{v}]$ with the scalar part $a = q_1 \in \mathbb{R}$ and the vector part $\mathbf{v} = \left(q_2, q_3, q_4\right)^{\mathrm{T}} \in \mathbb{R}^3$.

Definition 2.3.4 (Conjugate Quaternion)**.** The conjugate $\bar{\mathbf{q}}$ of the quaternion $\mathbf{q}$ is given by $\bar{\mathbf{q}} := q_1 - q_2\mathbf{i} - q_3\mathbf{j} - q_4\mathbf{k}$.

Definition 2.3.5 (Unit Quaternion)**.** A versor or **unit quaternion** $\mathbf{q} \in \mathbb{H}_1$ with

$$1 \stackrel{!}{=} \left\|\mathbf{q}\right\| := \mathbf{q} \cdot \bar{\mathbf{q}} \tag{2.12}$$

gives a compact and numerically stable parametrization to represent orientation and rotation of objects in $\mathbb{R}^3$ which avoids gimbal lock [163].

Definition 2.3.6 (Quaternion Inverse)**.** The inverse of quaternion $\mathbf{q}$ has the form $\mathbf{q}^{-1} = \bar{\mathbf{q}} / \|\bar{\mathbf{q}}\|^2$.

Definition 2.3.7 (Quaternion Product)**.** The non-commutative multiplication of two quaternions $\mathbf{p} := [p_1, \mathbf{v}_p]$ and $\mathbf{r} := [r_1, \mathbf{v}_r]$ is defined to be:

$$\mathbf{pr} = [p_1 r_1 - \mathbf{v}_p \cdot \mathbf{v}_r;\ p_1\mathbf{v}_r + r_1\mathbf{v}_p + \mathbf{v}_p \times \mathbf{v}_r] \tag{2.13}$$

Quaternions are of particular interest in computer vision due to their connection with spatial rotations [93].

Rotations with quaternions The rotation around the unit axis $\mathbf{v} = (v_1, v_2, v_3)^{\mathrm{T}} \in \mathbb{R}^3$ with angle θ is thereby given by

$$\mathbf{r} = \left[\cos(\theta/2), \sin(\theta/2)\,\mathbf{v}^T\right]^T. \tag{2.14}$$

Identifying antipodal points $\mathbf{q}$ and $-\mathbf{q}$ with the same element in $SO(3)$, the unit quaternions form a double covering group of the 3D rotations and any **point quaternion** or pure quaternion $\mathbf{p} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ of the point $\mathbf{u} = \left(x, y, z\right)^{\mathrm{T}} \in \mathbb{R}^3$ is rotated by the versor $\mathbf{r}$ via the sandwiching product map

$$\mathbf{p} \mapsto \mathbf{r} \cdot \mathbf{p} \cdot \bar{\mathbf{r}}. \tag{2.15}$$

Derivatives with quaternions The rotated quaternion $\mathbf{p}' = \mathbf{r}\mathbf{p}\bar{\mathbf{r}}$ needs to be differentiated with respect to the rotating quaternion $\mathbf{r}$, during various numerical optimization. The derivative can be represented using the matrix calculus notation:

$$\begin{aligned} \frac{\partial \mathbf{p}'}{\partial \mathbf{r}} &= \left[\frac{\partial \mathbf{p}'}{\partial \mathbf{r}_1}, \frac{\partial \mathbf{p}'}{\partial \mathbf{r}_2}, \frac{\partial \mathbf{p}'}{\partial \mathbf{r}_3}, \frac{\partial \mathbf{p}'}{\partial \mathbf{r}_4}\right] \\ &= [\mathbf{pq} - \overline{\mathbf{pq}}, \overline{\mathbf{pqi}} - \mathbf{pqi}, \overline{\mathbf{pqj}} - \mathbf{pqj}, \overline{\mathbf{pqk}} - \mathbf{pqk}] \end{aligned} \tag{2.16}$$

Relative quaternion rotation The relative orientation between two quaternions $\mathbf{q}$ and $\mathbf{p}$ is obtained simply by non-commutative Quaternion division $\mathbf{r} = \mathbf{q}\mathbf{p}^{-1}$.

Manifold of quaternions Unit quaternions form a hypersphere, $\mathbb{S}^3$, that is an embedded Riemannian submanifold of $\mathbb{R}^4$. It is not hard to see that $\mathbb{S}^3$ forms a Hausdorff space: Let $h : \mathbb{R}^d \to \mathbb{R}, h(x_1, \cdots, x_d) = x_1^2 + \cdots + x_d^2$. Due to the surjectivity of the Jacobian, h is a C^∞-submersion on $S^{d-1} = h^{-1}(1)$, valid for any $d > 1$. Due to the topology of the sphere, there is no unique way find a globally covering coordinate patch.

Exponential and Logarithm map in $\mathbb{H}$ The identity in $\mathbb{H}_1$ is given by $\mathbf{1} = (1, 0, 0, 0)^{\mathrm{T}}$. The tangent space $\mathcal{T}_{\mathbf{1}}\mathbb{H}_1$ is thus the hyperplane to the hypersphere $S^3 \in \mathbb{R}^4$ and parallel to the axes x_2, x_3, x_4 passing through $\mathbf{1}$. Any quaternion in $\mathcal{T}_{\mathbf{1}}\mathbb{H}_1$ is of the form

$$\mathbb{H} \ni \mathbf{q} = \left[0, \phi\mathbf{v}\right] \tag{2.17}$$

with $\mathbf{v} \in \mathbb{R}^3$, $\|\mathbf{v}\| = 1$ and the series writes as

$$\begin{aligned} \exp : \mathscr{T}_{\mathbf{1}}\mathbb{H}_1 &\to \mathbb{H}_1 \\ \mathbf{q} &\mapsto 1 + \sum_{k=1}^{\infty} \frac{\mathbf{q}^k}{k!} := \sum_{k=0}^{\infty} \frac{\mathbf{q}^k}{k!} \\ &= \cos(\phi) + \frac{\sin(\phi)}{\phi}\mathbf{q} = [\cos(\phi), \sin(\phi)\mathbf{v}] =: \mathbf{r}. \end{aligned} \tag{2.18}$$

where the second last step is done by recognizing the Taylor series for the sine and cosine function at 0. Note, that this relationship directly aligns with the notation in (2.14) while $\phi = \theta/2$ and the inverse function is given by

$$\begin{aligned} \log : \mathbb{H}_1 &\to \mathscr{T}_{\mathbf{1}}\mathbb{H}_1 \\ \mathbf{r} &\mapsto [0, \phi\mathbf{v}]. \end{aligned} \tag{2.19}$$

Sphere-specifically, the exponential map can be defined as:

$$\mathrm{Exp}(\mathbf{x}, \mathbf{u}) = \mathbf{x}\cos(\theta) + \mathbf{u}\sin(\theta)/\theta \tag{2.20}$$

where $\mathbf{u}$ denotes a tangent vector to $\mathbf{x}$. This gives multiple hints: 1) A unique geodesic can be defined by a single point and a direction vector. 2) The cut locus is only on the equator and thus the tangent space at any point can parameterize the rotation space and in fact this corresponds to the exponential coordinates. 3) This property decorates quaternions with a known analytic geodesic flow, given by [54]:

$$\begin{bmatrix} \mathbf{x}(t) & \mathbf{u}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{x}(0) & \mathbf{u}(0) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1/\alpha \end{bmatrix} \begin{bmatrix} \cos(\alpha t) & -\sin(\alpha t) \\ \sin(\alpha t) & \cos(\alpha t) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \alpha \end{bmatrix} \tag{2.21}$$

where $\alpha \triangleq \|\mathbf{u}(0)\|$. It is also useful to think about a quaternion as the normal vector to itself, due to the unitness of the hypersphere. By this property, projection onto $\mathscr{T}_x$ reads $P(\mathbf{x}) = \mathbf{I} - \mathbf{x}\mathbf{x}^T$ [54].

2.3.3 A Discussion of Rotation Representations

In general, regarding the rotation representations, it is possible to speak of a trade-off between the number of parameters vs the capability. The naive rotation matrices of the $SO(3)$ group are linear, capable of interpolation, gimbal lock free and yet have no

ambiguity [88]. The columns of a rotation matrix can be interpreted as a coordinate frame or the images of unit vectors under the rotation. These nice properties come at the cost of over-parameterization that one requires 9 parameters and hurts the interpolation - interpolating preserves neither speed nor acceleration.

The physically intuitive *axis-angle* parameterization uses only four values, but is unfortunately discrete around the identity and possesses infinitely many ambiguities. To overcome the limitation of discontinuity, *Rodrigues* parameterization absorbs the angle into the vector part, reducing four parameters to three. Unfortunately, any three-parameterization suffers from infinitely many ambiguities as the zero degree rotation can potentially correspond to infinitely many axes. As the norm gets potentially unbounded, there also exist infinitely many redundancies [115]. It is still hard to interpolate those vectors and tedious to define a proper metric.

Despite hardnesses, due to the minimal parameterization, many optimization algorithms tend to favor Rodrigues vector (exponential coordinates) [203]. That is due to $\mathbb{R}^3$ embedding and geodesics being straight lines [199]. This also leads to simpler Jacobian forms. In this thesis, we argue for the unit quaternions that are more suitable for the problems we tackle: Singularities in minimal representations make it hard to define distributions and perform continuous statistical optimization. Contrary to the exponential coordinates, for quaternions, the natural antipodally symmetric Bingham distributions exists.

2.3.4 Parameterizing the Special Euclidean Group $SE(3)$

Probably the most intuitive way to couple the required translation with rotations in order to perform a 6DoF transformation is by considering the $SE(3)$ group:

$$SE(3) = \left\{ \mathbf{T} \in \mathbb{R}^{4\times 4} : \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \text{ and } \mathbf{R} \in SO(3) \text{ and } \mathbf{t} \in \mathbb{R}^3 \right\}. \tag{2.22}$$

$\mathbf{T}$ parameterizes the full rigid body motion and acts linearly. A point $\mathbf{x} \in \mathbb{R}^3$ is transformed via the homogeneous form $\mathbf{T}\hat{\mathbf{x}} = \mathbf{R}\mathbf{x} + \mathbf{t}$.

Augmented quaternions An alternative, yet more compact parameterization of $SE(3)$ arises when rotational part is formulated as a quaternion and translational part

is treated disjointly. This leads to the following rigid body transformation equation:

$$\mathbf{x}' = \mathbf{R}\mathbf{x} + \mathbf{t} = \mathbf{q}\mathbf{x}\bar{\mathbf{q}} + \mathbf{t} \tag{2.23}$$

Note the slight abuse of rotation where the point $\mathbf{x}$ is considered to be of the pure form. We term this new product manifold $\mathbb{H}_1 \times \mathbb{R}^3$ as the *augmented quaternions*.

Both of the aforementioned representations split the translations and thereby require the practitioner to take explicit care of the components. Dual quaternions of Clifford, which couple the Riemannian and Euclidean manifolds via dual numbers give a more compact representation for the combination. The next section will cover this algebra.

2.3.5 Dual Quaternions

Similar to the representation of rotations by quaternions, we can use **dual quaternions** of unit length to represent spatial displacements, involving rotations and translations. To do so, we can define a dual quaternion as an ordered pair of quaternions with dual numbers as coefficients. A **dual number** Z is an element of the Clifford algebra $\mathbb{D}$ that can be written as [149] $Z = r + \varepsilon s$ where $r, s \in \mathbb{R}$ and $\varepsilon^2 = 0$, where r is the real-part, s is the dual part, and ε is called the dual operator. The dual conjugate is similar to the complex conjugate of $\mathbb{R} + i\mathbb{R}$. It is given by $\hat{Z} := r - \varepsilon s$. Extending this concept to quaternions, we can define dual quaternions:

Definition 2.3.8. A **dual quaternion** $\mathbf{w} \in \mathbb{DH}$ is an ordered set of quaternions

$$\mathbf{w} = \mathbf{r} + \varepsilon\mathbf{s} = \left(q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\right)^{\mathrm{T}}, \tag{2.24}$$

where $\mathbf{r}, \mathbf{s} \in \mathbb{H}$, $\left(q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\right)^{\mathrm{T}} \in \mathbb{R}^8$ and

$$\varepsilon^2 = 0, \quad \varepsilon\mathbf{i} = \mathbf{i}\varepsilon, \quad \varepsilon\mathbf{j} = \mathbf{j}\varepsilon, \quad \varepsilon\mathbf{k} = \mathbf{k}\varepsilon. \tag{2.25}$$

The Clifford algebra of dual quaternions contains the real numbers $\mathbb{R}$, the complex numbers $\mathbb{C}$, the dual numbers $\mathbb{D}$, and the quaternions $\mathbb{H}$ as sub-algebras. With the conjugate $\overline{\mathbf{w}} := \bar{\mathbf{r}} + \varepsilon\bar{\mathbf{s}}$ of the dual quaternion $\mathbf{w} = \mathbf{r} + \varepsilon\mathbf{s}$ we can study the constraints given for unit **unit dual quaternions** $\mathbf{w} \in \mathbb{DH}_1$. If we demand unit length, it holds

$$1 \stackrel{!}{=} \|\mathbf{w}\| := \mathbf{w} \cdot \overline{\mathbf{w}} = \mathbf{r}\bar{\mathbf{r}} + \varepsilon\left(\mathbf{r}\bar{\mathbf{s}} + \mathbf{s}\bar{\mathbf{r}}\right), \tag{2.26}$$

which gives the two distinct constraints

$$\mathbf{r}\bar{\mathbf{r}} = 1 \quad \text{and} \quad \mathbf{r}\bar{\mathbf{s}} + \mathbf{s}\bar{\mathbf{r}} = 0. \tag{2.27}$$

Displacements with Dual Quaternions

The unit dual quaternions are isomorphic to the group of rigid body displacement $SE(3)$ [3] and the two constraints (2.27) reduce the eight parameters of the dual quaternions to the six degrees of freedom of a rigid motion in space with its translation and rotation. If we write the translation as a pure quaternion $\mathbf{t}$ and the rotation as a unit quaternion $\mathbf{r}$ (2.14), we can construct the unit dual quaternion

$$\mathbb{DH}_1 \ni \mathbf{w} = \mathbf{r} + \varepsilon \frac{1}{2}\mathbf{t}\mathbf{r}. \tag{2.28}$$

Analogously to the quaternions, we formulate the **dual pure quaternion P** for the point $\mathbf{p} = [0, \mathbf{u}]$ as $\mathbf{P} = \mathbf{1} + \varepsilon\mathbf{u}$ and the spatial displacement becomes the sandwiching product map on dual quaternions

$$\mathbf{P} \mapsto \mathbf{w} \cdot \mathbf{P} \cdot \overline{\mathbf{w}} = 1 + \varepsilon\left(\mathbf{r}\mathbf{u}\bar{\mathbf{r}} + \mathbf{t}\right), \tag{2.29}$$

where the conjugates for the dual quaternion and the dual are calculated consecutively.

Geometry of $\mathbb{H}_1$ and $\mathbb{DH}_1$ With constraint (2.12), the unit quaternions form the three dimensional hypersphere $S^3 \in \mathbb{R}^4$. Thus $\mathbb{H}_1$ is isomorphic to the real projective space $\mathbb{RP}^3$. Looking at the two constraints from (2.27), we can analyze the structure of the unit dual quaternion space. The first equation $\|\mathbf{r}\| = 1$ forces the real part $\mathbf{r}$ of $\mathbf{w}$ to be of unit length, hence $\mathbf{r} \in \mathbb{H}_1$. This gives the 7-dimensional hypersphere $S^7 \in \mathbb{R}^8$ and the identification of antipodal points forms the seven dimensional real projective space $\mathbb{RP}^7$. The second equation reads as $\mathbf{r}\bar{\mathbf{s}} = -\mathbf{s}\bar{\mathbf{r}}$ and thus defines a quadric in $\mathbb{RP}^7$. Thus $\mathbb{DH}_1$ is not a hypersphere. This need to be considered for any operation on the manifold.

Exponential and Logarithm map in $\mathbb{DH}$ As a next step, we want to derive the exponential and logarithm maps at the identity for the elements of the groups $SO(3)$

and $SE(3)$ in quaternion notation. For matrices these maps are well studied objects [203, 286]. We study the exponential maps directly in (dual) quaternion space using its definition as a Maclaurin series.

Let $\mathbb{DH} \ni \mathbf{w} = \omega \mathbf{w} + \varepsilon \psi \mathbf{w}_\varepsilon$ be a pure dual quaternion with the two pure quaternions $\mathbf{w}, \mathbf{w}_\varepsilon \in \mathbb{H}_1$. Simplification of the Maclaurin series [245] for the exponential map then yields

$$\begin{aligned} \exp : T_{\mathbf{1}}\mathbb{DH}_1 &\to \mathbb{DH}_1 \\ \mathbf{w} \mapsto \sum_{k=0}^{\infty} \frac{\mathbf{w}^k}{k!} &= \frac{1}{2}\left(2\cos(\omega) + \omega \sin(\omega)\right) - \frac{1}{2\omega}\left(\omega\cos(\omega) - 3\sin(\omega)\right)\mathbf{w} \\ &\quad + \frac{1}{2\omega}\left(\sin(\omega)\right)\mathbf{w}^2 - \frac{1}{2\omega^3}\left(\omega\cos(\omega) - \sin(\omega)\right)\mathbf{w}^3. \end{aligned} \tag{2.30}$$

Before we compute the inverse function, we make the observation that any unit dual quaternion $\mathbf{w} = \left[\phi, \mathbf{v}\right] + \varepsilon\left[\phi_\varepsilon, \mathbf{v}_\varepsilon\right] =: [\Phi, \boldsymbol{\nu}]$ with the dual entities $\Phi = \phi + \phi_\varepsilon \varepsilon$ and $\boldsymbol{\nu} = \mathbf{v} + \mathbf{v}_\varepsilon \varepsilon$ can be written [77] equivalently to (2.14). For this, we calculate the dual trigonometric operators through a series expansion which brings

$$\begin{aligned} \sin(\Phi) &:= \sin\left(\phi\right) + \varepsilon\phi_\varepsilon \cos\left(\phi\right) \\ \cos(\Phi) &:= \cos\left(\phi\right) - \varepsilon\phi_\varepsilon \sin\left(\phi\right). \end{aligned} \tag{2.31}$$

We prove the following lemma by explicit calculation of the DQ representation:

Lemma 2.3.9. *Any unit dual quaternion* $\mathbf{w} \in \mathbb{DH}_1$ *can be written as*

$$\mathbf{w} = \left[\cos(\Theta/2), \sin(\Theta/2)\,\boldsymbol{\nu}\right], \tag{2.32}$$

where $\boldsymbol{\nu} \in \mathbb{DH}$ *is a pure dual quaternion.*

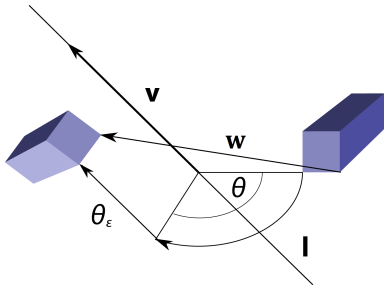


Figure 2.4: Screw linear displacement of rigid body with dual quaternion $\mathbf{w}$ along screw axis $\mathbf{l}$ with angle θ and pitch θ_ε in the direction of $\mathbf{v}$.

Proof. Analogously to the quaternion rotation, the formulation (2.32) can be understood as a parametrization of the rigid body motion. According to Chasles' Theorem [64], a displacement can be modeled via a translation along a unique axis with a simultaneous rotation about the same axis. This is visualized in Fig. 2.4. We construct

the dual quaternion displacement for this motion explicitly in the form (2.32). Let a rigid body transformation be given by a translation $\mathbf{t} \in \mathbb{R}^3$ and a rotation $\mathbf{R} \in \mathbb{R}^{3\times 3}$ around the axis $\mathbf{v}$ with $\|\mathbf{v}\| = 1$ with angle θ. From (2.28) we know already the unit dual quaternion for this displacement. The parameters for the screw motion are angle θ, pitch θ_ε, screw axis $\mathbf{l}$ with moment $\mathbf{v}_\varepsilon$ (i.e. $\mathbf{v}_\varepsilon = \mathbf{p} \times \mathbf{v} \;\forall\; \mathbf{p} \in \mathbf{l}$) and direction $\mathbf{v}$. The angle θ is directly given. We first compute the pitch θ_ε in the direction $\mathbf{v}$ of the axis as the projection of the translation onto the axis. This is $\theta_\varepsilon = \mathbf{t}^{\mathrm{T}}\mathbf{v}$. In order to recover the moment $\mathbf{v}_\varepsilon$, we pick a point $\mathbf{u}$ on the axis. With this we can describe $\mathbf{t}$ in terms of θ_ε, $\mathbf{v}$, $\mathbf{R}$ and $\mathbf{u}$ as

$$\mathbf{t} = \theta_\varepsilon \mathbf{v} + (\mathbf{I} - \mathbf{R})\,\mathbf{u} \tag{2.33}$$

and with the Rodrigues formula it holds

$$\mathbf{R}\mathbf{u} = \mathbf{u} + \sin(\theta)\,\mathbf{v} \times \mathbf{u} + (1 - \cos(\theta))\,\mathbf{v} \times (\mathbf{v} \times \mathbf{u})\,. \tag{2.34}$$

Thus substituting this into (2.33) gives with $\mathbf{u}^{\mathrm{T}}\mathbf{v} = 0$

$$\mathbf{u} = \frac{1}{2}\left(\mathbf{t} - \left(\mathbf{t}^{\mathrm{T}}\mathbf{v}\right)\mathbf{v} + \cot\left(\frac{\theta}{2}\right)\mathbf{v} \times \mathbf{t}\right), \tag{2.35}$$

which brings for the moment vector

$$\mathbf{v}_\varepsilon = \mathbf{u} \times \mathbf{v} = \frac{1}{2}\left(\mathbf{t} \times \mathbf{v} + \cot\left(\frac{\theta}{2}\right)\mathbf{v} \times (\mathbf{t} \times \mathbf{v})\right). \tag{2.36}$$

Substituting the rotation quaternion $\mathbf{r} = \left[q_0, \mathbf{w}\right]$ and using $\theta_\varepsilon = \mathbf{t}^{\mathrm{T}}\mathbf{v}$ yields

$$\sin\left(\frac{\theta}{2}\right)\mathbf{v}_\varepsilon + \frac{\theta_\varepsilon}{2}\cos\left(\frac{\theta}{2}\right)\mathbf{v} = \frac{1}{2}\left(\mathbf{t} \times \mathbf{w} + q_0\mathbf{t}\right), \tag{2.37}$$

which is the pure quaternion of the dual part. Thus

$$\mathbf{w} = \left[\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right)\mathbf{v}_\varepsilon\right] + \varepsilon\left[-\frac{\theta_\varepsilon}{2}\sin\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right)\mathbf{v}_\varepsilon + \frac{\theta_\varepsilon}{2}\cos\left(\frac{\theta}{2}\right)\mathbf{v}\right] \tag{2.38}$$

which equals (2.32) if we apply the trigonometric operators (2.31), (2.3.5) and the dual entity represantations.

We note that this representation separates the line information of the screw axis from the pitch and angle values in an algebraical way where the dual vector $\boldsymbol{\nu}$ represents the axis of a screw motion with its direction vector and the dual angle Θ contains

both the translation length and the angle of rotation.
Since for the exponential of a dual quaternion of the form $\mathbf{w} = \boldsymbol{\nu}\frac{\Theta}{2}$ it holds [146]

$$\exp\left(\boldsymbol{\nu}\frac{\Theta}{2}\right) = \left[\cos\left(\frac{\Theta}{2}\right), \sin\left(\frac{\Theta}{2}\right)\boldsymbol{\nu}\right], \tag{2.39}$$

the inverse function of exp, $\log : \mathbb{DH}_1 \to T_{\mathbf{1}}\mathbb{DH}_1$, for dual quaternions of the form (2.32) is then given by

$$\left[\cos\left(\frac{\Theta}{2}\right), \sin\left(\frac{\Theta}{2}\right)\boldsymbol{\nu}\right] \mapsto \boldsymbol{\nu}\frac{\Theta}{2}. \tag{2.40}$$

2.3.6 Metrics on Rigid Transformations

There exists multiple ways to measure distances in pose spaces. We will briefly summarize them below.

Definition 2.3.10 (Two-Valued Metric)**.** The simplest way to treat the rotational and translational parts of the pose is to use a two valued metric, one for rotation, one for translation, relegating the problem of fusing the two [88].

It is though desirable to have a single value in order to ease multiple aspects of the implementation; for instance during pose clustering, one likes to define closeness and belonging to the same cluster - defining two thresholds separately increases the efforts of parameter tuning. For that reason, other strategies, try to combine the two:

Definition 2.3.11 (Weighted generalized mean)**.** Treating the rotation and translation distances individually and then fusing them via a weighted average leads to the following, frequently used $SE(3)$ distance:

$$d(\mathbf{T}_1, \mathbf{T}_2) = \left(\alpha d_{\text{rot}}(\mathbf{R}_1, \mathbf{R}_2)^p + \beta d_{\text{trans}}(\mathbf{t}_1, \mathbf{t}_2)^p\right)^{1/p} \tag{2.41}$$

for any $p \in [1, \infty]$ and under any rotational d_{rot} and translational d_{trans} metric. Typically, d_{trans} is chosen to be Euclidean, whereas the choice for d_{rot} varies depending on the desired way to parameterize the rotation manifold. The scale factors α and β balance the contributions of the two metrics. When the object geometry or prior information on scene dimensions are available, $\mathbf{t}$ can be normalized for a better balance.

The need for cherry picking the scale factors often lead to suboptimal metrics. Therefore recent works [46, 88] consider an object space alternative, that is only valid when points of interest are at hand. For many practical applications of 3D computer vision, this is the case.

Definition 2.3.12 (Object space distance)**.** Brégier *et al.*define a frame-invariant pose distance between corresponding 3D points of instances of the object at given poses as

$$d(\mathbf{T}_1, \mathbf{T}_2) = \frac{1}{V}\Big(\int \mu(\mathbf{x}) \|\mathbf{T}_2(\mathbf{x}) - \mathbf{T}_1(\mathbf{x})\|^p \, dv\Big)^{1/p}. \tag{2.42}$$

$\mu(\mathbf{x})$ is a density distribution relative to the object and V a normalizing constant. This physically inspired metric takes into account the object shape. Drost [88] suggests a variant, that is defined over the convex hull rather than the entire object surface:

$$d_2(\mathbf{T}_1, \mathbf{T}_2) = \max\{\|\mathbf{T}_1(\mathbf{x}) - \mathbf{T}_2(\mathbf{x})\| : \mathbf{x} \in \mathbf{O}\} = \max\{\|\mathbf{x} - \mathbf{D}(\mathbf{x})\| : \mathbf{x} \in \mathbf{O}\} \tag{2.43}$$

where $\mathbf{O}$ defines the object and $\mathbf{D} = \mathbf{T}_1^{-1}\mathbf{T}_2$. Equality follows from the frame invariance.

2.3.7 Distributions on Pose Spaces

Definition 2.3.13 (Bingham Distribution)**.** Derived from a zero-mean Gaussian, the Bingham distribution [29] is an antipodally symmetric probability distribution conditioned to lie on $\mathbb{S}^{d-1}$ with probability density function (PDF) $\mathscr{B} : \mathbb{S}^{d-1} \to R$:

$$\mathscr{B}(\mathbf{x}; \mathbf{\Lambda}, \mathbf{V}) = \frac{1}{F}\exp(\mathbf{x}^T\mathbf{V}\mathbf{\Lambda}\mathbf{V}^T\mathbf{x}) = \frac{1}{F}\exp\big(\textstyle\sum_{i=1}^{d} \lambda_i (\mathbf{v}_i^T\mathbf{x})^2\big) \tag{2.44}$$

where $\mathbf{V} \in \mathbb{R}^{d\times d}$ is an orthogonal matrix ($\mathbf{V}\mathbf{V}^T = \mathbf{V}^T\mathbf{V} = \mathbf{I}_{d\times d}$) describing the orientation, $\mathbf{\Lambda} = \mathrm{diag}(0, \lambda_1, \cdots, \lambda_{d-1}) \in R^{d\times d}$ with $0 \geq \lambda_1 \geq \cdots \geq \lambda_{d-1}$ is the concentration matrix, and F is a normalization constant.

With this formulation, the mode of the distribution is obtained as the first column of $\mathbf{V}$. The antipodal symmetry of the PDF makes it amenable to explain the topology of quaternions, i. e., $\mathscr{B}(\mathbf{x}; \cdot) = \mathscr{B}(-\mathbf{x}; \cdot)$ holds for all $\mathbf{x} \in \mathbb{S}^{d-1}$. When $d = 4$ and $\lambda_1 = \lambda_2 = \lambda_3$, it is safe to write $\mathbf{\Lambda} = \mathrm{diag}([1, 0, 0, 0])$. In this case, the logarithm of the Bingham density reduces to the dot product of two quaternions $\mathbf{q}_1 \triangleq \mathbf{x}$ and the mode of the distribution,

say $\bar{\mathbf{q}}_2$. For rotations, this induces a metric, $d_{\text{bingham}} = (\mathbf{q}_1 \cdot \bar{\mathbf{q}}_2)^2 = \cos(\theta/2)^2$, that is closely related to the true Riemannian distance:

$$d_{\text{riemann}} = \|\log(\mathbf{R}_1 \mathbf{R}_2^T)\| \triangleq 2\arccos(|\mathbf{q}_1 \bar{\mathbf{q}}_2|) \triangleq 2\arccos(\sqrt{d_{\text{bingham}}}). \tag{2.45}$$

It is easy to verify that adding a multiple of the identity matrix $\mathbf{I}_{d\times d}$ to $\mathbf{V}$ does not change the distribution [29]. Thus, we conveniently force the first entry of $\mathbf{\Lambda}$ to be zero. Moreover, since it is possible to swap columns of $\mathbf{\Lambda}$, we can build $\mathbf{V}$ in a sorted fashion. This allows us to obtain the mode very easily by taking the first column of $\mathbf{V}$.

Covariance The covariance matrix $\mathbf{C} \in \mathbf{R}^{d\times d}$ of the Bingham distribution reads:

$$\mathbf{C} = \text{Cov}(\mathbf{x}) = -0.5(\mathbf{V}(\mathbf{\Lambda} + c\mathbf{I})\mathbf{V}^T)^{-1} \tag{2.46}$$

where $c \in \mathbb{R}$ can be arbitrarily chosen as long as $(\mathbf{\Lambda} + c\mathbf{I})$ is negative definite. Bingham distributions have been extensively used to represent distributions on quaternions [107, 108, 159]; however, to the best of our knowledge, never for the problems that are of concern in this thesis.

Mode vs Rotation Averaging Oftentimes, for instance during clustering, one needs to fit a Bingham distribution to a set of quaternions. Indeed, it is possible to show that the mode of such a fit lies exactly at the Markley average of the quaternions $\{\mathbf{q}_i\} \in \mathbf{Q}$ [185]. To estimate the mean we form the weighted dot product matrix:

$$\mathbf{A} = \frac{1}{n_q} \sum_{i=1}^{n_q} w_i^{\mathbf{q}} (\mathbf{q}_i^T \cdot \mathbf{q}_i) \tag{2.47}$$

where n_q is the number of rotations. The mean quaternion $\mathbf{q}_{avg}$ is given by the eigenvector $\mathbf{e}_{max}$ corresponding to the maximum eigenvalue of $\mathbf{A}$, λ_{max}. It is usually helpful to imagine Bingham distribution as a heatmap over the sphere centered around a particular (mode) quaternion.

2.4 Projective Geometry in Computer Vision

Projective geometry studies the properties of figures which remain unaltered (invariant) in projection. All the propositions in projective geometry occur in *dual pairs*, meaning that one proposition infers the other. Typically, duality is achieved by interchanging the parts played by the words "point" and "line".

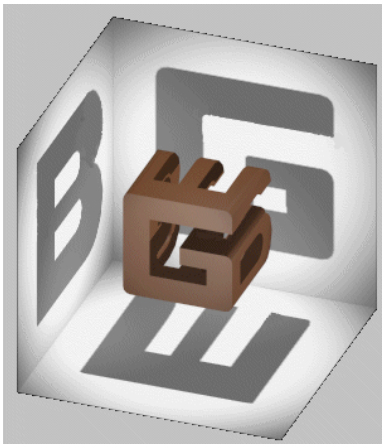


Figure 2.5: *Gödel, Escher, Bach,* Douglas Hofstader.

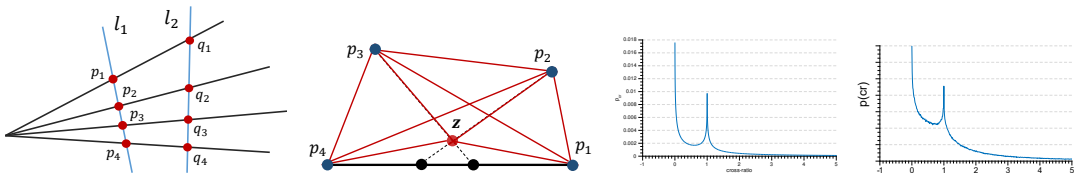
Less restrictive than the Euclidean geometry, the non-metrical projective geometry exhibits two fundamental invariants. First, as the third axiom of Whitehead posits, the notion that *lines remain as lines* and the existence of intersection is preserved under projection, giving rise to the *incidence structure*. The second one is the *cross-ratio*. We will now briefly summarize how projective geometry is used in computer vision and its invariants.

Homogeneous coordinates Introduced by August Ferdinand Möbius [196], homogeneous coordinates give an algebraic model for doing projective geometry analgous to the style of Cartesian coordinates of analytic geometry. Hence, they are of particular interest to computer vision community - they allow common vector operations e.g. translation, rotation, scaling and perspective projection, to be represented matrix-vector products. For a point on the image plane $\mathbf{p} = (p_x, p_y)$ one system of homogeneous coordinates is given by $\hat{\mathbf{p}} \mapsto (p_x, p_y, 1)$ with the inverse map is written as $(\alpha, \beta, \lambda) \mapsto (\alpha/\lambda, \beta/\lambda)$.

2.4.1 Cross Ratios

In its simplest form, cross ratio (CR) is defined for a pencil of lines passing through a center O and intersecting two lines (ℓ_1, ℓ_2) at points $\{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4\}$ and $\{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4\}$, respectively, where $\mathbf{p}_i \in \mathbb{R}^2$ and $\mathbf{q}_i \in \mathbb{R}^2$. This configuration is visualized in Fig. 2.6(a). The cross ratio of 4 such collinear points is defined as:

$$cr(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4) = cr(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4) = \frac{|\mathbf{p}_1\mathbf{p}_3||\mathbf{p}_2\mathbf{p}_3|}{|\mathbf{p}_1\mathbf{p}_4||\mathbf{p}_2\mathbf{p}_4|} \tag{2.48}$$



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

Figure 2.6: 1D line **(a)** and 2D triangle **(b)** configurations for computing a cross ratio. **(c,d)** plot the analytical and approximated distribution of cross ratios, respectively.

This invariant is naturally extended to 2D space [229], when the points are non-collinear, but co-planar. In this case, the configuration of five points defines the cross ratio using the ratio of product of triangle areas:

$$cr_{2D}(\mathbf{z},\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3,\mathbf{p}_4) = \frac{\Delta(\mathbf{z},\mathbf{p}_1,\mathbf{p}_2)\Delta(\mathbf{z},\mathbf{p}_3,\mathbf{p}_4)}{\Delta(\mathbf{z},\mathbf{p}_1,\mathbf{p}_3)\Delta(\mathbf{z},\mathbf{p}_2,\mathbf{p}_4)} \tag{2.49}$$

This is illustrated in Fig. 2.6(b). Clearly, one could generate multiple CR, by altering the permutations of points. Thus, a set of points define 24 CR, of which only 6 are unique.

Theorem 2.1 ([2, 136])**.** *The probability density function of 1D cross ratios have the following analytical form:*

(2.50)

$$f_X(x) = \begin{cases} f_1(x) + f_3(x) & if\, x < 0 \\ f_3(x) + f_2(x) & if\, x < 0 < 1 \\ f_2(x) + f_1(x) & if\, 1 < x \end{cases}$$

$$f_1(x) = \frac{1}{3}\Big((2x-1)ln(\frac{x}{x-1}) - 2\Big)$$
$$f_2(x) = \frac{1}{3}\Big(\frac{(x+1)ln(x) + 2(1-x)}{(x-1)^3}\Big)$$
$$f_3(x) = \frac{1}{3}\Big(\frac{(x-2)ln(1-x) + 2x}{x^3}\Big)$$

Similarly, it is also possible to define the cumulative distribution function (cdf). To validate whether the practice follows the analytic finding, we have collected a large dataset of random 1D cross ratios, and found out that also in practice, distribution of a set of CR closely approximates the analytical one. Fig.2.6(c) plots the analyitcal PDF of cross ratios and Fig.2.6(d) the estimated one over a large synthetic point dataset.

Joint invariants Even though [12] raises a contradictory claim, it is well known that cross ratio is very sensitive to noise [170, 190, 191]. This sensitivity and the non-uniqueness of single cross ratios, however, can be circumvented up to a certain extent

by relying on multiple invariants, a set of cross ratios, extracted from multiple combinations of input points [12]. Such a set is termed as the *joint invariants*, and defines the point set uniquely up to a projective transformation.

2.4.2 Pinhole Camera Model

Pinhole Camera Model approximates the image formation by assuming that light rays directed towards the camera pass through a tiny aperture, *a pinhole* and intersect at an infinitesimally small image plane, forming *camera obscura*, a flipped and inverted projection.

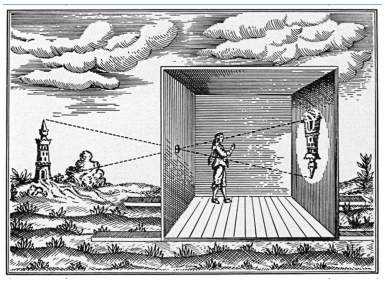


Figure 2.7: *Camera Obscura.* Source: PhotoIon Photography School `https://www.photoion.co.uk/blog/camera-obscura/`

In the most simplistic sense, a pinhole camera is represented by four main parameters: The distance between the camera center and the image plan, termed *focal length* is represented in pixels (f_x, f_y), and the point (c_x, c_y) where the principal axis hits the image plane, termed *principal point*. These quantities can be conveniently represented by the *intrinsic camera matrix*:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.51}$$

We will ignore the distortions caused by the skew and optic characteristics. Taking into account the position of the camera, a 3D point $\mathbf{x}$ gets projected onto the image plane using the homogeneous coordinate representation:

$$\hat{\mathbf{p}} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \hat{\mathbf{x}} \tag{2.52}$$

Finally the pixel $\mathbf{p} \in \mathbb{R}^2$ is obtained from $\hat{\mathbf{p}}$ by de-homogenization.

2.5 Quadrics: 3D Conic Sections

Being generalizations of conic sections, a *quadric* is a hypersurface (of dimension d) embedded in a $(d+1)$-dimensional space. The quadrics of concern in this work concentrate on the zero set of an irreducible polynomial of degree two and $d = 2$.

Definition 2.5.1 (Quadric)**.** Formally, a quadric in 3D Euclidean space is a hypersurface defined by the zero set of a polynomial of degree two:

$$f(x,y,z) = Ax^2 + By^2 + Cz^2 + 2Dxy + 2Exz + 2Fyz + 2Gx + 2Hy + 2Iz + J = 0.$$

Equation (2.5.1) can be written in the vector form as the coefficient-variable dot product $\mathbf{v}^T\mathbf{q} = 0$, with:

$$\mathbf{q} = \begin{bmatrix} A & B & C & D & E & F & G & H & I & J \end{bmatrix}^T \tag{2.53}$$
$$\mathbf{v} = \begin{bmatrix} x^2 & y^2 & z^2 & 2xy & 2xz & 2yz & 2x & 2y & 2z & 1 \end{bmatrix}^T$$

Using homogeneous coordinates, quadrics can be analyzed uniformly. The point $\mathbf{x} = (x,y,z) \in \mathbb{R}^3$ lies on the quadric, if the projective algebraic equation over $\mathbb{RP}^3$ with $d_q(\mathbf{x}) := [\mathbf{x}^T 1]\mathbf{Q}[\mathbf{x}^T 1]^T = 0$ holds true, where the matrix $\mathbf{Q} \in \mathbb{R}^{4\times4}$ is defined by re-arranging the coefficients:

$$\mathbf{Q} = \begin{bmatrix} A & D & E & G \\ D & B & F & H \\ E & F & C & I \\ G & H & I & J \end{bmatrix}, \quad \nabla\mathbf{Q} = 2\begin{bmatrix} A & D & E & G \\ D & B & F & H \\ E & F & C & I \end{bmatrix}. \tag{2.54}$$

$d_q(\mathbf{x})$ can be viewed as an algebraic distance function. Similar to the quadric equation, the gradient at a given point can be written as $\nabla\mathbf{Q}(\mathbf{x}) := \nabla\mathbf{Q}[\mathbf{x}^T 1]^T$. Quadrics are general implicit surfaces capable of representing cylinders, ellipsoids, cones, planes, hyperboloids, paraboloids and potentially the shapes interpolating any two of those, as shown in Fig. 2.8 on the right. All together there are 17 sub-types [288].

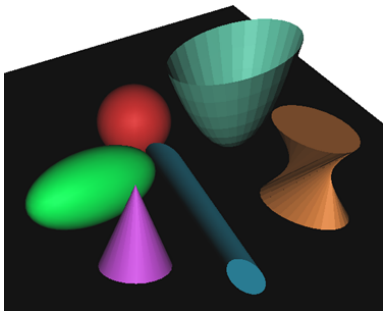


Figure 2.8: Possible quadrics.

Once $\mathbf{Q}$ is given, this type can be determined from an eigenvalue analysis of $\mathbf{Q}$ and its subspaces. Note that quadrics have constant second order derivatives and are practically smooth.

Definition 2.5.2 (Plane Pairs)**.** A quadric whose matrix is of rank 2 consists of all points on a pair of planes: $\mathbf{Q} = \mathbf{\Pi}_1\mathbf{\Pi}_2^T + \mathbf{\Pi}_2\mathbf{\Pi}_1^T$, where $\mathbf{\Pi}_1$ and $\mathbf{\Pi}_2$ are the homogeneous 4-vectors representing the two planes. A quadric whose matrix is of rank one consists of a single plane: $\mathbf{Q} = \mathbf{\Pi}\,\mathbf{\Pi}^T$.

Definition 2.5.3 (Polar Plane)**.** The polar plane $\mathbf{\Pi}$ of a point $\mathbf{x}$ relative to a quadric $\mathbf{Q}$ is $\mathbf{\Pi} = \mathbf{Q}\mathbf{x}$. Reciprocally, $\mathbf{x}$ is called the pole of plane Π. Note that if $\mathbf{Q}\mathbf{x} = \mathbf{0}$, then the polar plane does not exist for $\mathbf{x}$; also note that for a point that lies on the quadric, the polar plane is the tangent plane in that point.

Definition 2.5.4 (Central Quadric)**.** A quadric is called central if it possesses a finite center point $\mathbf{c}$ that is the pole of the plane at infinity: $\mathbf{Q}\mathbf{c} \sim \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$ e.g. ellipsoids, hyperboloids.

Definition 2.5.5 (Dual Quadric)**.** A dual quadric $\mathbf{Q}^* \sim \mathbf{Q}^{-1}$ is the locus of all planes $\{\mathbf{\Pi}_i\}$ satisfying $\mathbf{\Pi}_i^T\mathbf{Q}^{-1}\mathbf{\Pi}_i = 0$. Quadric dual space is formed by the Legendre transformation, mapping points to tangent planes as covectors. Every dual point represents a plane in the primal. Many operations such as fitting can be performed in either of the spaces [71]; or in the primal space using constraints of the dual. The latter forms a mixed primal-dual approach. Note that, knowing a point lies on the surface gives one constraint, and if, in addition, one knows the tangent plane at that point, then one gets two more constraints. This view will help to reduce the minimal point necessity.

2.6 3D Rigid Registration: Iterative Closest Point (ICP)

ICP [28] is an iterative algorithm, which alternates between correspondence search and rigid registration in order to find the best aligning rotation and translation between two point clouds. The basic form of the algorithm first hypothesizes a set of correspondence pairs and then solves for the rigid registration problem relying on these pairs. Correspondences are selected as the nearest point pairs belonging to source and target point sets. ICP is a well studied topic and a vast literature exists

building upon it [42, 65, 231, 302]. Here we briefly summarize the rigid registration, which forms the backbone of ICP.

Definition 2.6.1 (Point-to-Point Metric)**.** Point-to-point error measures the *object space distance*, between the source $P = \{\mathbf{p}_i \in \mathbb{R}^3\}$ and destination $Q = \{\mathbf{q}_i \in \mathbb{R}^3\}$ point clouds.

$$E_{pp} = \sum_{i=1}^{N} \left(\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\right)^2 \tag{2.55}$$

Definition 2.6.2 (Point-to-Plane Minimization)**.** Instead of directly minimizing the object space distance, point-plane error [178] minimizes the distance from point to the tangent plane at the corresponding destination point:

$$E_{pl} = \sum_{i=1}^{N} \left((\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i)^T \mathbf{n}_i\right)^2 \tag{2.56}$$

2.7 Point Pair Features (PPF)

Point Pair Features are anti-symmetric 4D descriptors of a pair of oriented 3D points $\mathbf{x_1} \in \mathbb{R}^3$ and $\mathbf{x_2} \in \mathbb{R}^3$, defined as a map $\mathbf{f}: \mathbb{R}^{12} \mapsto \mathbb{R}^4$ sending two oriented points to three angles and the pair distance [87]:

$$\mathbf{f}: (\mathbf{x}_r, \mathbf{x}_i)^T \mapsto (\angle(\mathbf{n}_r, \mathbf{d}), \angle(\mathbf{n}_i, \mathbf{d}), \angle(\mathbf{n}_r, \mathbf{n}_i), \|\mathbf{d}\|_2)^T \tag{2.57}$$

where $\mathbf{d}$ is the difference vector $\mathbf{d} = \mathbf{p}_r - \mathbf{p}_i$, $\mathbf{n_1}$ and $\mathbf{n_2}$ are the normals at $\mathbf{x_1}$ and $\mathbf{x_2}$. $\|\cdot\|$ is the Euclidean distance and we always compute the angle between two vectors as: $\angle(\mathbf{v_1}, \mathbf{v_2}) = \tan^{-1}\left(\|\mathbf{v_1} \times \mathbf{v_2}\| / \mathbf{v_1} \cdot \mathbf{v_2}\right)$. PPFs can also be used to identify whether a selected point pair lies on a known primitive or not [85]. For instance, two points live on the same plane if:

$$\mathbf{f}(\mathbf{x}_1, \mathbf{x}_2) \in \left[\frac{\pi}{2} - \delta_\triangleleft, \frac{\pi}{2} + \delta_\triangleleft\right] \times \left[\frac{\pi}{2} - \delta_\triangleleft, \frac{\pi}{2} + \delta_\triangleleft\right] \times [0, 2\delta_\triangleleft] \times \mathbb{R} \tag{2.58}$$

where $\delta_\triangleleft$ indicates a tolerance on the angular component. Similar relations can also be derived for spheres or other types of shapes [85].

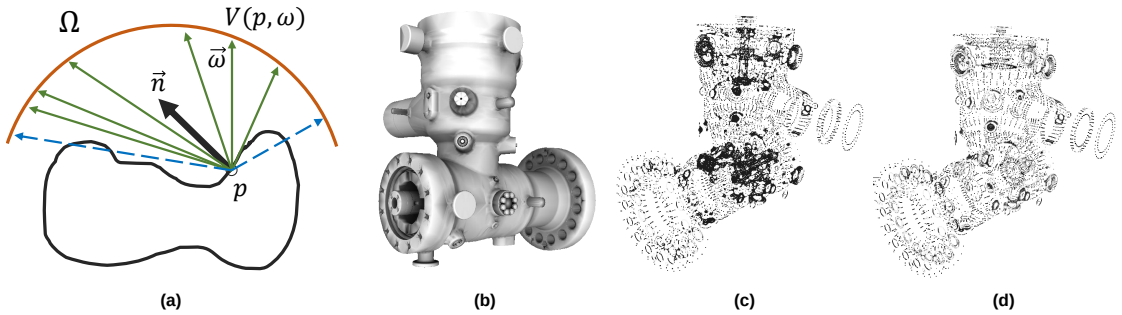


Figure 2.9: Model preprocessing. **(a)** Illustration of ambient occlusion computation. **(b)** The ambient occlusion map visualized on the object. **(c)** Original vertices of the CAD model. **(d)** Vertices prunned via ambient occlusion computations.

2.8 Ambient Occlusion Maps

The occlusion $A_{\mathbf{p}}$ at point $\mathbf{p}$ on a surface with normal $\mathbf{n}$ can be obtained by computing the integral of the visibility function V:

$$A_{\mathbf{p}} = \frac{1}{\pi} \int_{\Omega} V_p(\mathbf{n} \cdot \mathbf{w}) d\mathbf{w} \tag{2.59}$$

where V_p is the visibility function at point $\mathbf{p}$, defined to be 0 if $\mathbf{p}$ is occluded in the viewing direction $\mathbf{w}$ and 1 otherwise, and $d\mathbf{w}$ is the infinitesimal solid angle step of the integration variable Ω. In practice this integral is approximated via rendering the model from several angles and accumulating the visibility per each vertex. The cosine weighted average is then reported as the vertex-wise occlusion value. In order to compute the renderings, we try to choose a projection matrix, resembling the real setting as much as possible. Nevertheless, the correctness of such synthesized matrix is never an issue and very crude values are sufficient. Fig. 2.9a visualizes the computation procedure. Fig 2.1(f) maps the ambient occlusion values to vertex colors for the bunny object. Moreover, Fig. 2.9b depicts a different, industrial 3D model colored by its occlusion map. We then use these occlusion values to prune the vertices, with low visibility, that are either hidden, or are highly unlikely to be captured in real scans - see Fig. 2.9, where Fig. 2.9c shows the original scan and 2.9d shows a case with 30% of the vertices removed.

2.9 Robust Statistics

Many real-life optimization problems are subject to outliers one way or the other. In pure forms of least square estimations, such out-of-model points easily bias the solutions towards undesired minima. Robust statistics provide a simple framework for addressing issues arising from outlying data. Following [123], given a non-linear minimization of the form $E(\mathbf{x}) = \sum_{\mathbf{s}\in\mathbf{S}} d(\mathbf{x},\mathbf{s})$, we introduce a robustifier $\rho(\cdot)$ and write:

$$E(\mathbf{x}) = \sum_{\mathbf{s}\in\mathbf{S}} \rho\big(d(\mathbf{x},\mathbf{s})\big) \tag{2.60}$$

where $\mathbf{s} \in \mathbf{S}$ denote a set of observations, and $\mathbf{x}$ the variables to optimize for. We choose ρ to be Tukey's Biweight M-estimator as:

$$\rho(\hat{r}_i) = \begin{cases} \hat{r}_i(1-(\frac{\hat{r}_i}{c})^2)^2 & \text{if } |\hat{r}_i| \leq c \\ 0 & otherwise \end{cases} \tag{2.61}$$

where $\hat{r}_i = \frac{r_i}{\hat{s}}$ with r_i being the current residual and $c = 4.685$. $\hat{s} = 1.4826\text{mad}(\mathbf{r})$ is the estimated robust standard deviation. $\text{mad}(r_i)$ refers to the median absolute deviations: $\text{mad}(\mathbf{r}) = \text{median}\big(|\mathbf{r} - \text{median}(\mathbf{r})|\big)$. Tukey's M-estimator is aggressive towards outliers and tends to penalize them whenever $\rho(\mathbf{r}) = 0$.

CHAPTER 3

FIDUCIAL TAGS: RELIABLE GROUND TRUTH ACQUISITION

Obtaining accurate ground truth is crucial for assessing any computer vision system. In the specifics of 3D, such ground truth usually takes the form of camera poses and 3D points. Thus, an initial step to take, before delving into the procedures and techniques, is to establish practical ways to acquire such validation data. We now explain the *X-tag* system [34], that is a new fiducial tag design for such purposes.

Figure 3.1: Our markers, can be used in very cluttered scenes.

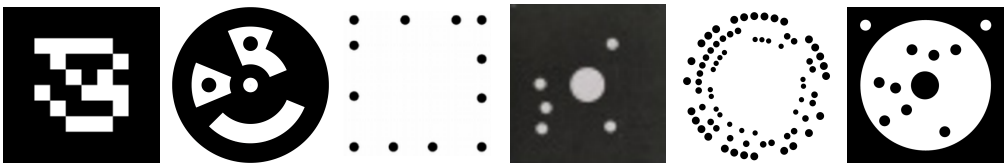
Identification and pose estimation of planar fiducial markers has a long gone history in photogrammetry, augmented reality and computer vision. 2D planar markers, one common form of fiducials, are the primary instruments for obtaining reference coordinates in controlled scenes. They were successful in constraining the algorithms in many tasks such as 3D reconstruction and camera calibration [18]. These simple artificial landmarks can be designed in a task specific way, and can be located with high speed, high repeatability and accuracy, contrary to the natural features. In spite of all the developments in this field (see Fig. 3.2), practitioners still face the problem of mis-detected codes, low true positive rates,

or inaccurate localization of the markers due to various distortions. Moreover, different applications have different demands, requiring custom code designs. Some of the available markers are not fully perspective invariant [279], while the others which have this property either require a good estimate of the intrinsics [24] for getting the marker pose or the detection complexity enormously increases with the increase of their number [25]. In this work, we propose the novel X-tag as a flexible alternative, which enjoys true projective invariance, high accuracy localization and fast identification. In the core of the method, we use a random-dot style marker design, which is described by a set of extended joint projective invariants, composed of multiple cross ratios and intersection preservation constraints. We then use a geometric-hashing framework, as illustrated in Fig. 3.3, to index a set of pre-generated dot positions. Simply, this forms the marker database. The decoding is cast a retrieval problem, in which the same features, extracted from query tags, are matched across the database through an inverted file. The correct matches are subject to further verification using Homography constraints. In contrast to previous works, which are also based on random dot patterns [279], our marker is truly projective invariant and thus is robust to viewpoint changes. This lets us to find more correct tags, enabling more advanced applications such as camera calibration, bundle adjustment and 3D object reconstruction. Due to the adjustable size of the marker, we could design codes which are resilient against radial distortions. Moreover, thanks to the increased number of internal dots, we could obtain more reliable pose estimates, and thus more reliable initialization for procedures such as bundle adjustment.

Our design advances on the good traits of both its ancestors: The square and circular tags. It is easier to detect than circular tags, while being even more accurate than the square counterparts. We apply X-tag to the problem of camera calibration, bundle adjustment and 3D reconstruction, advancing the state of the art.

3.1 Prior Work

Markers enjoy a wide literature in computer vision and augmented reality. While the history is rather unclear, the current simple targets are square markers. They typically contain the description in the inner region of the square as a form of binary code, or a unique image/geometry. AR-Tag [95], Aruco [101], ARToolkit [145] and AprilTag



(a) (b) (c) (d) (e) (f)

Figure 3.2: Markers belonging to different methods. **(a)** AR-Tag, **(b)** Intersense, **(c)** Pi-Tag, **(d)** Linearis, **(e)** Rune-Tag, **(f)** Ours.

[213] are some examples. On the pro side, these targets are very efficient to locate and identify either by correlation methods, or by a binary decoding schemes. However, the use of the squares, rectangles and lines limit the accuracy when detecting subpixel locations on the markers. This makes these markers inapplicable to certain scenarios, requiring high accuracy, such as camera calibration. Moreover, the necessity to spot a quad (collinearity) causes the marker to get affected from the radial distortions and occlusions easily. Thus, some of the aforementioned studies had to explicitly address such issues.

Motivated by the limitations of corner features of the square tags, the next generation fiducial tags made use of circular features, which are more accurate to localize and less sensitive to noise. Intersense [205] combines data-matrix concept with concentric circles to create bar-coded markers. Their design allows generation of 2^{15} codes for identification, but the pose estimation remains to be problematic [145]. Pi-Tag [25] uses a fiducial design composed of ordered circles. The detection benefits from cross-ratio invariants to handle perspective distortions. While, this approach is promising, the matching of cross-ratios is an issue, and the worst-case complexity is reported to be $O(N^4)$, which could quickly become impractical. Inspired by [206], random dots [279] choose to approximate the projectivitiy with affine constraints, resulting in an easier and more stable feature. The authors also devise a geometric hashing [292] framework to cast the code reading problem to a retrieval one. Yet, random dots still exhibit affine features and cannot handle full projectivity. In addendum, due to the frameless design, a large number of dots are required for reliability, increasing the computational load. In the recent state-of-the-art work [24], authors of Pi-Tag take a different standpoint proposing RuneTag, a non-concentric and disconnected

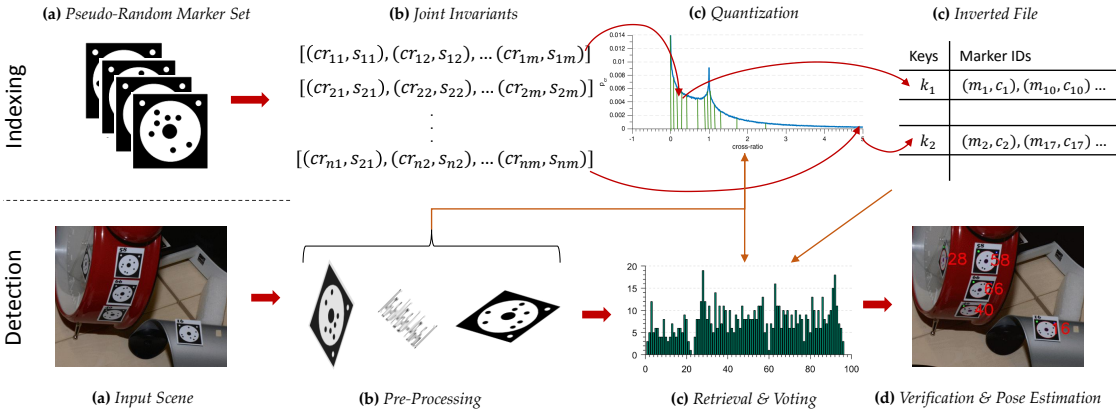
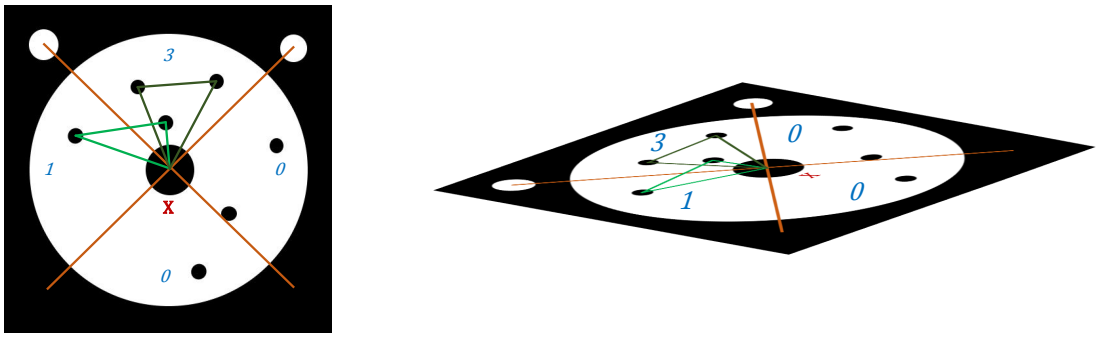


Figure 3.3: Our pipeline is composed of an online and offline stage. In the former, we index the markers using the joint invariants into a single hashtable (inverted file). In the retrieval stage, we query each detected marker-like blob. Each query consists of casting votes for various invariant configurations obtained by five or more points.

arrangement of circular marks around multiple rings, invariant to the projective transformations. This reduces the burden imposed by the feature extraction. The result is a very robust and occlusion tolerant fiducial marker, being reasonably fast to detect. A common point in all three designs of [24, 25, 279] is the fact that the tags are composed of individual circles, which link to form the whole. While this eases the processing stage, introduction of clutter, especially in the form of false ellipses causes the runtime to significantly increase, if not fail the detection completely. Another observation is that, many of the codes are designed to be large and redundant, i.e. close-by placement of individual ellipses are prone to merge under camera noise or blur, especially in distant views. This is not desired for applications targeting camera calibration, as it is important to distribute as many markers in 3D space as possible. The circular fiducials are also the method of choice, when implementing photogrammetry systems. Linearis [1], Aicon3d [5] or GOM TriTop [109] are some of those end-to-end measurement systems. The exact algorithms used in such products are not publicly available and hidden. Yet, we are aware, for example, that Linearis cannot handle large perspective distortions.



(a) (b)

Figure 3.4: **(a)** Cross ratio and sector information visualized on an X-tag. **(b)** Both features are preserved under projective transformations.

3.2 X-tag

Design and description X-tag consists of a random arrangement of several (> 5) black circular marks, distributed around a black central dot and on a white background. The design includes an additional black frame, acting as a contrast agent to ease the localization. The actual shape of the frame is irrelevant and can also be designed to be circular. In addendum, two white circles are placed on one side of the base frame. They are used for multiple purposes of feature extraction, verification and pose estimation. The marker is shown in Fig. 3.4. We describe our X-tag via a new *extended invariant set* (EIS). EIS is composed of two parts: The cross ratios and sector information. The former is the set of all cross ratios that can be computed from the inner dots, taking the central dot as the 5^{th} point. Fixing such a point increases the stability as opposed to complete randomness [279]. To further boost the discrimination power, we introduce the latter descriptor, *sector type*, which relies on the preservation of the intersection of lines. Formally, we partition our fudicial into 4 regions. 4 partitions are given birth by the intersection of two lines formed by joining the outer white dots, with the center point. These lines also separate the large inner white circle into 4 partitions. For each CR computation, i.e. for each 4 points taken out of the randomly generated points, our descriptor encodes the presence of the points within the sector. This can be seen in Fig. 3.4. In particular, the sector type $s = \{x \in \mathbb{R} : 0 \leq x \leq 500\}$ is computed as the polynomial expansion $s(p_1, p_2, p_3, p_4) = s_1\alpha^3 + s_2\alpha^2 + s_3\alpha + s_4$, where s_i is the number

of points out of $\{p_1, p_2, p_3, p_4\}$, lying within sector i. For our configuration we set $\alpha = 5$. Note that due to the intersections being preserved, this is a true projective invariant. Our final descriptor is the concatenation of the two distinct invariants described here, forming the set $\mathbf{D} = \{\mathbf{d}^i = \{s^i, cr^i_{2D}\}\}$.

Tag localization The first step in the pipeline is the low level image processing, in which we identify and localize the X-tag candidates. To do so, we apply a simple image processing algorithm. As image processing is not the scope here, we give only a brief overview: First, the dark regions are selected as marker candidates. Then, a connected component labeling discards the blobs, which do not satisfy a relaxed set of constraints (area and dimensions). Later, each candidate blob is tested for inclusion of a light (e.g. white) region and sufficient circular points. We also check for the two white dots, and the center dot, explicitly. At each step of this operation, elliptical regions are selected via the properties of the elliptic axis. Note that, survival of false positives at this stage are likely to be suppressed in subsequent retrieval and verification.

Indexing markers and database creation For each marker, we obtain $\mathbf{D}_i$, a long, extended descriptor. Indexing such a descriptor for nearest neighbor retrieval purposes is not always trivial. Here, we explain how we benefit from the special nature of our descriptors to use them in a geometric hashing framework.

The basic idea behind our algorithm is that we quantize the descriptor for each marker sequentially and store the quantized codes in an inverted file, along with the occurrence information c_i and the marker id m_i. c_i also helps us to compress the inverted file, as multiple occurrences of the same marker are stored as a single entry in the hashtable. The *Indexing* part of Fig. 3.3 illustrates this scheme.

Because the probability distribution of cross ratios is highly non-linear, a simple uniform quantization of the features wouldn't work, i.e., many cross ratios would fall in the same bin. Thus, we rely on a quantization scheme, which is aware of the joint feature distribution (see § 2.4.1). Our essential idea is to create a binning such that the integral of PDF in each bin is roughly equal. Formally, let f denote the PDF, F the CDF and F^{-1} the inverse CDF of cross ratios. Because we now (or can) estimate both F and F^{-1}, we choose to map any given cross ratio cr via CDF and to perform a uniform

quantization in this domain. Formally a $b-bit$ quantized value $\mathbf{f}_q[cr]$ is obtained by:

$$\mathbf{f}_q[cr] = b\left\lfloor \frac{F(cr)+E_q[cr]}{b} \right\rfloor, \quad E_q[cr] \sim U\big([-\frac{\delta}{2},\frac{\delta}{2}]\big) \tag{3.1}$$

$E_q[n]$ is uniformly distributed, due to the assumption that the errors are uniformly spread into the bins. In our implementation, we prefer to use the approximate CDF F^*, instead of the analytical F as F^* is a better representative of the data-subset. Such quantization requires a look-up over the CDF, which we perform via binary-search. Faster implementations might benefit also from interpolation search, as the distribution is available. By quantizing directly on F, we could avoid using F^{-1} to map back to the PDF, f. However, Fig. 3.3 plots the partitions in the PDF domain.

Contrary to cross ratios, the sector type is a simple uniformly distributed integer and is very friendly for indexing operations. Our hash index is simply $h_{cr} = \{\mathbf{f}_q[cr], s_{cr}\}$.

Identification of tag IDs Once the features are extracted for all combinations of points $\{\mathbf{p}_i\}$ in a candidate scene, we could resolve the tag id using the inverted file. To avoid the distance computation overhead and to retain the robustness, we achieve this through a procedure, similar to Hough voting. Each quantized feature h_{cr} retrieves a set of probable markers from corresponding bucket and casts a vote to the corresponding marker id. The vote is proportional to the occurrence in the database. Ideally, after voting for all joint invariants, the maximum vote reveals the marker ID.

Verification and pose estimation Even though the voting is very robust, it doesn't always guarantee the best solution. For that reason, we retain a set of surviving hypotheses for further verification. Moreover, the match-ability of the marker necessitates the correct identification of only three points: The center and two support points. This leaves us with one unknown to determine the projective transformation. Note that, using conics for pose estimation might be bad in this situation because it is very likely that a single ellipse would appear as a small dot.

To find the ID of the 4^{th} point, we could simply enumerate over all the possible point combinations and evaluate the reprojection error, but to save computation, we instead apply similar voting procedure as we use for matching of marker IDs. For each dot in marker cross-ratios with all other points are calculated and stored in the hashtable. On the verification stage the voting for dot ID is performed similarly

to voting for the marker ID using all cross ratios for given dot. Resolving the correspondences finally becomes more efficient since we verify the best hypothesis first. Formally, the fourth landmark is found via:

$$\mathbf{p}^* = \arg\min_{\mathbf{p}} \sum_{j=1}^{n} \|\mathbf{H}(\mathbf{p})\mathbf{m}_j - \mathbf{r}_j\| \tag{3.2}$$

where $\mathbf{H}(\mathbf{p})$ is a homography found after matching reference point to point $\mathbf{p}$ via voting procedure, $\mathbf{m}_{1..n}$ are dots locations inside marker, $\mathbf{r}_{j..n}$ are their correspondences in reference frame. Under occlusions or noise, a one-to-one correspondence is enforced for robustness. $\mathbf{H}$ is computed from 4 points using DLT algorithm [123]. The final pose is estimated using PnP algorithm [164] using all dot coordinates in marker. This is superior to standard square tags in two aspects: 1. The used dots are circular and are more accurate to localize. 2. We have always $N_d > 4$ dots in our marker. As we utilize all the found dots, our estimation is expected to be more correct.

Multi-camera bundle adjustment A useful application area for X-tag is camera calibration and bundle adjustment[272]. We propose to use X-tag as a calibration target and compute the extrinsics and intrinsics with BA. Our idea is to make the user entirely free from the using precise targets. We rather rely on the central ellipse of X-tag to give us the image cue. Our approach is similar to [73], but we do not constrain ourselves to planar targets. Given a set of images, captured either from moving cameras, or changing scenes, we run the following optimization:

$$\min_{\mathbf{\Pi},\mathbf{X}} \sum_{i=1}^{m} \sum_{j=1}^{n} \rho\left(w_{ij} d(\mathbf{\Pi}_i \mathbf{X}_j, \mathbf{x}_{ij})^2\right) + \sum_{i=1}^{k} \sum_{j=1}^{k} (d(\mathbf{X}_i, \mathbf{X}_j) - \sigma_{ij})^2$$

where $\mathbf{X}_1..\mathbf{X}_n$ are 3D points, $\mathbf{\Pi}_1..\mathbf{\Pi}_m$ are projection matrices of m cameras, $\mathbf{x}_{ij}$ is image coordinate of point j for camera i. The distance $d(\cdot)$ between any two points is subject to a weighting w_{ij} which based on the detection quality of image points. $\rho(\cdot)$ is a robust Cauchy norm. We compute its scale parameter from the elliptic axis properties. The second term is regularization that brings the reconstructed scene to metric space by keeping distance between known 3D points $(\mathbf{X}_i, \mathbf{X}_j)$ at the value σ_{ij}. We initialize this BA procedure from the pose of the most frequently visible marker. The pose is estimated using the inner random dot locations, w.r.t. the canonical marker frame. In BA, we simultaneously solve for $(\mathbf{\Pi}, \mathbf{X})$, using Brown distortion model [49].

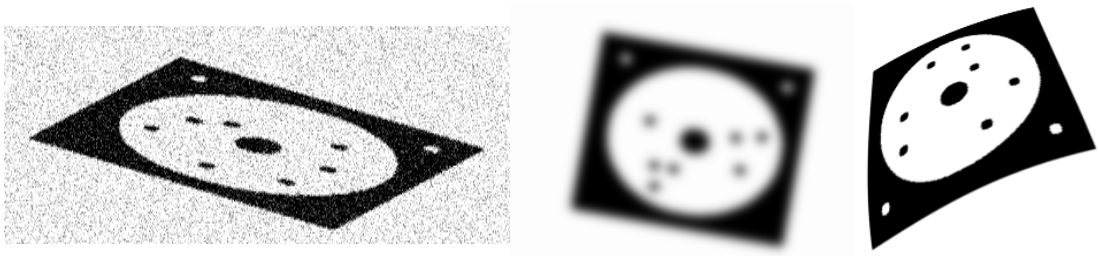


Figure 3.5: Shots from synthetic scenes for different noise, blur and radial distortion conditions.

3.3 Experimental Evaluation

We assess the performance of our method with extensive qualitative and quantitative evaluations. We first define the evaluation metrics, then perform ablation studies on synthetic data and finally, apply X-tag to multiple applications utilizing real datasets. In this section, our evaluation metric is based upon the individual errors of the distinct pose components (rotations and translations).

3.3.1 Experiments on Synthetic Data

We first evaluate the validity and robustness of our approach on a synthetic set. This way, we observe the performance under various degradation and capture the behavior of parameters. For this stage, our synthetic data is composed of $N_M = 2000$ markers. For testing, we sample 200 of this set and combine it with 20 other markers, which are outside of the database. The test data is subject to 50 warps per image, each having a different augmentation. These augmentations include blur, additive noise and radial distortion. The synthesized images are shown in Fig. 3.5.

Effect of hashtable size As the initial stage of experimentation, we would like to tune our system to use the optimal parameters. We asses how the performance, as well as the computation time is affected with the varying number of markers, number of bins and number of dots. Therefore, we conduct incremental evaluations. First, we want to find the optimal size of a hash table for the marker set. Ideally, the true marker ID should get the most votes from the hash table. So by tuning the hash table size we

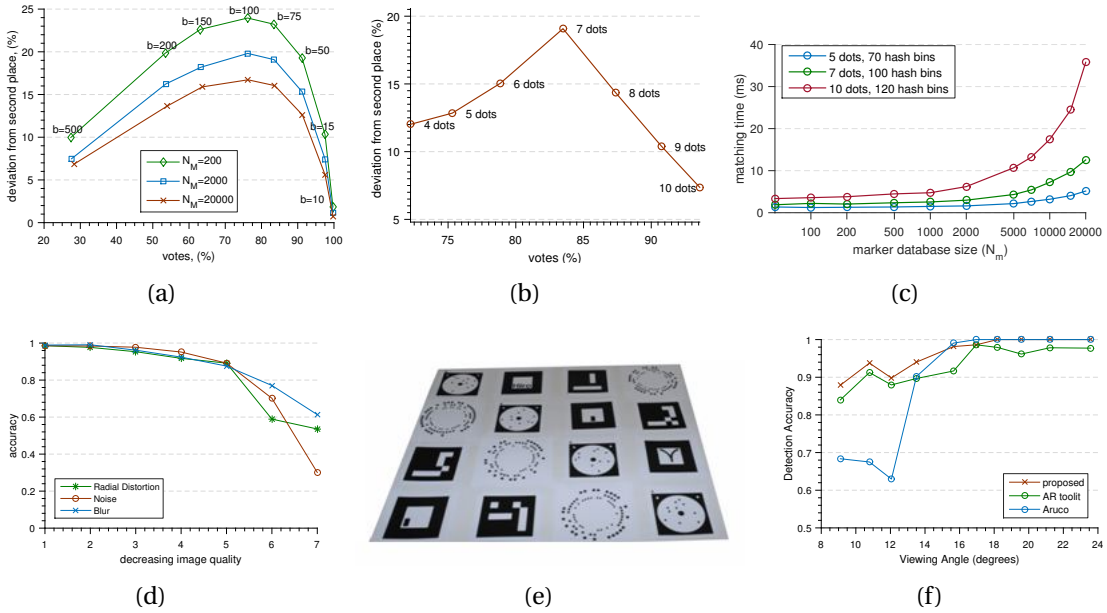
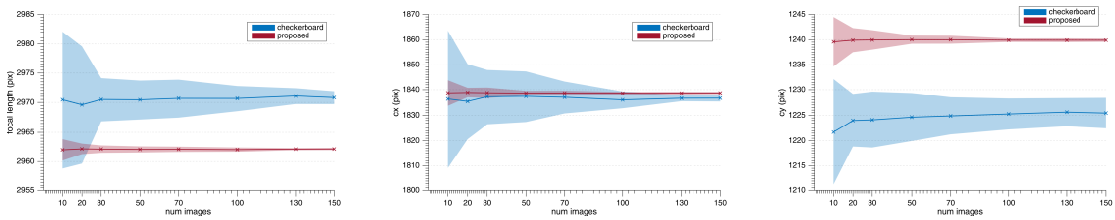


Figure 3.6: **(a)** Maximum percentage of votes and deviation from next best place in voting table for different sizes of marker databases **(b)** Votes for markers with different number of dots using hashtable with 100 bins **(c)** Matching time for one marker using optimal size of hash table **(d)** Marker detection rate on synthetic scenes

aim for the maximum percentage of votes for marker with the highest rank and largest deviation from the second best. While the desired number of dots is a parameter for our method, within the context of experiments, we fix it to 7. On our synthetic set, we conduct the aforementioned performance analysis and plot this in Fig. 3.6(a). It is visible that independent of the database size the optimal number of bins for markers with 7 dots is 100. It is interesting to see that only the number of cross ratios for one marker influences the optimal hash table size.

In a further experiment in Fig. 3.6(b), we fix the number of markers in our database to 2000 and also the hashtable bin sizes to 100. By varying the number of dots we could see that having 100 bins for markers with 7 dots will provide optimal voting results for that configuration. It is therefore immediate that when more dots are desired, the hashtable size should be tuned accordingly.

Next we evaluated matching time for markers with optimal hash table size. The time of matching depends both on number of cross ratios for one marker and the



(a) (b) (c)

Figure 3.7: **(a)** Images for checkerboard calibration and calibration with X-tag. **(b)** Focal length estimation **(c, d)** Estimation of principal points (x, y), respectively.

number of markers in database. Matching one marker takes $O(n\log(b) + bN)$ time in worst case, where n is number of cross-ratios, b - number of bins, and N - size of markers database. The matching time for markers with 5, 7 and 10 dots for Intel i7 3.20 GHz processor is shown on Fig. 3.6(c). In practice matching a single marker takes less than $3ms$, given that in real life, a database of 2000 markers is more than sufficient. Note that, thanks to the grouped inverted file structure and the quick voting, our computational time only marginally increases even when the database size is significantly increased. In that manner, our approach is very scalable and therefore amenable for real life applications requiring an abundance of fudicial tags.

Robustness to image distortions We assess the robustness (as detection accuracy) to different noise and perturbations and the computational performance in Fig. 3.6(d). The x-axis shows the image quality, which is gradually reduced by different augmentations. For real-life scenarios, quality level hardly exceeds 3. It is evident that while X-tag is generally robust, it is least affected by the blur. Increasing noise would have the most severe effect, while radial distortion is handled moderately.

3.3.2 Real Scenarios

Robustness to viewing angle To quantify the robustness under perspective changes, we print 4 of each RuneTag[24], Aruco[101], ARToolkit[145] and X-tag on a common paper as shown in Fig. 3.6(e). We image 4 different rotations of this pattern in varying distances and from severe (8°) to moderate (25°) camera angles. We then run each detector and compute the overall detection rate. The results are plotted in Fig. 3.6(f).

It is shown that while most methods perform very well, ours is slightly above all others, justifying the good detectability. RuneTag [24] and RandomDots [279] are not taken into this plot because: 1. RuneTag circles quickly get invisible with the increasing distance and tag starts to under-perform. 2. RandomDots are only robust up to affine warps and cannot handle perspective variations. Thus, the detection rate appears to be low for this experiment. While the square fiducials are known the best for this type of challenge, our circular tag still outperforms the rest.

How reliable is the estimation of intrinsics? As explained in §3.2, our method is suitable for complete bundle adjustment, where the intrinsics and extrinsics are jointly minimized. While it has always been a challenge to assess the accuracy of calibration (as exact principal point and focal length are not directly observable), we argue that, it is more important to obtain repeatable estimates, rather than accurate ones i.e. one could always use an offset to compensate for biases, once the repeatibility is achieved. We, therefore, use a slightly unorthodox experimentation and run our bundle adjustment multiple times for intrinsics estimation, repeatedly. We perform the same test with a OpenCV checkerboard calibration [45], which seems to be the de-facto standard in computer vision. The number of detected corners roughly equate to the number of detected ellipse centers. Fig. 3.7 plots our estimations along with the ones from checkerboard for principal point, as well as focal length. The standard deviation overlays the curve. It is apparent that our results are more deterministic and less prone to initialization errors, as well as errors in feature point computation. The deviation plots indicate that even with small number of images our estimations are more reliable than the standard techniques. Note that, an analogous experiment shows a similarity between OpenCV's checkerboard method and RuneTag calibration [24]. It is also worth mentioning that while both OpenCV and RuneTag rely on the availability of the 3D model of a calibration pattern, we are completely pattern-free and our markers could be positioned anywhere in the observed space.

Evaluation of pose estimation Here, we evaluate the power of a single tag for estimating extrinsic pose. For that, we set up a scene of 80 markers composed of 40 Aruco and 40 ours as shown on Fig. 3.9. This scene is then viewed from 100 distinct camera locations, including viewpoint variations. Afterwards, we run our bundle adjustment

Figure 3.8

Figure 3.9

Table 3.1

	Aruco	Ours
Rot Err	0.0216	0.0145
Tra Err	0.0091	0.0067
Repr Err	0.2472	0.0694

Figure 3.10: **(left)** Scene setup for intrinsic calibration. **(middle)** A screen-shot of a setup for extrinsic pose estimation. **(right)** Errors for the bundle adjustment scenario of the image in the middle.

proposed in §3.2 on Aruco markers and our markers separately and multiple times. We always initialize the adjustments by using the pose of one of the markers selected as a reference. We deliberately alter the selected reference over different BA runs to reduce the selection bias. BA procedure corrects for 3D locations as well as camera poses. Finally, the refined pose of the selected reference tag is compared against the initial estimation, both for Aruco and ours, disjointly. The difference in these poses is naturally the computed update by BA. The smaller the update is, the more correct the initialization, and therefore the better the estimation of extrinsics from a single marker. The results, averaged over a set of runs, are shown in Tab. 3.1 for the scene in Fig. 3.9. The findings indicate that our markers are much better at providing camera pose than Aruco. The reason why the reprojection error enjoys a relatively higher improvement is because it absorbs both the errors on the pose and on the 3D structure. An improvement of both increases the impact on the reprojection.

Object reconstruction performance At last, we evaluate our method for the problem of 3D object reconstruction using depth sensors. Our procedure is similar to KinectFusion [208], however, we replace the ICP (iterative closest point) [28] stage with poses coming from X-tag detection, and perform the conventional SDF-fusion [72]. This way, the object is not needed to be tracked and we could operate with only a handful of scans. Our setup consists of *Teddy* object, which is a 3D print from an ideal CAD model. The object is positioned on a turn-table sequence. The tags are distributed around different regions of the space. Because the state of the art fiducial tag for object reconstruction is RuneTag [24], we evaluate only against this method.

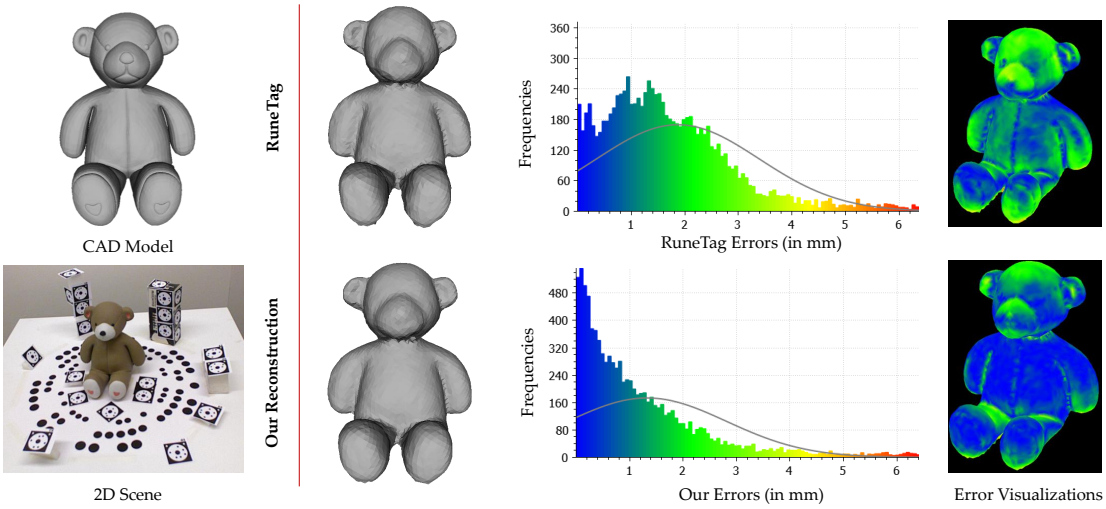


Figure 3.11: Assessing RuneTag and X-tag in reconstructing *Teddy*. Qualitatively and quantitatively X-tag leads to a better reconstruction.

Thus, we augment the scene with RuneTag marks. A shot from this setup can be viewed in the first column of Fig. 3.11. Following the sequential image acquisition, we then run our bundle adjustment both for our tags and for RuneTag. Note once again that, X-tag BA assumes neither camera calibration nor an apriori 3D model, while RuneTag is designed to operate best on calibrated settings (While RuneTag could also handle uncalibrated case, this capability depends on an enumeration over all possible focal lengths until a reasonable estimate is found. We consider this to be still calibration dependent.). Therefore we initialize RuneTag with the correct bundle adjusted intrinsics and let it estimate only the camera poses. BA output provides us both refined poses and point coordinates. In this stage, we use only the poses and discard the 3D structure. We retrieve this structure from the depth images of the 3D scanner.

We convert the absolute poses to the relative ones and starting from an initial volume, we run an SDF Fusion to capture the final 3D reconstruction. Thanks to the presence of the ideal model, we compare both results to the ground truth. These comparisons are depicted in Fig. 3.11. The colors are associated to the unsigned error magnitudes. Because our markers are located on non-coplanar regions of the space, they are better at binding the 3D transformation. This demonstrates that, better geometric constraints are more favorable than the availability of prior calibration

targets. In the end, our bundle adjusted 3D points can act as *calibration rigs*.

3.4 Discussion and Use in Sparse Reconstruction

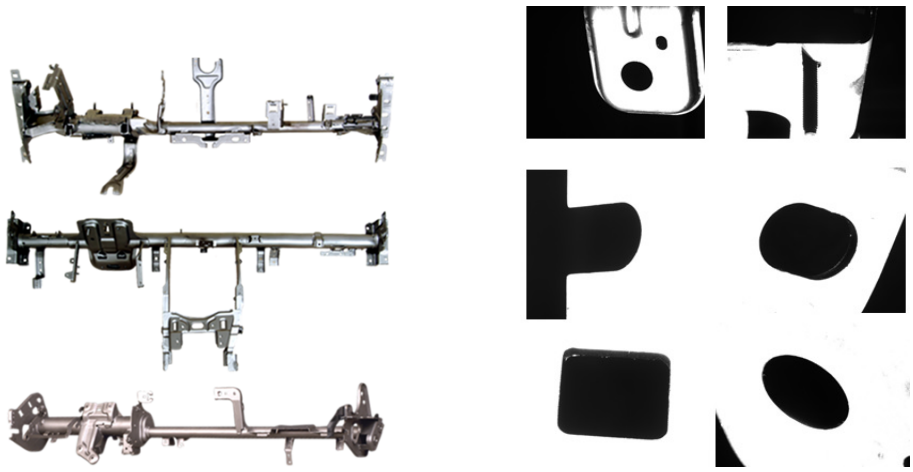
X-tag is a flexible tag amenable for model-free calibration, pose estimation and 3D reconstruction. It is truly invariant to projective changes, detectable in high clutter with reasonable robustness to radial distortions. Moreover the matching time of a single marker is extremely fast and the devised method is suitable to scaling large marker sets.

All the aforementioned attributes motivate us to use X-tag like markers in collecting ground truth poses and obtaining sparse surface reconstruction. These would be valuable in 1. Calibrating non-overlapping cameras by creating marker fields. 2. Measuring sparse keypoints in 3D scenes for ground truth capture in difficult scenarios. In the next chapter, we will give an application where the problem of online, real-time 3D coordinate measurement of manufactured parts is considered.

CHAPTER 4

SPARSE 3D RECONSTRUCTION FOR ONLINE COORDINATE MEASUREMENT

Today, every end product, present in our daily lives goes through an intense manufacturing process. Under the hood, significant effort is put on delivering products without flaws, which meet the desired standards. Dimensional monitoring has become an important part of these manufacturing processes due to the gradually advancing standards in many sectors, such as automotive (chasis), flight (wings) or energy (turbines). The welded components found on the backbone of all these high-end products are sensitive and require careful manufacturing and thus, careful quality control. The de-facto solution to inspect these critical parts relies on the contact-based CMMs (coordinate measuring machines), which are expensive, hard to maintain and slow to operate. These machines can neither be installed on the production lines nor provide 100% statistics on the manufactured parts. Such lack of online inspection coerces many assembly lines to revert to mechanical fixtures and apparatus to verify the quality of manufacturing. This has many drawbacks. First of all, notwithstanding the cost of assembling mechanical control fixtures, sustaining the precision of such assembly over the long term requires significant amount of maintenance effort. Yet, no information on errors or statistics could be provided by the fixed control equipment and the inspections cannot be documented. Last but not least, such inspection necessitates



(a) (b)

Figure 4.1: **(a)** Sample parts of interest in this work. **(b)** Images captured by the setup.

an operator constantly engaging in the production process.

The bottleneck caused by the current quality control procedures is addressed by Tuominen as the lack of reliable, accurate and generic metrology / photogrammetry techniques, which are amenable to be installed on production lines [276]. Tuominen further shows that the ultimate quality control performance is achieved when the parts are either fully inspected or no inspection is carried out at all. This makes CMMs an unviable option. While the industry suffers such drawbacks, the quality requirements imposed by the consumer market keep demanding more and more, stimulating the research on online 3D inspection [179].

This chapter presents a thorough methodology on the research and development of an inspection unit, which addresses all of the aforementioned downsides. Our system is composed of multiple static cameras, calibrated to a global coordinate frame, being able to measure arbitrary geometrical structures. Our algorithms are novel and carefully designed to fit the accuracy requirements. The resulting system is accurate within acceptable error bounds and flexible, in the sense that the design can be adopted to the inspection of different parts with minimal modifications to the software. The entire implementation cost is lower than a CMM, while the capability of inspection is 100%, saving us from the sampling requirement. We present quantitative comparisons with the industry-strength commercial metrology systems

by evaluating our approach against an optical metrology unit (ATOS) and a standard CMM (Hexagon) both for calibration and measurement. We also present our repeatability values along with the actual measurements. Finally, we demonstrate that our timings clearly outperform the state of the art, making the system applicable to online inspection. See Fig. 4.1(a) for the parts we can inspect as well as the images of the measured geometries.

The contributions of this chapter are three-fold: First, we propose an affordable and effective way to calibrate a muli-camera systetm, which is composed of local and global camera networks. We base our calibration on bundle adjustment and global registration of fiducial tags, as explained in Chapter 3. Secondly, we develop a novel, robust, multi-view projective CAD registration procedure along with a multiview edge detection scheme. We show that this new approach is real-time capable, while not compromising accuracy. We review in detail the implementation aspects and draw a complete picture in the design and realization of such a multi-view metrology unit. Finally, we carry out extensive experimental evaluation, in comparisons to existing, state of the art metrology methodologies.

4.1 Prior Work

Optical metrology enjoys a history of over 30 years. The different aspects of the issue such as calibration, triangulation, stereo reconstruction or pose estimation have been tackled many times in computer vision literature.

Analyzing the colinearity relations, photogrammetrists were the first to use image data to conduct measurements [177, 241]. Computer vision tried to improve the lower level sub-problems such as calibration [275, 311], pose estimation [237, 265] or SLAM [19]. In spite of the vast amount of literature in computer vision, the care for accuracy and precision is not very well established in such works.

Many commercial metrology software exist with different application areas [1][2]. While being highly accurate, some of these software are not capable of measuring generic CAD geometries, but rather rely either on feature points or pre-defined markers. In contrast, the tools which could recover arbitrary 3D geometries cannot operate

[1] http://www.photomodeler.com/

[2] http://www.gom.com/3d-software/atos-professional.html

on generic prior 3D models. Even today, a well described, truly capable machine vision technique to automate the process control remains to be intact [179].

Unlike the commercial arena, development of accurate and reliable close range machine vision systems is rather unexplored in academia [179]. Jyrkinen et al [141] designed a system to inspect sheet metal parts, but their approach cannot be generalized to arbitrary geometries. Mostofi *et al.*[201] target a similar problem like ours. Their technique is dependent on human intervention and utilize a single moving camera. Such choice is far from realtime concerns. Yet, authors report the visibility of measurement points from 6 cameras, which drastically increases the number of views and the effort for measurement. Moreover, they report an accuracy of 0.5mm even when the measurement points are visible in 6 views. Such accuracy is well below what is achievable today [280]. Bergamasco *et al.*[26] developed a novel method to precisely locate ellipses in images, but their method involves perfect overlap of views and doesn't generalize to measurement of arbitrary 3D shapes. Similar to our work, Malassiotis and Strintzis developed a stereo system to measure holes defined by CAD geometries [184]. While posing the CAD fitting problem as an optimization procedure, just like ours, they make use of explicit primitive modeling, which restricts the measurement capabilities. The 3D primitives are re-generated at each step of the optimization and this comes at the expense of computational complexity, degrading the real-time (or online) capabilities. Their approach is also not applicable to triangulate arbitrary 3D geometries. Moreover, all of these approaches assume a perfect calibration and lack a well established methodology to calibrate the camera networks.

Our system is uniquely positioned in the application field of non-contact multi-view measurement, which was shown to be one of the most accurate metrology methods. It is tuned for inspection of points of interest [180] and retrieves its power from the developments in low level and geometric 3d computer vision e.g. [28, 123].

4.2 Proposed Approach

System setup Our system consists of 48 *The Imaging Source* cameras providing 1280x960 pixels at 30 fps. Depending on the installation distance, we choose either 25mm, 35mm or 50mm industrial lenses. The scene is illuminated via 10 white global LED lights, installed on the external skeleton. The cameras are positioned such that

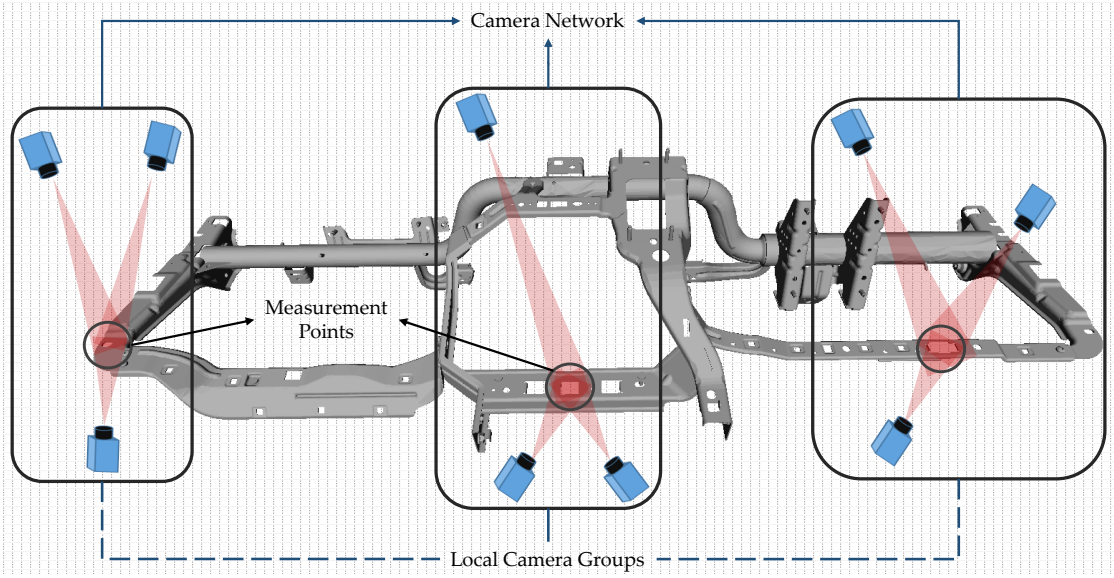


Figure 4.2: System setup. Figure depicts the locally overlapping camera groups forming a global camera network. The shown CAD model belongs to the actual part.

a point of interest is visible to at least 3 cameras. We will refer to these points of interests as the *measurement points* $\mathbf{P}_i \in \mathbb{R}^3$ and to such a local camera group as the *local camera network* $\mathbf{C}^i = \{[\mathbf{C}^i_1, \mathbf{C}^i_2, ..., \mathbf{C}^i_k] : k < N\}$. For the sake of accuracy, it is highly unlikely that a single camera would capture different measurement points and thus the local camera groups remain independent (no overlap and thus no direct calibration exists between camera groups). The set of these K local camera groups form the *global camera network* $\mathbf{C} = \{\mathbf{C}^1, \mathbf{C}^2, ..., \mathbf{C}^K\}$, in total composed of N cameras. Eventually, our system consists of multiple local camera networks, which are calibrated to form $\mathbf{C}$, the global camera network. In this sense, the entire network is arranged in a hierarchy, as shown in Fig. 4.2.

Target parts For demonstration purposes, and specific to our application, we aim to inspect 15 critical points ($\mathbf{P}_i$) on a $2m$ long and $80cm$ wide metal industrial automotive chasis. The part is shown in Fig. 4.2. Even though our setup is capable of inspecting arbitrary CAD models from different manufacturing areas, for the purpose of clarity, we will stick to the automotive application. Typically, the geometries to be measured, as well as the CAD models are known beforehand and the accuracy is limited to 1-1.5%.

Image acquisition We control all cameras over a GigE Network ($1000 Mbit/s$). The lights are strobed to be in synch with the exposure. This prevents the multi-threaded capturing. We initialize each local network group per each measurement, and utilize the full Gig-E bandwidth (using 30fps/camera). This optimizes the speed and durability. Some images of the measured points, acquired by our system during a single runtime are shown in Fig. 4.1(b).

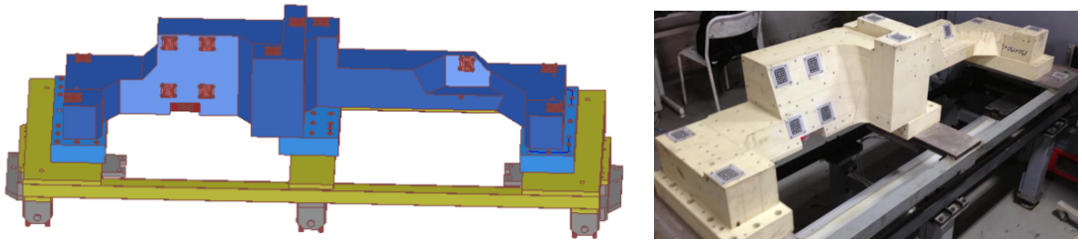
4.2.1 Calibration

We calibrate all cameras intrinsically and extrinsically to the same global coordinate frame. Intrinsic calibration is carried out prior to installation using the standard methods [43, 125]. We use calibration plates made up of 3mm circular reflective dot stains. On the other hand, considering the extremely non-overlapping nature of our cameras, and narrow depth of fields, the task of extrinsic calibration is significantly difficult and crucial.

Extrinsic calibration We use a pre-manufactured calibration object (reference body), *The Calibrator*, to calibrate the absolute poses. Counter intuitively, this object is not precisely manufactured and does not use expensive material or components, but only acts as a mounting apparatus for circular dots, forming the relation over different camera groups. It is produced at a CNC machine such that when it is imaged by the multiview system, sufficient 3D reference points (calibration dots) would be visible on each camera. The shape resembles the rough approximation of the part to be measured. We then manually attach random circular markers so that enough overlap is created. The calibrator (virtually and physically) is shown in Fig. 4.3.

The calibration grids and random dots attached on the calibrator are first reconstructed and bundle adjusted by taking multiple ($\sim$ 128) overlapping shots with a high-resolution SLR camera. This is done with a system similar to the one described in Chapter 3. The optimization is solved with Google Ceres [4].

After creating The Calibrator, we extrinsically calibrate the cameras by imaging the calibration plates multiple times for each view. During this repetitive stage, at each inspection, the positions of the calibration plates are computed and subsequent measurement errors are recorded. Once enough measurements are collected and



(a) (b)

Figure 4.3: The Calibrator **(a)** 3D model. **(b)** Manufactured. Sparse random dots (inter-reference points) exist in the real object, whereas they are absent in the model. They do not contribute to the extrinsic calibration, but serve as a guide for the optimization of the global reference frame. On the target part, instead of the calibration plates, lies the measurement points.

repeatability values become acceptable, the multiview calibration follows. This way, we make sure that enough variance of positioning is covered. Such extrinsic calibration is performed as a succeeding offline bundle adjustment, where the poses and 3D structure are refined simultaneously [126, 142].

Learning systematic errors Finally, the so-far neglected non-rigidness effects, mis-manufacturing and other systematic errors are compensated. To achieve that, we employ an error correction (bias reduction) procedure, in principle similar to [127, 219, 236]. While these methods were designed to perform on CMMs, we generalize the technology to optical inspection via a learning approach. Initially, we sample a set of carefully-manufactured 3D parts. We measure each of those on various CMM devices, repeatedly (5 times for each) to cancel out the biases. If the desired precision is verified, the median measurement per point is taken as the ground truth. We then measure the same parts with our system, repetitively, using the technique described in § 4.2.2. Based on the differences (errors) of two methodologies (CMM and ours), we train a Support Vector Machine [70] for regression. Our features are the concatenated $3D$ point coordinates. The responses (outputs) are then the differences of these measured coordinates to the collected training data i.e. coordinate-wise regressed offset from ground truth. Thanks to the acceptable mechanical precision, we could easily cover almost all the entire space of 3D variations.

4.2.2 Measurement

Our measurement stage follows 4 steps: Acquisition, extraction of geometries of interest, fitting 3D models and triangulation. The part is positioned with a linear axis conveyor belt, giving us a good initial pose. This section will focus on extracting region of interests and fitting CAD models. Note that, we are interested only in measuring the 3D center coordinates, but not the dimensions. Following the determination of 2D spatial interest points, we obtain the real 3D coordinates by triangulation.

4.2.2.1 Geometry Extraction

As the goal is to measure both the geometric primitives and arbitrary edges, we design a generic geometry extractor, which simultaneously determines the edges of interest in multiple views. We will refer to every connected edge fragment, which cannot be represented by a single geometric primitive, as an *arbitrary contour*. With arbitrary contours, one cannot utilize prior information of geometry. We rather rely directly on the edges to obtain the matching structures across multiple views. Our procedure starts with a spatial sub-pixel accurate edge detection performed individually in all views of a local camera network, using a 3^{rd} order contour extraction technique [258]. The resulting edges are smooth, continuous and accurate up to $1/20^{th}$ pixel. Due to edge detection in individual frames, we do not have a representation of correspondence information. In general, obtaining correspondences across multiple images involve stereo matching algorithms, which are computationally expensive. Luckily, our system is calibrated and we are not interested in exact correspondences but rather in sets of consistent points across views (correspondence candidacy). In other words, we seek to find a set of segmented edges per each view. The real correspondences are then generated through the fitting algorithm in § 4.2.3.

Multi-view edge segmentation Given a multiview setup, the essential matrix, relating view i to view j can be computed by $\mathbf{E}_{ij} = [\mathbf{t}_{ij}]_x \mathbf{R}_{ij}$, where $\mathbf{R}_{ij}$ is the relative rotation and $[\mathbf{t}_{ij}]_x$ is the relative translation between views i and j. Then $\forall \mathbf{p}^i, \exists \mathbf{l}^{i->j}_{\mathbf{p}^i_k}$: $\mathbf{l}^{i->j}_{\mathbf{p}^i_k} = \mathbf{E}_{ij}\mathbf{p}^i_k$, where $\mathbf{l}^{i->j}_{\mathbf{p}^i_k}$ is the epipolar line in view j, obtained by back-projecting k^{th} point in i^{th} view ($\mathbf{p}^i_k$). From the epipolar constraint, $\mathbf{l}^{i->j}\mathbf{p}^j_k = 0$ holds if $\mathbf{p}^j_k$ lies on $\mathbf{l}^{i->j}$. Next, we derive an algorithm for a correspondence search in multiview images.

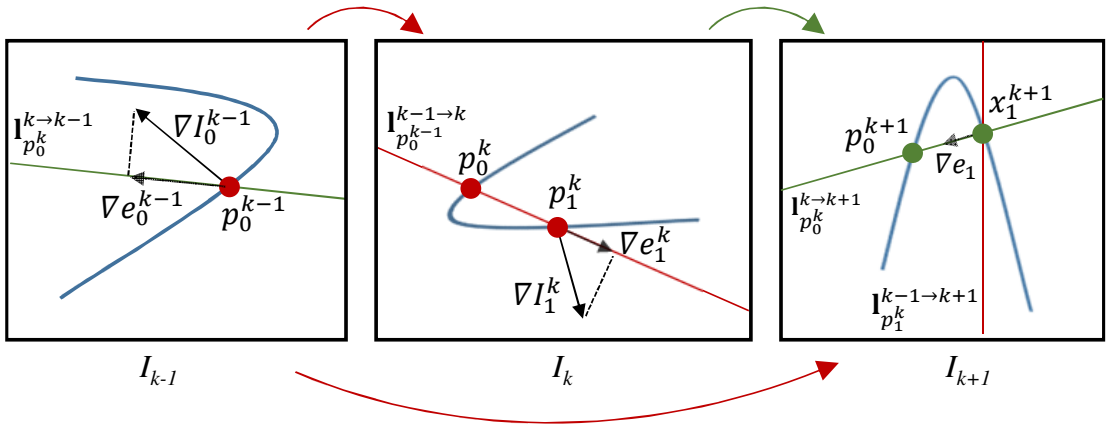
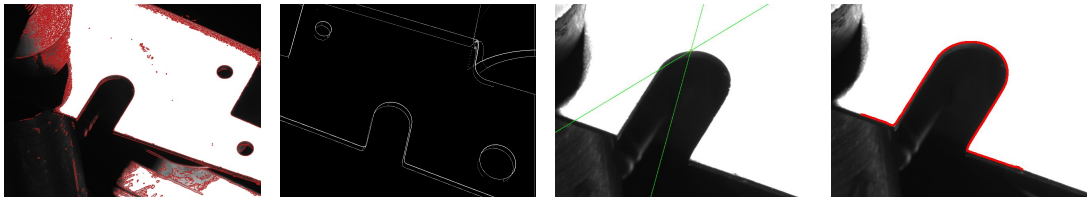


Figure 4.4: Edge pruning. The figure demonstrates a 3-view scenario: The edges in the first image I_{k-1} (e.g. red point) are transferred to view I_k. The edge points of I_k lying on the epipolar line (red) that have consistent epipolar gradient features $\nabla\mathbf{e}$ are selected and transferred to the I_{k+1}. The intersections of this new epipolar line with the edges of the I_{k+1} are selected in a similar fashion. In parallel, the epipolar line from I_{k-1} to I_{k+1} ($\mathbf{l}_1^{\mathbf{k-1->k+1}}$) is also generated. At this point, the edges lying closest to the intersection are found to correspond in all 3 views (intersection of red and green lines). All the other correspondence hypotheses are discarded.

The idea is to iteratively transfer the edges from the first view to the last one, using the essential matrix. Each transfer step involves the intersection of the target frame's edges with the epipolar line, generated from the source edge point in the previous image. The third view possesses a unique intersection. This is often referred as the trinocular constraint [20]. For the rest of the views, the distance between the edge pixels and the optimum intersection is minimized. More formally, we start by obtaining the view with the least number of edge pixels, $\mathbf{C}_0$. Then, let $\mathbf{p}^0 = \{\mathbf{p}_i^0\}$ be all the subpixel edge pixels in this view. Then, given a pixel in $\mathbf{C}_0$, $\mathbf{p}_i^0$, we find a single correspondence hypothesis in view n in the following fashion:

$$x_i^n = \begin{cases} \underset{\mathbf{p}_j^n}{\arg\min} \|(\mathbf{l}_{\mathbf{p}_i^0}^{0\to n} \cap \mathbf{l}_{\mathbf{p}_k^1}^{1\to n}) - \mathbf{p}_j^n\|, & n = 2 \\ \underset{\mathbf{p}_j^n}{\arg\min} \sum_{k=2}^{n-1} d(\mathbf{p}_j^n, \mathbf{l}_{\mathbf{x}_i^k}^{k\to n}), & n > 2 \end{cases} \tag{4.1}$$

$\mathbf{x}_i^n$ denotes the optimum intersection for $\mathbf{p}_i^0$ in view n. $d(\cdot)$ is the point to line distance in 2D space. This is indeed a recursive formulation, where $\mathbf{x}_i^0 = \mathbf{p}_i^0$ and $\forall\mathbf{x}_i^0, \exists\mathbf{x}_i^1$:



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

Figure 4.5: Edge synthesis from CAD models. **a)** 1^{st} image of multi-view system with subpixel edges overlayed. **b)** View synthesized by edge selection on CAD model and shows the visible edges of the model. **c)** Epipolar lines in 1^{st} image given the correspondents on other two. **d)** Selected edges by multi-view epipolar intersection.

$\mathbf{x}_i^1 = \mathbf{p}_j^1 \in (l_{\mathbf{p}_i^0}^{0->1} \cap \mathbf{p}^1)$. Multiple hypotheses are pruned by retaining only the closest intersection points. Similarly, only the intersections are transferred to the next view. Still, computing all intersections of epipolar lines increases the search space and can result in undesired matches. Thus, we only match and transfer the contours respecting the consistency of intensity gradient in the direction of the epipolar line, *epipolar gradient feature* [285]. Eventually, as summarized in Fig. 4.4, for each point $\mathbf{p}_0^i$ a tracked trajectory is obtained if the point is visible in at least 3 cameras.

Edge selection on CAD models Typically, CAD geometry involves mesh edges, which do not correspond to physical structure. Therefore, we also process the CAD models to obtain image consistent edges. As in [281], synthetic views of CAD model are generated by projecting & rendering the model onto a 3-channel image, where each channel encodes one direction of the normal information, establishing the angle-magnitude relations. This image is then processed to select the image-visible-CAD-edges, using standard edge extraction techniques, e.g. Canny. Finally, a noise cleaning algorithm is conducted in the form of a connected edge segment filtering. The goal is to retain only the longest edge segment. Steger proposed an edge linking algorithm suited for sub-pixel contours, which we directly use in our scenario [252]. Succeeding the sub-pixel edge linking, we filter the short edge segments and merge the collinear ones to form the final curvilinear structures. The steps as described in this entire section are depicted in Fig. 4.5. The output of this section is a set of 2D contours in distinct views, that are to be measured via CAD fitting. An image consistent 3D point cloud is also computed to be used in model fitting.

4.2.3 CAD Model Fitting and Triangulation

Even though the subpixel edge segmentation is reliable, the parts are almost always subject to severe noise due to cracks, crusts, surface defects, mis-manufacturing and calibration errors. Moreover, even if the multiview edges constitute a reasonable cue, there is no clear method on how to reconstruct those arbitrary contours.

Hence, we seek to find a robust way to relate the 2D geometric information with the CAD model, under the constraint that the registration respects the inspected 3D shape. To do that, we propose a projective variant of robust ICP. For the moment, we start by assuming that the inspected part goes through a rigid transformation, i.e. we handle non-rigidity in error compensation, and the CAD model is available apriori. Let $\mathbf{X}_k = \{\mathbf{x}_{k1}^T, \mathbf{x}_{k2}^T, ..., \mathbf{x}_{kM_k}^T\}$ denote the points sampled around the measurement point k, on the partial 3D CAD model, $\mathbf{x}_i \in \mathbb{R}^3$. $\mathbf{X}_k$ exists only for the cameras containing this partial model in their FOV. Thus, $\mathbf{X}^i = \{\mathbf{X}_k^i\}$ is a vector of edges of the measurement points being visible per each camera view i when projected. These points are obtained via sampling the CAD model to desired resolution. The generated model point cloud is further subject to visible edge selection. Thus, number of CAD points differs per each view e.g. $\|\mathbf{X}_k^1\|_L \neq \|\mathbf{X}_k^2\|_L$. Similarly, let $\mathbf{p}^k = \{\mathbf{p}_1^k, \mathbf{p}_2^k, ..., \mathbf{p}_{N_k}^k\}$ be the selected image edges in view k. We have $\mathbf{P} = \{\mathbf{p}^1, \mathbf{p}^2, ..., \mathbf{p}^K\}$, where K refers to the number of cameras capturing X_k. Due to the positioning on the conveyor belt, we assume that initial pose is within reasonable error bounds, i.e. the inspected part should remain fully within the view of the camera. We then formulate the CAD matching as the minimization of multiview re-projection error, as follows:

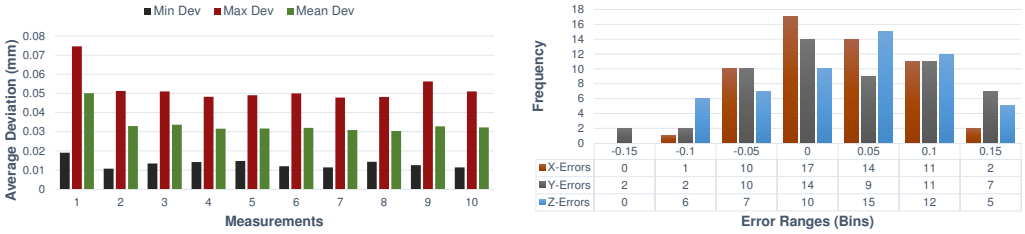
$$E(\boldsymbol{\theta}) = \sum_{k=1}^{K} \sum_{i=1}^{M_k} w_{ki} \|\Omega(\mathbf{x}_{ki}, \mathbf{p}^k) - \mathbf{x}_{ki}\| \qquad \mathbf{x}_{ki} = \Pi(\boldsymbol{\theta_i}, \mathbf{X}_i^k) \tag{4.2}$$

where $\boldsymbol{\theta}^{\boldsymbol{T}} = \{f_x, f_y, p_x, p_y, \mathbf{q}^T, \mathbf{t}^T\}$ are the standard pinhole projection parameters (intrinsic and extrinsic). We parametrize the rotations by the quaternion $\mathbf{q}$. Π denotes a perspective projection function according to the pinhole camera model. For this step we avoid self-calibration of intrinsic orientation and only update the exterior pose parameters. Since fixing the intrinsics frees us from FOV adjustment, we could simply back-project the image points beforehand and work purely on normalized coordinates. This significantly decreases the computational load and speeds up the

convergence as we only optimize over 7 parameters per camera. Ω is a closest point function, used to find the point $\mathbf{p}_j^k$ in view k, which has the shortest Euclidean distance to the projection of the 3D point $\mathbf{X}_i^k$. w_k^j are the robust weights associated with the residuals and $\|\cdot\|$ is the Euclidean norm. Note that the visible CAD edges, and thus the synthesised model are unique per each rendered view. Naturally the multi-view correspondences are not available preventing us from simply referring to the bundle adjustment literature. For these reasons, we exploit an iteratively re-weighted least squares optimization where at each iteration we obtain the correspondences in the projected domain, while updating the parameters (and solve for the transformation) in full $3D$. The weighting is introduced to wane the effect of the outliers. This is already well studied in robust computer vision literature [189]. For our application, we select the weights w_k^j to be Tukey's bi-weights. We benefit from Levenberg Mardquardt solvers to directly minimize $E(\boldsymbol{\theta})$. A similar idea is already presented in the context of ICP by [97] and named *LM-ICP*. Thus, we refer to our solver as the *Robust Projective LM-ICP*. Our formulation can also be thought as an alternating minimization between the closest point assignment step and bundle adjustment.

Malassiotis *et al.*[184] uses a similar method for the case of stereo. However, they take into account the parametric nature of the CAD model and re-generate the model points in each iteration. This requires a parametric CAD geometry, decreases the well-posedness of the problem and increases the number of parameters to optimize. Also, it is not apparent in their method, whether the scale or anisotropic changes in the model points are due to the registration error or really stem from the model manufacturing. We find such setting restrictive. In contrast, by using less parameters, our method is efficient, could well operate on non-analytical CAD models and does not suffer from scale ambiguities.

To improve the convergence speed, we employ a coarse-to-fine scheme, in which the pose is refined in a hierarchy of down-samplings. We also use the idea of Picky-ICP [319], which establishes unique correspondences at each iteration. Finally, 3D measurement is concluded by a simple look up of the center position, of the registered and transformed CAD part. The computed 3D positions are corrected via error compensation procedure as described in § 4.2.1.



(a) (b)

Figure 4.6: **(a)** 180 feature points on the calibration plates are measured with ATOS and with our system, 10 times. Using all the measurements, a 3D transformation is computed between the coordinate systems and the coordinate-wise difference in the common frame is recorded. It is shown that our system shows resemblance with ATOS system in the measurement of reference points ($\epsilon_{max} < 0.1mm$). Note that, ATOS is good at measuring such points, as the fiducials lie on planar surfaces. Yet, it will later become ineffective in measuring complicated structures due to fringe projection drawbacks. **(b)** Accuracy Evaluations: Histogram of registration errors to the CMM coordinate space. Bins indicate the error amount, while frequencies correspond to number of registrations. The data is taken over 55 measurements. Note that, the errors are recorded after coordinate space registration between our system and common CMM frame. In this respect, the figure plots the accuracies relative to a CMM, which is the main measure of error in our system. Values are in milimeters.

4.3 Experiments and Results

4.3.1 Calibration Accuracy

Intrinsic calibration Intrinsic orientation is one of the least significant sources of error in our system and is kept constant throughout all the experiments, where the re-projection errors were below 0.125 pixels. Considering the image resolution, such accuracy is enough for capturing as fine details as 10μ.

Extrinsic calibration Our system relies not on the accurate production of the calibrator but on the measurement of reference points, placed on it. As described in § 4.2.1, 3 Canon 6D cameras are assembled and the calibration plates, as well as the overlapping inter-reference points (thanks to random dots) are measured with multiple shots and bundle adjustment. Due to the presence of optical inspection points, the manufactured calibrator cannot be measured with a CMM and we do not have

any ground truth to compare against. Thus, we compare our measurements with ATOS, an industrial structured light triangulation system. As ATOS relies on fringe projection, it cannot recover the tiny details over complex structures (such as subpixel edges) accurately. But, it is very accurate in measuring reference points found on planar/smooth surfaces, such as the calibrator.

Fig. 4.6(a) presents a comparison between our system and ATOS in measuring the reference field composed of 180 reference points, on the calibrator. To be compatible with ATOS's precision requirements and to minimize the distortion effect, we use circular markers with a radius of 3 mm. The measurements are repeated 10 times, resulting in a total of 1800 measurements. We register ATOS and our system to the same coordinate frame. We accomplish this by a well known transformation estimation proposed by [133] using all the measured points. To this end, we assume that because both systems are sufficiently accurate, the errors in registration would have minimal disturbance to the coordinate frame alignment. It is noticed from the results that the registration error is small enough and the root mean squared error (RMSE) is 0.03mm. Every residual per coordinate contributes equally to RMSE. We accept this error and conclude that the calibrator is measured accurately and precisely to achieve the desired accuracy.

4.3.2 Measurement Analysis

CAD registration The iterations of refinement for a local camera group are visualized in Figures 4.7, 4.9(a). Regarding the fitting of CAD model to arbitrary subpixel contours, the final RMSE registration error is typically in the range of 0.25 pixels. To quantify the numerical accuracy, we conducted a set of experiments on different types of model parts. Fig. 4.8(a) plots the convergence of our algorithm, while Fig. 4.8(b) shows the inliers retained after each iteration.

Accuracy (errors in comparison to CMM) Error compensation step involves repeated measurement and registration to CMM space in order to reduce the effect of systematic errors. CMMs report an error of 0.01mm for the points of interest and stand out to be the ground truth reference for our system [283]. The fact that we accept CMM results as a baseline for our measurements, make us interested in the deviations

from this pseudo-ground truth. For this reason, we sample 55 random measurements of 5 different parts, which are measured both with our system and with Hexagon CMM. We then register each part from our coordinate space to CMM and record the errors. As this experiment is performed on the measured part but not on the calibrator, we end up with an evaluation of the accuracy of our system, with respect to the CMM. Fig. 4.6(b) shows a histogram over the collected data. It is shown that the maximum registration error is well below 0.2mm. This is both within the industry standards and within the tolerances we require.

Precision We will again refer to CMM comparisons. Fig. 4.9(b) enlists a series of experiments conducted on 5 different measurement points. In total 25 random measurements are displayed and the inter-measurement errors, which we will be referring as the repeatability measures are tabulated. The maximum error remains significantly under 0.15mm, while appearing below 0.05mm most of the times.

The reported accuracy in the previous section is less than the precision values due to the undesired compensation errors. In other words, the bias learned in § 4.2.1 does not always represent the systematic components and is, to a certain extent, subject to random/unstructured noise. For this reason, while we are able to get measurements with maximum ±0.15mm, we introduce certain errors in the registration to final coordinate space. Nevertheless, this error is not more than 0.05mm and is tolerable in our scenario.

4.3.2.1 Runtime Performance

Our system aims online operations, where the complete unit is installed on an operating production line. Therefore speed plays a key role in our design. Tab. 4.1 tabulates the timings for different stages of the runtime on a machine with 3GhZ Intel i7 CPU and 8GB of RAM. Note that image acquisition (30fps) step is responsible for most of the delays due to synchronous capture and bandwidth. Respectively, each camera captures the individual frames with a 0.5sec of strobe timing (until LEDs reach the desired intensity level). No processing starts until all the cameras finish the acquisition. Triangulation is only applied after every feature point is successfully extracted.

Our system is incomparably fast w.r.t. ATOS or CMM. It is, in that sense, not a replacement for a CMM device but a much more effective online competitor of the

Table 4.1: Average timings in measurement stage. Timings are reported as an average of 10 runs. MP refers to *measurement point*. Even though processing of the measurement points differ, their contribution to final runtime is averaged.

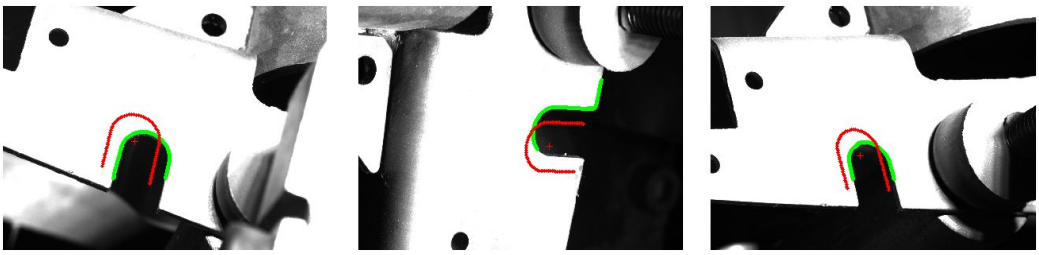
	Per MP (sec)	For All MPs (sec)
Image Acquisition	5,00	75,00
Feature Extraction	0,30	4,50
CAD Registration	0,15	2,25
Bundle Adjustment	0,10	1,50
	Total	**83,25**

existing mechanical fixtures. Inspecting a single part took 1h when measured with ATOS and 45min on CMM. Both of these results are far from online 100% inspection.

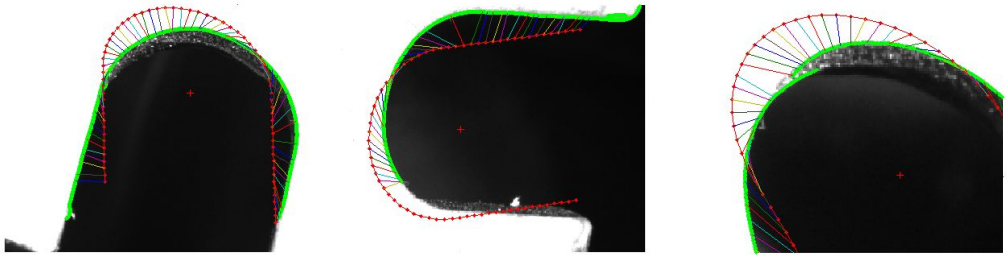
4.4 Discussions

We presented an in depth analysis of a novel, accurate optical metrology system, designed to operate on industrial settings, where realtime performance and durability are of concern. We showed how to extrinsically calibrate such a system that is composed of multiple non-overlapping camera groups via marker based photogrammetry. We compared our system against the industry standard measurement devices, one mechanical and one optical. Our measurement results are well within the accepted tolerances.

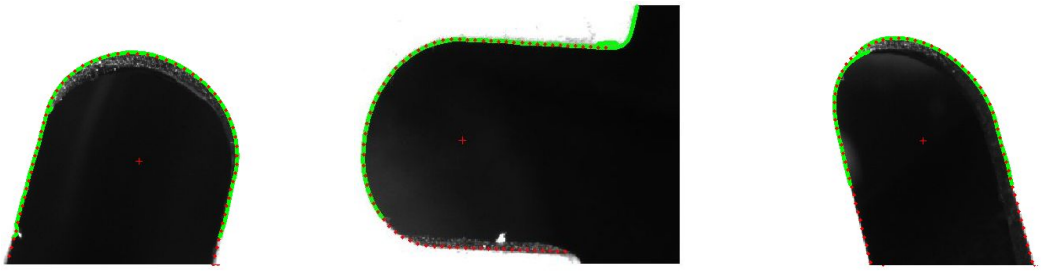
While one possible feature work involves the evaluation of self calibration methods and other error handling mechanisms to recover from severe misuses, we stop at that stage and come back to the point where we started: 3D reconstruction. Proposed system, together with the markers, can be either used to generate a sparse 3D reconstruction, or as a means to acquire 3D keypoints and calibration information in the form of a ground truth. The latter will provide us the data to evaluate our dense reconstruction methods, in the following chapters, that are specialized on dense 3D point cloud stitching.



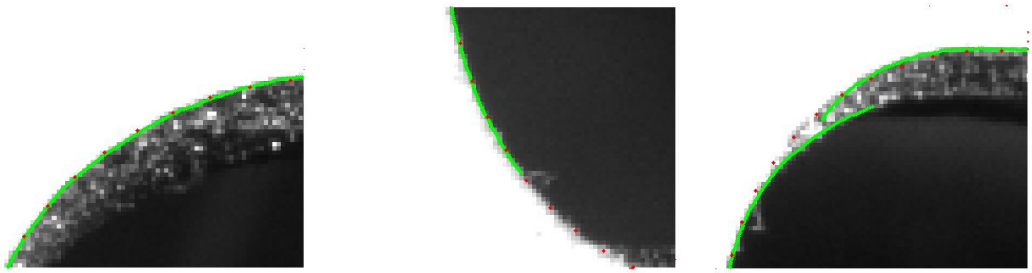
For each view, the initial poses with non-ideally extracted image edges.



Unique correspondences for each view after a single iteration of projective-ICP.

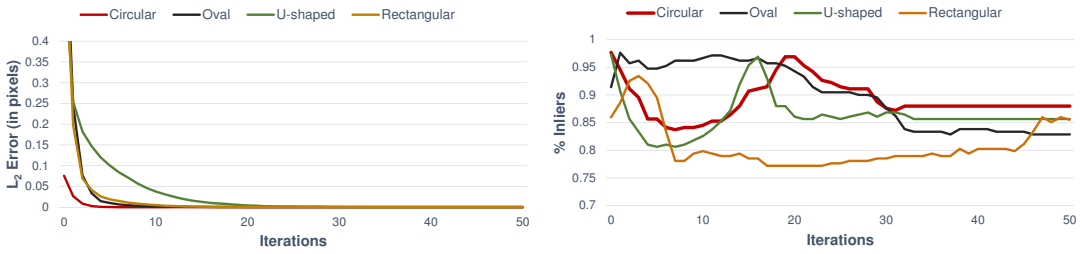


Final poses in each view.



Close up view of the robust sub-pixel registration.

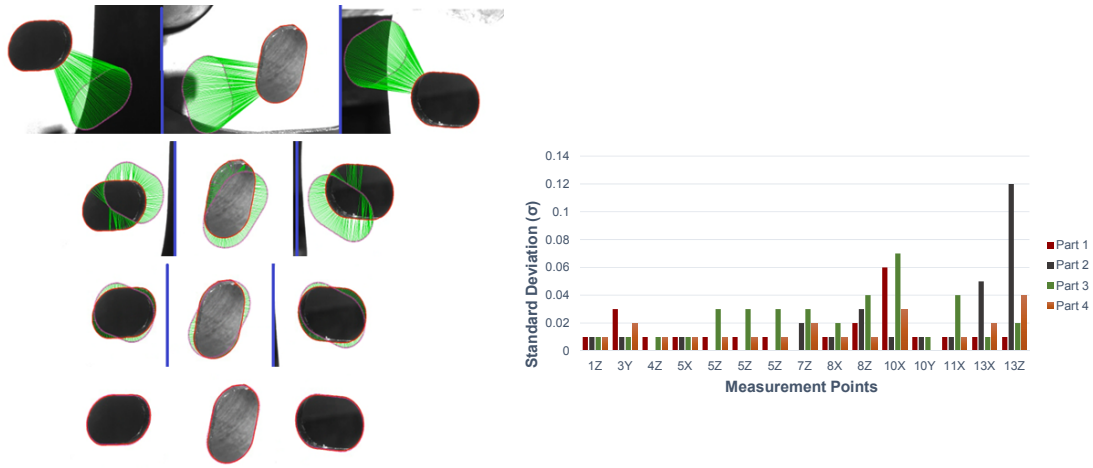
Figure 4.7: The entire process of projective multiview fitting. Red points indicate the model projections, while the green points are the extracted edge pixels. Edges are consciously sub-sampled to 60% of the true amount to simulate undesired occlusion. Sampled points are not the view-wise correspondents. Even though such a severe scenario does not occur in real life, it is again done intentionally to demonstrate the performance in a rather extreme case. It is worth noting that despite the sparse sampling of the CAD model and missing edges, the algorithm succeeds to converge to an acceptable minimum. Image on Row 4, Column 3 visualizes the occlusion robustness. The effects of manufacturing errors in creating surface defects are clearly apparent on the edge image in Row 4, Column 4.



(a) Registration errors for different types of measurement points

(b) Inliers per iteration

Figure 4.8: Evaluation of registration performance: **a)** Each run is started with 2.5cm translational shift and 10° rotational offset. The registration error between the inliers is recorded at each iteration. Note that the convergence is super-linear in all types of geometrical structures. **b)** Ratio of Inliers over iterations.



(a)

(b)

Figure 4.9: **(a)** The process of CAD fitting: Red points indicate the found edges, purple points are the model projections, while the green points are the extracted edge pixels. Each column is a view and each row shows an iteration. **(b)** Repeatability results for 4 different parts, manufactured from the identical CAD model. For each part, 15 out of 26 points are measured, 5 times each (randomly taken out of 41). The standard deviation per point is plotted.

CHAPTER 5

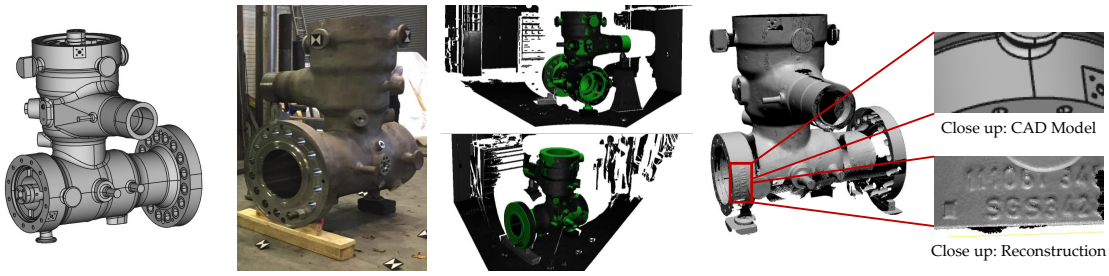
A NEW PIPELINE FOR DENSE 3D RECONSTRUCTION

"Programming is one of the most difficult branches of applied mathematics; the poorer mathematicians had better remain pure mathematicians."

— Edsger Dijkstra

Despite the huge demand, many marker-free approaches based solely on 3D data either involve acquisition of ordered scans[148, 208], or follow the de-facto standard pipeline [135] in case of unordered scans. The former suffers from the requirements of redundant depth capture with large overlap and scenes with very little clutter or occlusions. Due to the volumetric nature of scan fusion, such techniques also do not scale well to large objects while retaining high precision. The latter exploits 3D keypoint matching of all scans to one another, alleviating the order constraint. Thanks to 3D descriptors, it could well operate in full 3D. Yet, matching of scans to each other is an $O(N^2)$ problem and prevents the methods from scaling to an arbitrary number of scans. In addition, neither of those can handle scenes with extensive dynamic clutter or occlusions. Nowadays, with the capability of collecting high quality, large scale and big data, it is critical to offer efficient, automated solutions for providing highly accurate reconstructions regardless of the acquisition scenarios.

In this chapter, we tackle the problem of 3D dense instance reconstruction from



(a) CAD Model **(b)** Image of Object in Scene **(c)** Object Detection in Laser Scans **(d)** 3D Reconstruction

Figure 5.1: Our 3D reconstruction method. **(a)** Input 3D CAD model. **(b)** Image of the instance to reconstruct. **(c)** Detection of 3D model in point clouds. **(c)** Final reconstruction we obtain, with close-up comparisons to the nominal CAD prior.

a handful of unorganized point clouds, where the object of interest is large in size, texture-less, surrounded by significant dynamic background clutter and is viewed under occlusions. Our method can handle scenes between which no single transformation exists, i.e. the same objects appear in different locations, such as the one in Fig. 5.2. We also do not impose any constraints on the order of acquisition. To solve all of these problems simultaneously, we make use of the reasonable assumption that a rough, nominal 3D CAD model prior of the object-to-reconstruct is available beforehand and propose a novel reconstruction pipeline. Such assumption of a nominal prior is valid for many applications especially in industry, where the objective is to compare the reconstruction to the designed CAD model. Even for the cases where this model does not exist, one could always generate a rough, inaccurate mesh model with existing methods, e.g. KinectFusion [208], to act as a prior. Note that, due to manufacturing errors, sensor noise, damages and environmental factors, physical instances deviate significantly from the CAD models and the end-goal is an automatic algorithm to accurately recover the particular instance of the model. With the introduction of the prior model, we re-factor the standard 3D reconstruction procedures via multi-fold contributions. We replace the scan-to-scan matching with model-to-scan matching resulting in absolute poses for each camera. Unlike the case in object instance detection where false positives (FP) are tolerable, object instance reconstruction is easily jeopardized by the inclusion of a single FP. Therefore, one of our goals is to suppress FP, even at the expense of some true negatives. To achieve this, we benefit from robust and well engineered object detection & pose estimation algorithms, that are described in detail

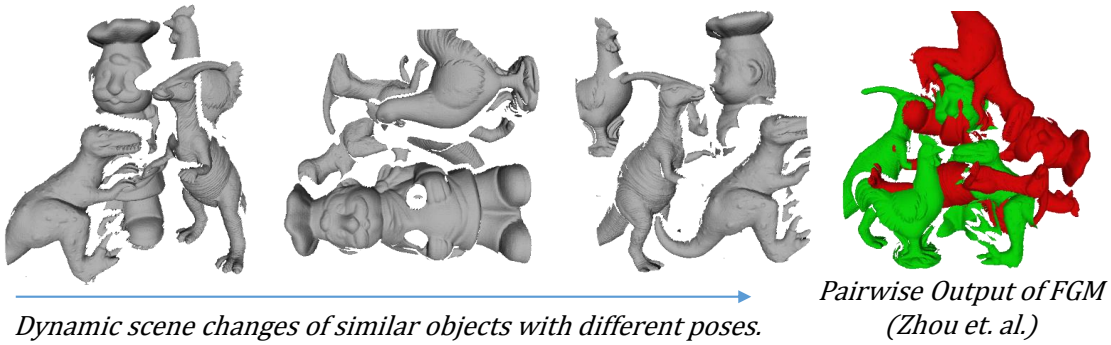


Figure 5.2: Dynamic clutter and occlusions on Mian dataset [192]: A change in relative locations of the objects between scenes easily fools the modern global registration algorithms [314]. We operate on the object level and circumvents this problem.

in the following chapters. The resulting match is followed by automatically segmenting out the points belonging to the object and transforming them back onto the model coordinate frame. Doing this for multiple views results in roughly aligned partial scans in the CAD space. To fully recover the exact object, the CAD prior is then discarded (as it might cause undesired bias) and a global multi-view refinement is conducted only to optimize the camera poses. It is a by-product of our method that the registration is directly in the space of the model and this makes further operations such as deviation analysis easier. In global scan registration, the creation of a *pose graph* indicating which scans can be registered to each other is required. The typical complexity of obtaining this graph is $O(N^2)$, where all scans are matched to one another. We profit from the CAD prior and contribute by automatically computing this graph, in which only cameras sharing significant view overlap are linked. This reduces the complexity to $O(N)$, and robustifies the whole pipeline. The entire procedure is made efficient so that large scans are handled in reasonable time. Exhaustive evaluations demonstrate high accuracy regardless the object size, clutter and occlusions.

This chapter and thereby the pipeline forms the basis of the upcoming chapters that are devoted to refining and advancing the individual components of the foundation explained here. Fig. 5.1 shows the types of industrial objects we can handle, while a supplementary video demonstrating the method devised in this chapter can be found under: `https://youtu.be/KPA_8BNuOvg`.

5.1 Prior Work

Arguably, the most wide-spread 3D reconstruction methods are KinectFusion[208] and its derivatives [56, 148, 249]. These methods have been successful in reconstructing small isolated objects, but their application is not immediate when the size increases, or clutter and occlusions are introduced [209]. Due to extensive usage of signed-distance fields, they are bound to depth images and a sequential acquisition.

Abundance of works exist in multi-view global alignment from multiple 3D unorganized point clouds [17, 40, 76, 89, 89, 91, 114, 154, 259, 266, 269, 314, 315]. These methods assume the scans to be roughly initialized and reasonably well-segmented. They, in general, handle slight synthetic noise well enough, but they do not deal with cluttered and occluded data. Another track of works try to overcome the those constraints by using keypoint detectors and matching descriptors in 3D scans. One of the pioneering works proposing a feature agnostic, automatic and constraint-free algorithm is the graph based in-hand scanning from D. Huber and M. Hebert [135]. The authors set a baseline for this family of methods. Novatnack and Nishino [212] developed a scale dependent descriptor for better initialization and fused it with [135] to assess the power of their descriptor. Yet both of these relied on range image data. Mian *et al.* [192] proposed a tensor feature and a hashing framework operating on meshes. Fantoni *et al.* [92] uses 3D keypoint matching as an initial stage of multiview alignment to bring the scans to a rough alignment. Zhu *et al.*[315, 317] as well as Liu and Yonghuai [172] use genetic algorithms to discover the matching scans and use this in global alignment context. These stochastic schemes are correspondence free but slow. Similar to [192], Zhu *et al.*[316] devises a local-to-global minimum spanning tree method to align the scans. A majority of these automatic alignment procedures suffer from increased worst case complexity of $O(N^2)$, where N is the number of views. Moreover, since there is no integrated segmentation, the registration procedure cannot handle clutter and occlusions.

Use of CAD models in reconstruction is not novel by itself. Savarese *et al.*[305] enrich the multiview reconstruction from 2D images with a CAD prior. Guney and Geiger [116] use object knowledge to resolve the stereo ambiguities. Birdal *et al.*use models in triangulation by registration [31]. Other works [168, 301, 318] use CAD prior to detect categories, while we focus on proxy instance priors for initial alignment and

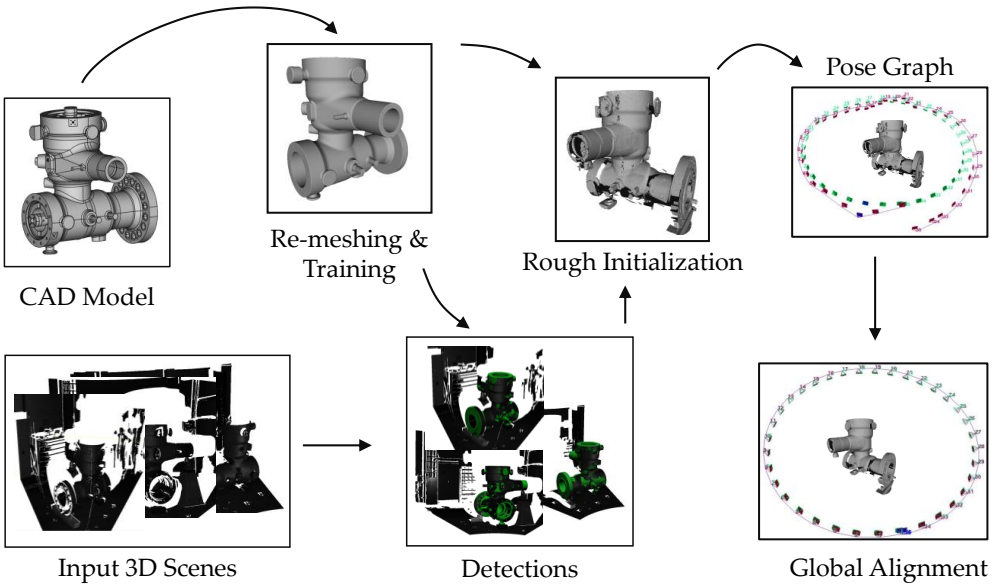


Figure 5.3: Proposed 3D reconstruction pipeline: Prior CAD model is trained to create the model representation. Input scenes are then parsed for the model pose. Pose estimates initialize a rough reconstruction, with segmentation and automatically computed pose graph. This is then further refined to the full reconstruction.

then operate directly on 3D points. Salas *et al.*[234] propose SLAM++ using object priors to constrain a SLAM system. Unlike that, we perform instance reconstruction using the CAD prior, and not SLAM. In our setting, both the background and the object are allowed co-change dynamically from scene to scene.

5.2 Method

We now formally elaborate on the details of the pipeline, illustrated in Fig. 5.3. Given a set of unstructured and unordered 3D scenes $\{\mathbf{P}_i\} \in \mathbf{P}$, we seek to find a set of transformations $\{\mathbf{T}_i\} \in SE(3)$ so as to stitch and reconstruct a global model $\mathbf{P}_G$:

$$\mathbf{P}_G = \bigcup_{i=1}^{N} \left(\mathbf{T}_i^0 \circ \mathbf{P}_i\right) \tag{5.1}$$

$\mathbf{T} \circ \mathbf{P}$ applies transformation $\mathbf{T}$ to the scene $\mathbf{P}$. In this setting, both the transformations $\{\mathbf{T}_i^0\} \in SE(3)$ and the global model $\mathbf{P}_G$, as well as the initialization are unknown. Due to the lack of a common reference frame and apriori information about $\{\mathbf{P}_i\}$, obtaining

the set $\{\mathbf{T}_i^0\}$ typically requires $O(N^2)$ worst case complexity, where all the scene clouds are matched to one another to obtain the relative transformations aligning them. To better condition the problem and reduce its complexity, we introduce the supervision of a CAD proxy $\mathbf{M}$ in form of a mesh model and re-write Eq. 5.1:

$$\mathbf{P}_G = \bigcup_{i=1}^{N} \left(\mathbf{T}_i^M \circ (\mathbf{P}_i | \mathbf{M}) \right) \tag{5.2}$$

where $\mathbf{T}_i^M \in SE(3)$ is the transformation from the scene to the model space, such that the segmented scene points $(\mathbf{P}_i|\mathbf{M})$ come to the best agreement. To estimate $\{\mathbf{T}_i^M\}$, we follow a two stage technique. First, a rather approximate estimate $\{\tilde{\mathbf{T}}_i^M\}$ is made by matching the CAD model to a single scene. Note that this time, the set $\{\tilde{\mathbf{T}}_i^M\}$ can be computed in $O(N)$, since it only requires CAD to scan alignment. However, because scene clouds suffer from partial visibility, noise and deviations w.r.t. the CAD model, the discovery of the pose of the model in the scans provides only rough initial transformations to the model coordinate frame. For this reason, as the final stage, the CAD prior is disregarded and the scans are globally registered, simultaneously, refining $\{\tilde{\mathbf{T}}_i^M\}$ to $\{\mathbf{T}_i^M\}$. This lets us reconstruct configurations deviating significantly from the CAD prior.

Multi-view refinement Our procedure of multi-view refinement is similar to [92], where a global scheme for scan alignment is employed. Let $\mathbf{P}_1, \dots \mathbf{P}_M$ be the set of scans that are to be brought in alignment. To generalize and formalize the notation for registrations of all point clouds to each other, we maintain a directed pose graph in form of an adjacency matrix $\mathbf{A} \in \{0,1\}^{M \times M}$, such that $\mathbf{A}(h,k) = 1$ iff cloud $\mathbf{P}_h$ can be registered to cloud $\mathbf{P}_k$. Let $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_M)$ be the absolute camera poses of each view. The alignment error between two clouds $\mathbf{P}_h$ and $\mathbf{P}_k$ then reads:

$$E(\boldsymbol{\theta}_h, \boldsymbol{\theta}_k) = \mathbf{A}(h,k) \sum_{i=1}^{N_h} \rho\left(\| d(\boldsymbol{\theta}_h \circ p_i^h, \boldsymbol{\theta}_k \circ q_i^h) \|^2 \right) \tag{5.3}$$

where $\{\mathbf{p}_i^h \rightarrow \mathbf{q}_i^h\}$ are the N_h closest point correspondences obtained from the clouds $\mathbf{P}_h$ and $\mathbf{P}_k$. $d(\cdot,\cdot)$ is the point-to-plane distance. The overall alignment error, which we want to minimize at this stage, is obtained by summing up the contributions of every pair of overlapping views:

$$E(\boldsymbol{\theta}) = \sum_{h=1}^{M} \sum_{k=1}^{M} \mathbf{A}(h,k) \sum_{i=1}^{N_h} \rho\left(\| d(\boldsymbol{\theta}_h \circ p_i^h, \boldsymbol{\theta}_k \circ q_i^h) \|^2 \right) \tag{5.4}$$

where ρ denotes Tukey's Biweight M-estimator as defined in §2.9. The final absolute poses are the result of the minimization $(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M) = \arg\min_{\boldsymbol{\theta}}(E)$.

Implementation details In contrast to the pairwise registration error in Eq. (5.3), which has closed form solution for the relative transformation $\boldsymbol{\theta}$, there are no closed form solutions in the multiview setting. Therefore, we use a non-linear optimization procedure, Levenberg Marquardt. The rotations are parameterized with angle-axis representation. We constrain the frame with the highest number of points found on the CAD model to be static and update (solve for) the rest of the poses. In practice, this leads to faster convergence. Note that, in contrast to the methods that exploit pairwise registration, our poses are absolute and do not suffer from drifts or tracking artifacts. We also do not require a conversion from relative poses to absolute ones, which are usually obtained by the computation of a minimum spanning tree or shortest paths over the pose graph [113, 135]. This property eases the implementation and reduces errors, that are to be encountered in usual heuristics.

Due to the accuracy requirements, unlike [92], we omit using distance transforms at this stage. We rather use speeded up KD-Trees to achieve exact nearest neighbors [202]. Since we optimize over the poses, and not over 3D points, the trees are built only once in the beginning and all closest point computations are done in the local coordinate frame of the view of interest. This is important for efficiency. For reasons of accuracy, we use analytical Jacobians. As cloud sizes become large, this optimization exhibits significant computational costs. This is why, a priori sampling plays a huge role, where we use $\approx$ 20k to 30k points per scan, distributed evenly in space.

To summarize, our key contribution lies in obtaining $\{\mathbf{T}_i^M\}$ in a robust, efficient and accurate manner. In Chapter 7, we will explain the pose estimation methods used to compute the rough alignment $\{\tilde{\mathbf{T}}_i^M\}$. The next section is devoted to obtaining the pose graph (adjacency matrix) $\mathbf{A}$ in an automated fashion for such a CAD-based scenario.

5.2.1 Computing the Pose Graph and Live Feedback

Any global optimization algorithm requires an adjacency graph $G = (V, E)$, which encodes the existence of overlap between camera views. The nodes of this sparse graph contain the cameras $V = \{C_1..C_N\}$, whereas an edge E_{ij} is only created between nodes

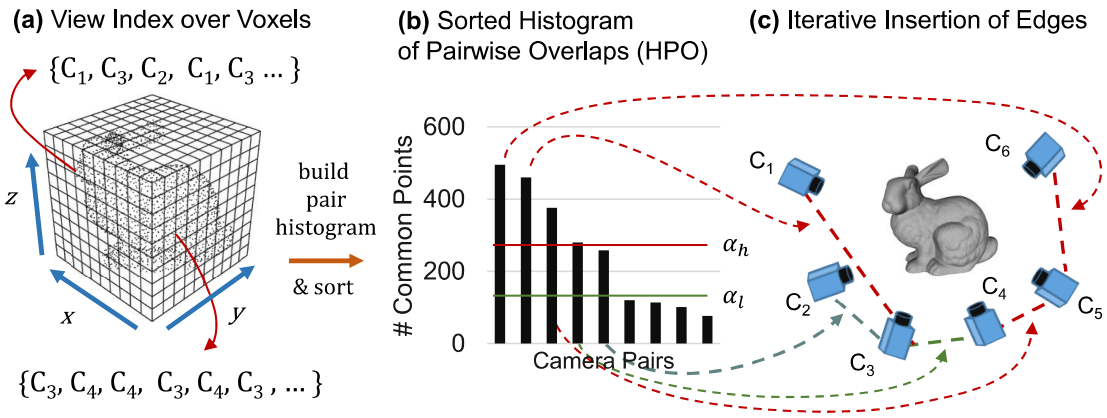


Figure 5.4: Pose graph computation. *See text for details.*

(C_i, C_j) if they share significant overlap. An absolute pose $\mathbf{T}_i$ is associated to each node and a relative pose $\mathbf{T}_{ij}$ is to each edge. Traditionally, this requires pair-wise overlap computation between all cameras. While a naive approach would involve linking the cameras, whose centers are found to be close, this is by no means a guarantee for shared overlap. Therefore, we present a more accurate approach, without sacrificing efficiency, thanks to the availability of the CAD model.

Consider the voxel grid index $\mathbf{D}$ of model $\mathbf{M}$ as in Fig. 5.4(a). Each segmented scene point $\mathbf{p}'_i \in \mathbf{T}_i^{-1} \circ \mathbf{p}_i$ is mapped to a voxel D_k, which stores a set of cameras $\{C_i\}$ observing it. Whenever the point $\mathbf{m}_k$ belonging to the voxel D_k is visible in the camera C_j, this camera is added to the list of cameras seeing that model point. Each list stores unique camera indices. From that, we compute the histogram of pairwise overlaps (HPO) as shown in Fig. 5.4(b).

While all the possible edges are now generated (as the bins in HPO), it is not recommended to use all these in the multiview alignment i.e. the overlap might be little, causing a negative impact. Instead, we adopt an iterative algorithm, similar to hysteresis thresholding. First, HPO is sorted with decreasing overlap (Fig. 5.4(c)). Next, two thresholds α_l and α_h are defined. All pairs with overlap less than α_l are discarded. All cameras with overlap larger than α_h are immediately linked and edges are constructed in the graph. If, at this stage, the graph is not connected, we start inserting edges from the remaining bins of HPO into A until either the connectivity or the threshold α_l is reached. This is illustrated in Figures 5.4(c) and 5.4(d). If the final graph is still not connected, we use the largest connected sub-graph, to ensure

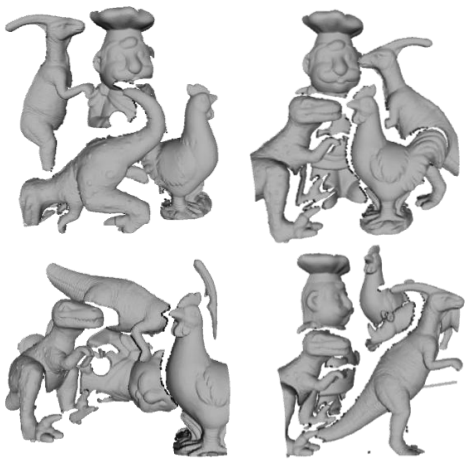
optimize-ability. For efficient online update, a modified union-find data structure is used to store the graph and dynamically insert edges when new views are encountered. Unlike quadratic complexity of the standard pose graph creation methods, ours has linear complexity.

Live feedback Due to the connected-ness of pose graph, our method is able to keep track of the overlap between all the point clouds, at all times, informing the user whenever graph disconnects or overlap is small. The complement of the already reconstructed part reveals the unscanned region, which is also fed back to the operator. Incoming scans directly propagate and form links in the pose graph, allowing online response to the user's actions.

Mesh generation While the literature enjoys a series of works for meshing (e.g. marching cubes or Poisson reconstruction), many of them target binary volumes, rather than arbitrary precision point clouds. They also cannot tolerate the large noise that is inherent in data. As our purpose is to retain the sensor accuracy as much as possible, we opt to mesh the point cloud using an octree implementation of the SSD-method of Calakali and Taubin [55].

5.3 Experimental Evaluation

We evaluate our dense reconstruction method against a set of real datasets acquired by laser scanners and structured light sensors. The CAD models we work with might contain uneven distribution of vertices or inner geometry. We always eliminate the inner structure by thresholding the ambient occlusion values [194] before the models are re-meshed [165]. The details of the sampling procedures are to be presented in the following chapter (Chapter 6). At detection time, a relative model and scene sampling distance of $d = \tau diam(\boldsymbol{M})$ is used, where $0.05 \geq \tau \geq 0.025$ depending on the object. We also adjust another threshold on the distance to consider a scene point to be on the model based on the sensor quality. For accurate scanners we use $1.5mm$, while for less accurate ones $0.5cm$. This does not affect the segmentation, but the hypothesis verification.



(a) Scenes of Mian

(b) T-Rex (Gnd. Truth vs Ours)

(c) Chef (Gnd. Truth vs Ours)

(d) Parasaurolophus (Gnd. Truth vs Ours)

Figure 5.5: Results on Mian dataset. **(a)** Subset of scenes from the dataset. **(b, c, d)** The ground truth models (left) and our reconstruction (right) for three objects.

5.3.1 Mian Dataset

We first compare the reconstruction quality on Mian dataset [192]. This dataset includes 50 laser scanned point clouds of 4 complete 3D objects, with varying occlusion and clutter. The objects change locations from scan to scan, creating dynamic scenes. The clutter and background also varies as the objects appear together with other different ones in each scene. We quantify this dynamic clutter by relating it to the

Table 5.1: Reconstruction results on Mian dataset (in mm). Each object is compared to the model provided by [192] using [106].

Model	w/o Opt.	with Opt.	# Scans	Clutter
Chef	2.90 ± 2.40	1.07 ± 0.65	22	0.58±0.11
Chicken	1.71 ± 1.60	0.33 ± 0.24	29	0.61±0.12
Para.	2.52 ± 2.00	0.41 ± 0.30	12	0.24±0.20
T-rex	2.36 ± 2.08	0.88 ± 0.62	27	0.14±0.22

provided occlusion values:

$$\text{Clutter} = 1 - \frac{\text{(Model Surface Area)*(1-Occlusion)}}{\text{(Scene Surface Area)}} \tag{5.5}$$

and provide it in Tab. 5.1 for each object. The models present in the scenes are provided by [192] to act as ground truth. We do not perform any prior operation to the scenes such as segmentation or post-processing except meshing via SSD [264]. For Parasaurolophus and Chicken objects, the pose graph becomes disconnected and therefore, we optimize individually the two sub-components and record the mean. We also report the number of scans in which the model is detected and verified. Not every model is visible in every scan. In the end, our mean accuracy is well below a millimeter, where the used sensor, Minolta Vivid 910 scanner, reports an ideal accuracy of ~ 0.5mm. We are also not aware of any other works, reporting reconstruction results on such datasets. Fig. 5.5 visualizes our outcome, and Tab. 5.1 shows our reconstruction accuracy both prior to and after the optimization. While our error is quantitatively small, the qualitative comparison also yields a pleasing result, sometimes being superior even to the original model.

5.3.2 Toy Objects Dataset

Since our objective is to assess the fidelity of the reconstruction, we opt to use the objects from the 3D printed dataset [249]: *Leopard*, *Teddy* and *Bunny* and *Tank* (See Fig. 5.8). The diameters of objects vary in the range of $15-30cm$. The print accuracy is up to 50μ, well sufficient for consideration as ground truth. To capture the real scenes, a home-brew, high accuracy phase-shift sensor, delivering <0.4mm point accuracy is chosen. We sample up to 10 scans per object, taken out of a 100 frame sequence. To

Table 5.2: Reconstruction errors on toy objects dataset (mm).

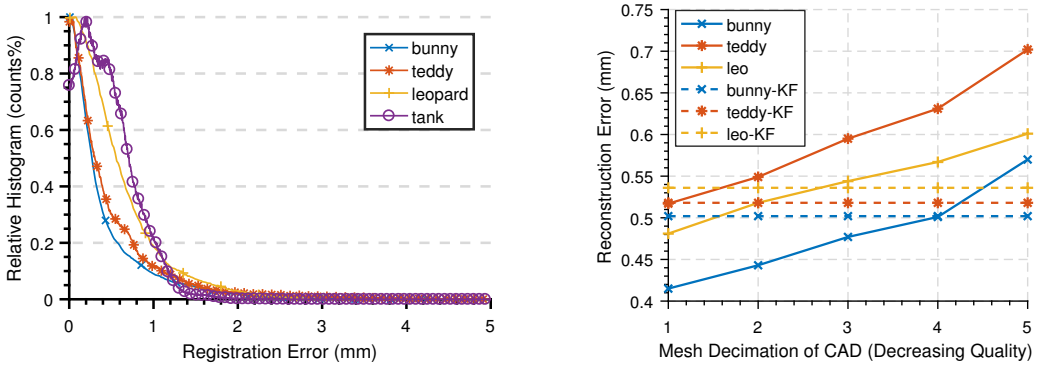
	Leo	**Teddy**	**Bunny**	**Tank**
KinFU	1.785 ± 1.299	0.998 ± 0.807	0.664 ± 0.654	1.390 ± 1.315
Kehl	1.018 ± 1.378	1.028 ± 0.892	0.838 ± 0.860	1.573 ± 2.250
Sdf2Sdf	0.652 ± 0.614	0.910 ± 0.584	0.541 ± 0.436	0.466 ± 0.416
Ours	$\mathbf{0.481 \pm 0.519}$	$\mathbf{0.517 \pm 0.572}$	$\mathbf{0.415 \pm 0.501}$	$\mathbf{0.451 \pm 0.322}$
Ours-KF.	0.536 ± 0.411	0.519 ± 0.582	0.502 ± 0.529	0.468 ± 0.474
Ours-CO.	0.651 ± 0.628	0.544 ± 0.601	0.698 ± 0.506	0.475 ± 0.433

disrupt the acquisition order, we randomly shuffle this subset and apply our algorithm. Next, we compute the CAD-to-reconstruction distances in CloudCompare [106]. We do not explicitly register our reconstruction to CAD model, because we already end up on model coordinate frame (Having the result in the CAD space is a by-product of our approach). Moreover, we use the original 100-frame, ordered sequence as an input to standard reconstruction pipelines such as KinectFusion [233], Kehl *et al.*[148] (also uses color) and Slavcheva *et al.*[249] all of which require a temporally ordered set of frames, with a large inter-frame overlap. All of these algorithms take depth image as input, whereas ours uses the unstructured 3D data and the model.

Our results on this dataset are shown in Tab. 5.2 (*Ours*) when original CAD is used. We also report the results when KinectFusion (KF) prior is used (*Ours-KF*). The individual error distribution of the objects are shown in Fig. 5.6(a). Because our method does not suffer from drift and computes absolute poses, although we use 10 times less scans, we are still 2-4 times more accurate then conventional methods. This also shows that our method could retain the good accuracy of the sensor.

Next, we augment this dataset with further scenes of the same objects, such that clutter and occlusions are present. Some shots are shown in Fig. 5.8 (mid-row). Our reconstruction accuracy (*Ours-CO*) is shown in Tab. 5.2. These results are still better than or close to Sdf2Sdf [249]. Due to inclusion of some outliers, our results get slightly worse than the one in no clutter, yet they are still acceptable. However, none of the other approaches can run on this new set due to the existence of significant outliers.

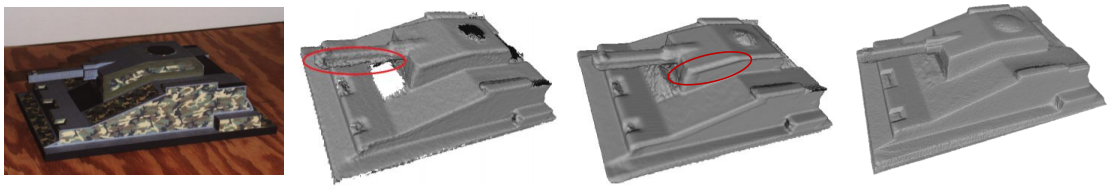
In a further experiment, we gradually decimate the toy models down to a mesh of $\approx$500 vertices. We exclude tank as the decimation has little effect on the planarities.



(a) Error Histo. in Toy Objects

(b) Sensitivity to CAD prior

Figure 5.6: Performance on the Toy Objects dataset.



(a) Tank Object **(b)** Kehl et. al. **(c)** Slavcheva et. al. **(d)** Ours

Figure 5.7: Visual comparisons on Tank object. Note the ability of our method in preserving sharp features.

As shown in Fig. 5.6(b), even when the CAD prior gets very crude, we are still able to achieve a reasonable reconstruction. Note that, results of KF prior is plotted in dashes as it is also a form of rough mesh approximation. Furthermore, Fig. 5.7(a) visually compares our reconstruction to the state of the art on the tank object. Because we do not use smoothing voxel representations (such as SDF), our method is much better at preserving sharp features at the model edges.

5.3.3 Augmented Toy Objects Dataset

We now provide additional results by augmenting the Toy Objects dataset with 4 more objects: *bird, colorful dog, dinosaur*, and *a caravan*. They are shown in Fig. 5.9 together with a 3D scan acquired by our in-house phase shift scanner. Note that the objects are not isolated for reconstruction but directly captured in dynamic clutter and occlusions.

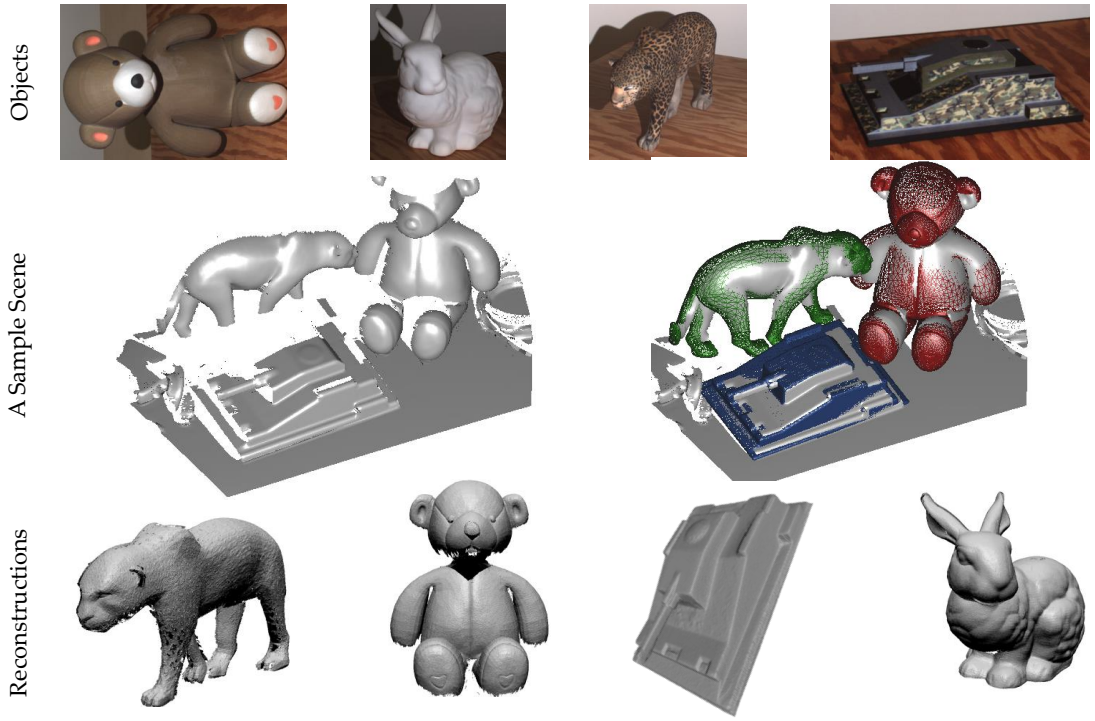


Figure 5.8: Qualitative results on toy objects. **first row** Real images of objects **second row** A sample scene and detections visualized **third row** Our results.

Figure 5.9: Collecting dataset of 4 additional toy objects. **(from left to right)**: RGB images of objects given to aid perception; the actual stereo images acquired by the scanner; triangulation of the stereo capture.

Using KinectFusion models as priors We now show how to use our method in improving already reconstructed models. This is an important application as scanning an object with traditional methods such as [148, 208, 249] is cumbersome: 1) they require an isolated and clean environment, 2) a sequential scanning is required with careful attention to registration errors. Our method is contradictorily easy to operate as we do not assume scanning order and can deal with clutter, occlusions and partial visibility.

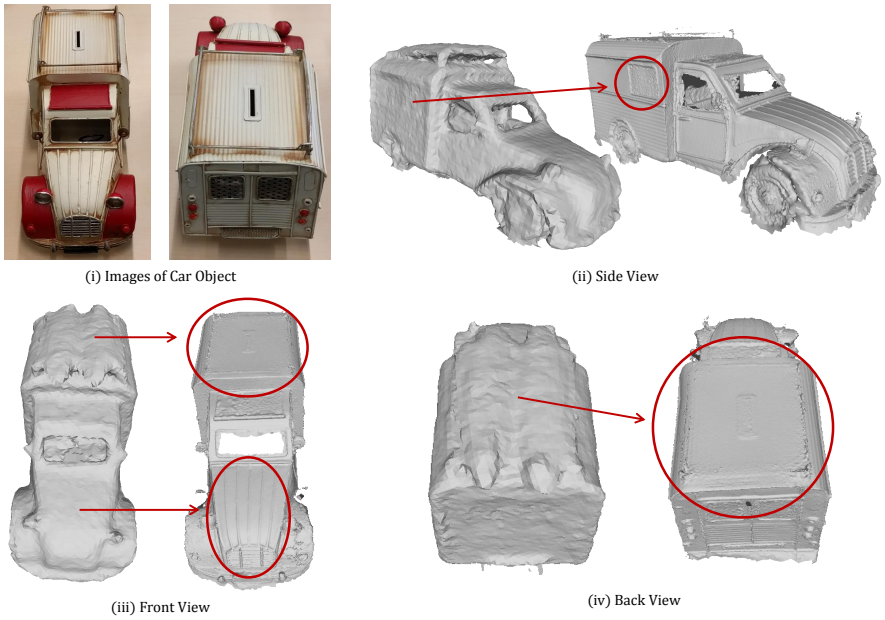


Figure 5.10: Improving KinFu reconstruction of Caravan object.

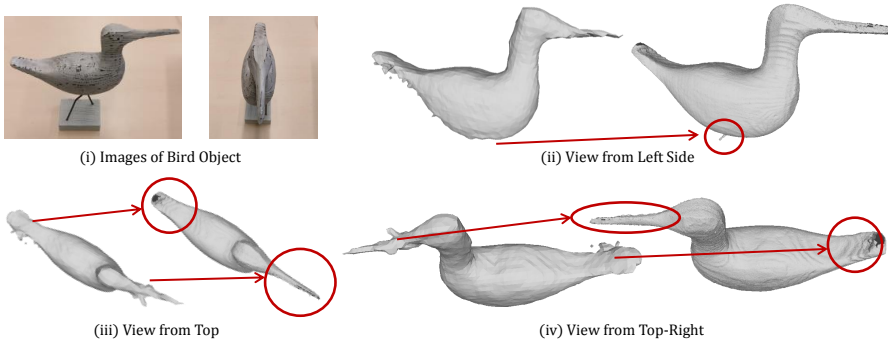


Figure 5.11: Improving KinFu reconstruction of Bird object.

Simply using a scanner providing better quality than Kinect, we could improve upon 3D models with only a handful of shots. We reconstruct the objects in our set with Kinect Fusion from Point Cloud Library and use these reconstructions as proxy for our algorithm. Figures 5.10,5.12,5.13,5.11 show our results versus the proxy Kinect Fusion model. Note that for Kinect Fusion we deliberately use a large voxel size (low resolution) to emphasize the margin of recovery. Thus these models are incomplete, and sometimes quite noisy. Our results open up application areas, where coarse models such as the Kinect Fusion ones, are improved with ease by using another acquisition

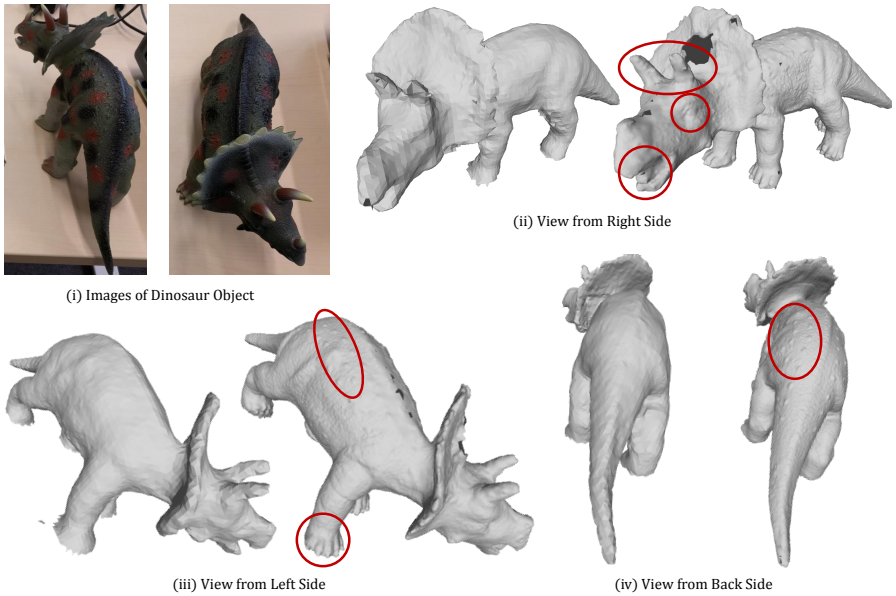


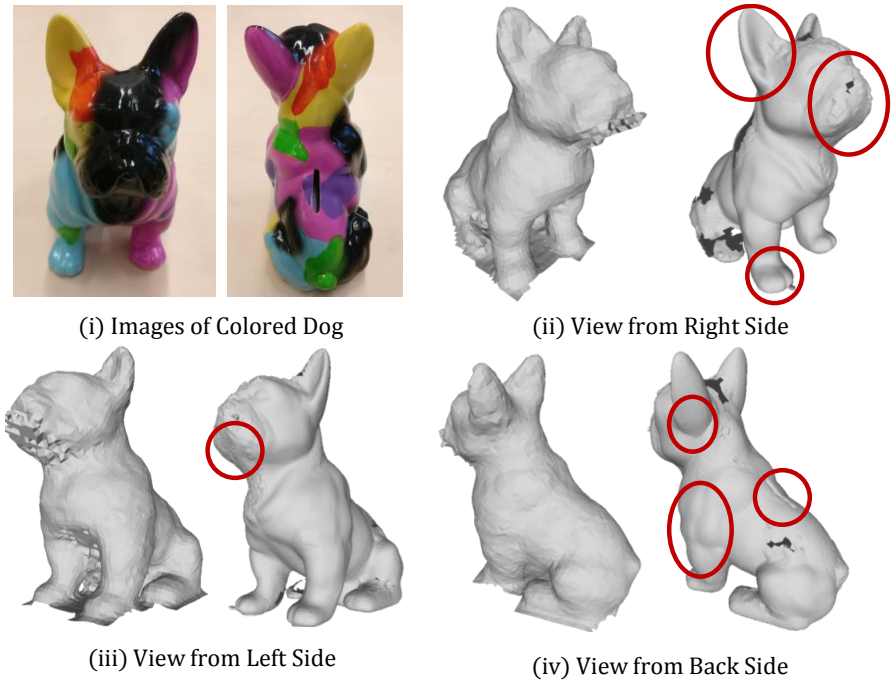
Figure 5.12: Improving KinFu reconstruction of Dino object.

device. In our algorithm, there is no need to perform a full fusion, and in all these examples we can operate with as low as 10 scans.

Effect of global optimization Due to discrepancies, the registration to CAD space is never perfect. When multiple scan registrations come together, they never align perfectly. Hence, we perform the CAD-free optimization as explained. In Fig. 5.14, we show visual results on the abilities of this global optimization stage. In this illustration, we show how a rough registration to the CAD model is refined to reveal the underlying geometry. The objects used are Caravan and Bird. The point clouds colored in the figure represent different scans automatically segmented from the scenes following the detection. It is noticeable that, after optimization, a better alignment is obtained, resulting in sharper and more uniform blend of the point cloud colors.

5.3.4 Dataset of Large Objects

Finally, we apply our pipeline to quality inspection of real gas turbine casings and large objects. In this real scenario, CAD models come directly from the manufacturer. Due to space constraints, we summarize the data modality in Tab.5.3. The manufactured

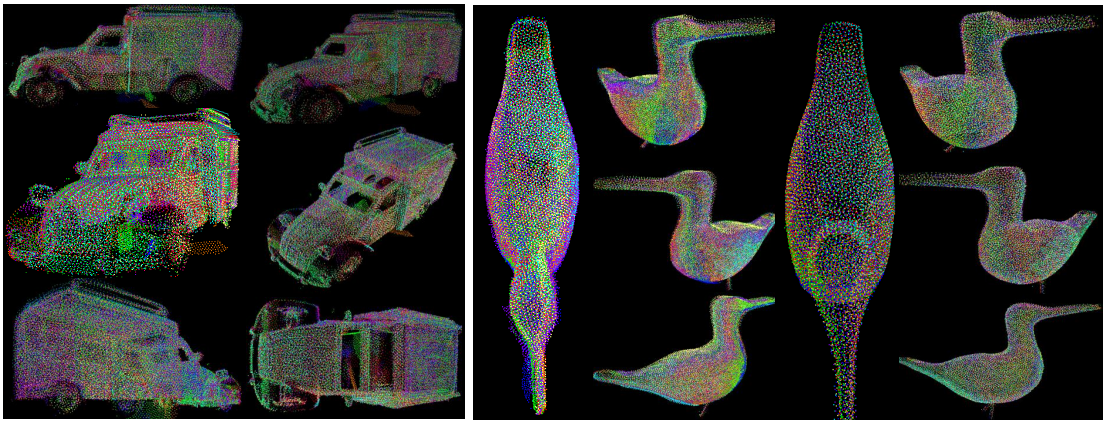


(i) Images of Colored Dog (ii) View from Right Side

(iii) View from Left Side (iv) View from Back Side

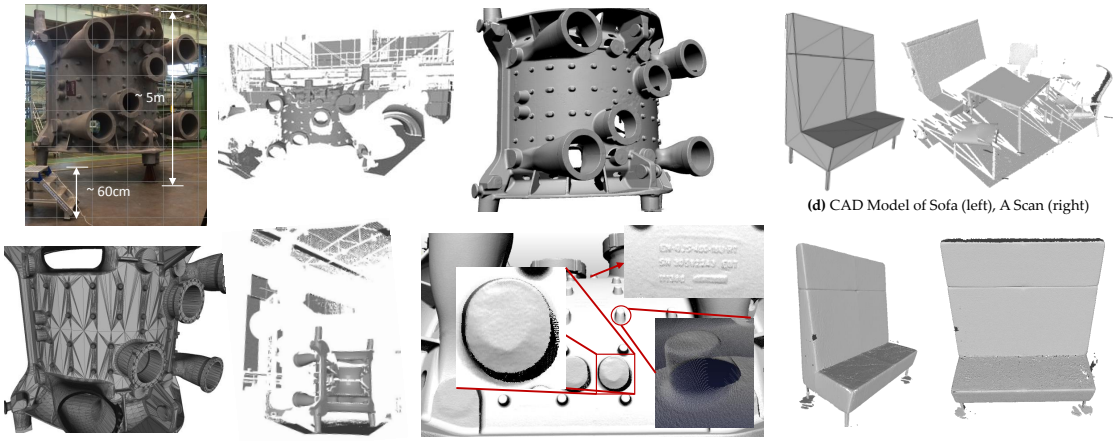
Figure 5.13: Improving KinFu reconstruction of Dog object.

parts deviate significantly from the ideal model due to errors in the process and we scan them in the production environment within clutter and occlusions. With such large objects and little resemblance of the CAD prior, obtaining ground truth becomes a challenging task. Thus, we use previously described photogrammetry (PG) and sparse reconstruction system [31, 34] to collect a set of accurate scene points. These points are found by attaching markers on the objects, capturing many images (see Tab.5.3) from different angles and running bundle adjustment. We treat the resulting structure as ground truth and run a deviation analysis. The mean errors obtained by CloudCompare [106] are plotted in Tab. 5.3, along with the running times of the individual stages. In Fig. 5.16 we also plot, for Ventil and Turbine objects, the histogram of errors in the CAD registration. The performance in objects of varying sizes indicate that our reconstruction method is applicable from small to large scale while maintaining repeatability. Fig. 5.15 presents further qualitative results on our reconstruction of the Turbine and Sofa objects, whereas Fig. 5.1 provides close-up shots from the reconstruction of Ventil object.



(a) Car w/o Optimization (b) Car with Optimization (c) Bird w/o Optimization (d) Bird with Optimization

Figure 5.14: Effect of global optimization on Caravan and Bird objects. Even though, in this example, the initial alignment was relatively close, the quality improvement in aligning multiple scans is still apparent. Different colors are assigned to different point clouds, segmented from the actual scenes.



(a) Turbine Object **(b)** Laser Scans **(c)** Final 3D Reconstructions of Turbine **(d)** CAD Model of Sofa (left), A Scan (right) **(e)** Reconstructions of Sofa

Figure 5.15: The reconstruction of Turbine **(a)** in captured cluttered scans **(b)** is presented in **(c)**. Similar results for the Sofa object are shown in **(d,e)**.

Table 5.3: Dataset statistics of large objects, average reconstruction errors w.r.t. photogrammetry (in mm) and timings.

Object	Scan Res.	Obj. Size	#Scans	#PG Images	PG vs CAD	Our Accuracy	Detect	Verify	Refine
Ventil	0.3 mm	$8m^3$	8	180	1.3cm	**2.2±0.4**	3.10s	0.27s	112.94s
Turbine	0.4mm	$125m^3$	10	180	3.4cm	**2.5±1.3**	3.72s	0.54s	126.13s
Sofa	1mm	$1.7m^3$	6	68	0.85cm	**1.4±1.2**	1.44s	0.31s	68.82s

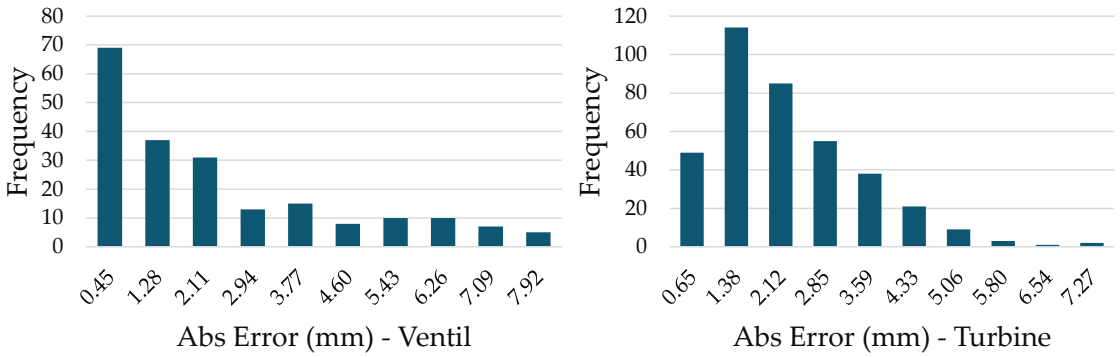


Figure 5.16: Evaluations against photogrammetry (PG) system. Our dense reconstruction is registered onto the sparsely reconstructed 3D points and deviations are recorded.

5.4 Conclusions and Moving Further

We have presented the *reconstruction-via-detection* framework, our transition from sparse to efficient, robust and automated dense 3D instance reconstruction from unconstrained point cloud scans. Our framework integrates probabilistic object detection, hypothesis verification, pose graph construction and multi-view optimization. Such a scheme allowed us to deal with problems of dynamic clutter, occlusion and object segmentation. Moreover, the computational cost is reduced, due to model-to-scan alignment. To the best of our knowledge, this is the first method, capable of reconstructing instances within clutter and occlusions, without explicit segmentation.

The rest of this thesis is devoted to explanation of the multiple components in this pipeline and their improvements, namely: point cloud sampling, re-meshing, object detection & pose estimation and pose graph processing. Advancements of these technologies can, in return, advance this pipeline, leading to better reconstructions. There are, though, still open issues such as handling rotational symmetries or next-best view prediction that we do not touch in the scope of this thesis.

CHAPTER 6

DOWNSAMPLING 3D CAD MODELS AND POINT CLOUDS

The initial stage of a majority of real applications is the pre-processing, in which the input data is converted to a more meaningful, or rather suitable representation. Our input consists of large point sets arising due to the prevalence of 3D data capture, and computerized models, which are composed of densely meshed parametric forms and primitives and which have immense use in industry. The *big-ness* of the data increase poses a natural challenge for the existing algorithms consuming 3D input. One way to go around this problem is *sampling*, in which a sparser representation of the 3D data is computed, to best achieve the goal at hand. In the context of this thesis, 3D sampling will mostly refer to the downsampling of a CAD-model or a point cloud.

3D geometry that is of concern in this thesis is mainly composed of point clouds and meshes (to represent CAD models). We admit noise-corrupted point clouds but assume noise-free CAD models. Yet, not all CAD models are designed for detection or registration, even if they are perfect for manufacturing. Many of the industry's man-made models include uneven distribution of triangles and their vertices are often found only around the critical points. There are also many invisible vertices, which are not easily accessible by the acquisition sensors. Such models cannot be used to identify or register the object. Thus, prior to any operation, we either remove the invisible vertices/faces, and sample the mesh geometry automatically or completely handle the invisible faces within sampling algorithms. Preparation of these models

for tasks such as reconstruction or object detection is of paramount importance. This chapter is devoted to describing the typical sampling techniques we use, as well as a new mesh to cloud sampling algorithm.

6.1 Re-meshing

In computer graphics, the state of the art to generate more suitable discrete representations is through *re-meshing* [9, 165], where a *better* mesh is obtained in terms of vertex sampling, regularity and triangle quality. A large body of the works in this category are variational, use Voronoi diagrams and concentrate even-ness of vertex distributions. A reasonably fast method with very satisfactory results is proposed by Levy *et al.* [165]. The method is based on intersection of *Restricted Voronoi Diagrams (RVD)* with the surface itself. Since the purpose of this work is not to present another re-sampling algorithm, we adopt theirs as it is. Given n discrete seed points, $\mathbf{X} = \{\mathbf{x_i} \in \mathbb{R}^{\mathbf{N}}\}_{\mathbf{i=1}}^{\mathbf{n}}$, the Voronoi diagram of $\mathbf{X}$ is defined as n Voronoi cells $\mathbf{V} = \{\Omega_i\}$, where

$$\Omega_i = \{\mathbf{x} \in \mathbb{R}^N : \|\mathbf{x} - \mathbf{x}_i\| \leq \|\mathbf{x} - \mathbf{x}_j\|, \forall\, j \neq i\} \tag{6.1}$$

Centroidal Voronoi Diagram (CVD) is a special form, where each seed $\mathbf{x}_i$ coincides with the center of mass $\mu_i = \int_{\Omega_i} \rho(\mathbf{x})\mathbf{x}d\sigma / \int_{\Omega_i} \rho(\mathbf{x})d\sigma$, where $\rho(\mathbf{x})$ is a density function. *Centroidal RVD* is then a type of *CVD*, which is obtained by restricting the Voronoi cells Ω_i on the surface $\mathbf{M}$, $\mathbf{R}_i = \Omega_i \cap \mathbf{M}$. We use *CRVD* to extract an isotropic dual triangle mesh. To do so, a uniformly distributed set of initial seeds $\mathbf{X}$ are generated randomly. Then, *CRVD* is computed via an optimization procedure, using [165]. After the optimization, the final mesh is extracted as the restricted Delaunay triangulation (*CRDT*). Finally, we re-correct the face and vertex normals of the resulting mesh using the rendering based method from Takayama *et al.*[257].

It is possible that remeshing algorithms create erroneous samples to satisfy the structural penalties. It is also probable that the sharp features are not preserved [165, 287]. On the runtime aspect, many re-meshing algorithms easily reach minutes, making them an overkill, when only point samples are desired. Moreover, naive re-meshing cannot distinguish invisible structures and is thus sub-optimal for vision tasks. This encourages us to further investigate mesh sampling when the desired output is a point cloud, as we present in §6.3.

6.2 Sampling Point Clouds

Random sampling (RS) RS is a basic sampling technique, where each 3D point is sampled by chance, independently. This strategy doesn't take into account the distribution of the points, it rather randomizes the selection itself.

Normal space sampling (NS) Normal Space Sampling [231] selects the points such that the resulting samples are uniformly distributed in the normal space. To do that, a set of bins $\mathbf{V} = \{V_1 ... V_N\}$ evenly distributed on a sphere S is generated by recursively dividing an icosahedron with 12 vertices into equally spaced N_s vertices. Then each point is assigned to a bin, resulting in a distribution of normals. Finally, we uniformly sample from this distribution, to draw points across different normals. However, this method is found to suffer from translational sliding by multiple scholars [103, 270].

Covariance sampling (Cov) Gelfand *et al.*[103] propose to sample the points that bind the rigid transformation the most by constraining the eigen-vectors of the covariance matrix of the torque and force. Let $(\mathbf{R}^*, \mathbf{t}^*)$ denote the optimal transformation:

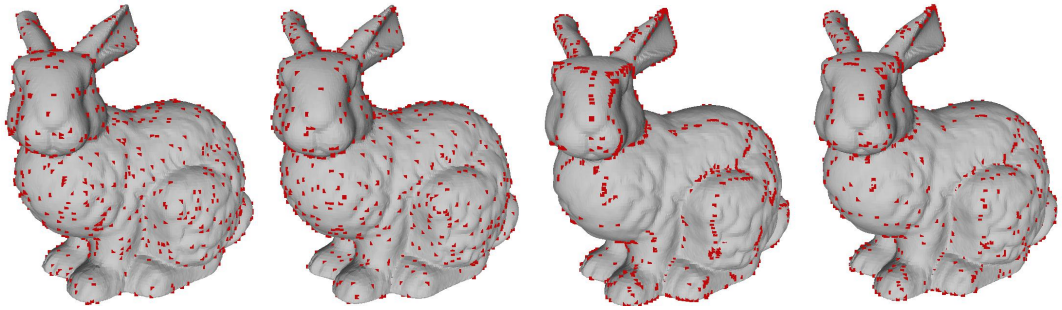
$$(\mathbf{R}^*, \mathbf{t}^*) = \underset{\mathbf{R},\mathbf{t}}{\arg\min}\, E_{pl} \tag{6.2}$$

If $\mathbf{R}^*$ is small (due registration in close vicinity) then one could perform Taylor expansion around $sin(\theta) \approx \theta$ and $cos(\theta) \approx 1$ to linearize the $\mathbf{R}$. This results in the energy:

$$E_{pl} = \sum_{i=1}^{N} \left((\mathbf{p}_i - \mathbf{q}_i)^T + \mathbf{r}^T (\mathbf{p}_i \times \mathbf{n}_i) + \mathbf{t}^T \mathbf{n}_i \right)^2 \tag{6.3}$$

where $\mathbf{r} = [\mathbf{r}_x, \mathbf{r}_y, \mathbf{r}_z]^T$. Then to solve Eq. 6.2, an overdetermined linear system $\mathbf{Ax} = \mathbf{b}$ could be assembled:

$$\mathbf{A} = \begin{bmatrix} \mathbf{p}_1 \times \mathbf{n}_1 & \mathbf{n}_1 \\ \mathbf{p}_2 \times \mathbf{n}_2 & \mathbf{n}_2 \\ \vdots & \vdots \\ \mathbf{p}_k \times \mathbf{n}_k & \mathbf{n}_k \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} (\mathbf{q}_1 - \mathbf{p}_1)^T \mathbf{n}_1 \\ (\mathbf{q}_2 - \mathbf{p}_2)^T \mathbf{n}_2 \\ \vdots \\ (\mathbf{q}_k - \mathbf{p}_k)^T \mathbf{n}_k \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \mathbf{r}_x \\ \mathbf{r}_y \\ \mathbf{r}_z \\ \mathbf{t}_x \\ \mathbf{t}_y \\ \mathbf{t}_z \end{bmatrix} \tag{6.4}$$



(a) RS (b) NS (c) Cov (d) Rel

Figure 6.1: Sampling from different strategies. 1000 samples are generated per each algorithm.

Here, the covariance matrix $\mathbf{C} = \mathbf{A}^T\mathbf{A}$, characterizes the change in error, when the surfaces are moved away from perfect alignment. Eigenvectors of $\mathbf{C}$ with small eigenvalues correspond to sliding directions. The idea is then to sample such that these small eigenvalues are constrained. For more details, we refer the reader to [103]. This method is designed specifically for algorithms, which solve the rigid body transformation, in the aforementioned fashion. Covariance sampling is known to generate clusters of point samples around the critical regions [228].

Relevance sampling (Rel) Torsello *et al.*[270] and Rodola *et al.*[228] propose to select the points with high distinctiveness. They associate distinctiveness to the local relevance, which is computed from the connectivity structure of the mesh. First, authors define an influence area as follows:

$$\mathbf{Y}_p = \{\mathbf{m} \in \mathbf{M} \colon \mathbf{n}_p^T\mathbf{n}_m > T \text{ and } \mathbf{p} \sim \mathbf{m}\}. \tag{6.5}$$

Here, $\sim$ shows the existence of a path from $\mathbf{p}$ to $\mathbf{m}$. Then, the relevance measure is defined as $f(\mathbf{p}) = |\mathbf{Y}_p|^{-k}$ with $|\mathbf{Y}_p|$ being the number of points in $\mathbf{Y}_p$. An inverse transform sampling is then conducted to select points, which are uniformly distributed over relevance. This method is shown to work well under several tasks, such as object recognition or detection. Note that this is the only method requiring mesh input described so far.

The results obtained by running different strategies are plotted in Fig. 6.1.

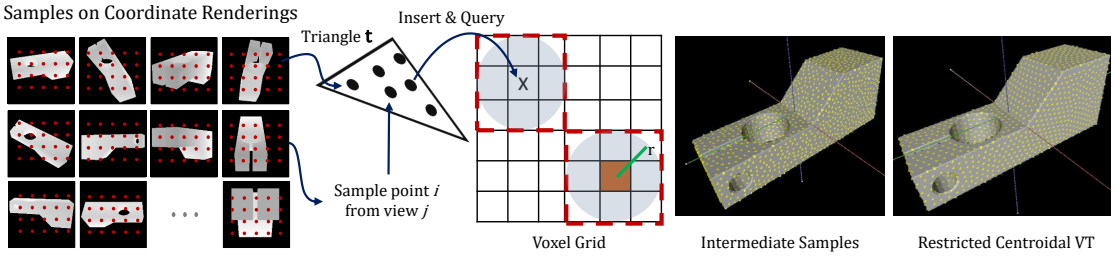


Figure 6.2: Samples generation. See text for details. CAD Model from MVTec Halcon: `http://www.mvtec.com`

Learning to sample While many of the aforementioned algorithms work well in practice, especially for the case of ICP registration, they cannot provide guarantees for the optimality of the result. In fact, we argue that finding a universally *good* sampling is difficult, if not impossible and rather propose, in [30], a task-tailored solution, where the sampling is obtained by running a genetic algorithm that gradually refines the estimate to a better solution. When the problem is picked to be 3D registration, a more acceptable solution to the discrete combinatorial problem of downsampling is found. The experimental evalution in of [30] clearly demonstrates that we are able to obtain better results than those state-of-the-art hand crafted strategies.

6.3 Sampling for PPF Based Object Detection

State of the art object/scene perception techniques treat the input model as a well behaving point cloud [35, 87, 246, 313]. While the research in keypoint extraction is numerous [220, 246, 267], these methods already operate on a reasonably well distributed vertex set, covering the entire object surface. These render the unstructured meshes or parametric forms very unfriendly for computer vision.

In this section, we address the particular problem of generating vision-compatible 3D point representations from irregular, non-conforming mesh geometries, amenable to object detection and registration. We contribute by designing a point resampling algorithm, aiding the Geometric Hashing framework of Drost *et al.*[87] and Birdal and Ilic [35], that is to be presented in detail in Chapter 7. Outcome of our method could certainly be used in other detection pipelines, but we particularly address this one because it inspired a broad range of research and is very practical due to the capability

of handling raw point clouds. The PPF matching relies on uniformly quantizing surfel features. Quantized values act like visual words and are used in retrieval of the model pose. This quantization step is shown to be the most critical source of error [129]. While Birdal and Ilic [36] tackle such problems via soft quantization, they do not devise methods for keypoint selection. With our method, we address the even sampling of both vertices and surface normals resulting in improved detection performance.

Our input is a 3D mesh. This is the modality we will refer as *CAD model*. Our method starts by presenting a multi-view rendering strategy to form an oversampling of the visible model part. Such sampling is computed by casting rays from all the pixels of all the views and intersecting them with the surface. This can be implemented efficiently by bounding volume hierarchies. A major challenge then stands out to globally unify the sampled views. One of our contributions is to prune and fuse/merge together the points in an effective way to create a bias-free, sparser output. We achieve this by using shallow trees for representing voxel grids. While, at this stage, desired output characteristics can be imposed by the practitioner, we specifically choose to constrain the minimum distances between randomly distributed samples (a.k.a. Poisson Disk Sampling) and the distribution of normals (a.k.a. Normal Space Sampling - NSS) because of the quantization necessity of PPF based geometric hashing [87]. With that, we are able to achieve bias-reduced blue noise (white noise with even spacing) characteristics, which is appealing for this and also other applications [297]. Moreover, for more regularity in the output, we employ a restricted Lloyd relaxation, in which the average disk radius is increased iteratively. With the introduction of this relaxation, blue noise characteristics can be traded-off to distant samples and regular structures. See Fig. 6.2 for a brief summary. Main contributions of this work are summarized as:

- We develop a mesh resampling method applicable to any mesh, regardless of the triangles being large, small, acute or elongated.

- We integrate view rendering to bias-reduced sample generation in order to gracefully remove the hidden/invisible geometries.

- We introduce an efficient sparse voxel based algorithm to address the global distribution of resulting vertices and normals using *Poisson Disk Sampling* and

Normal Space Sampling. We suggest to use the restricted Lloyd relaxation to balance the regularity and randomness.

- We present a GPU implementation for the costly parts of our algorithm.

Qualitative evaluations and spectral analysis show that we could generate visually appealing samplings with good theoretical properties. Quantitative assessments demonstrate that our algorithm can significantly boost object detection tasks. Our supplementary video can be viewed under `https://youtu.be/uQo535jQ52s`.

6.3.1 Method

Given an object model, we generate a set of synthetic cameras on a sphere encapsulating the object, as in Fig. 6.3(a) and cast rays for each view, from the camera center towards the origin. Each ray intersects the mesh, and creates a sample point and a normal at the intersection. We then collect all these samples and prune them using an efficient voxel grid, with Poisson constraints. Finally, with a Llyod relaxation, the sampling gains a balanced regularity. The entire procedure is summarized in Alg. 1 and illustrated in Fig. 6.2. Formally, given a mesh model $C = (\mathbf{M}, \mathbf{T})$, with vertices $\mathbf{M} = \{\mathbf{m}_1..\mathbf{m}_{N_m}\} \in \mathbb{R}^{N_m \times 3}$ and triangles $\mathbf{T} = \{\mathbf{t}_1..\mathbf{t}_{N_t}\} \in \mathbb{Z}^{N_t \times 3}$, we aim to generate the point cloud $\mathbf{P} \in \mathbb{R}^{N_p \times 3}$ and normals $\mathbf{N} \in \mathbb{R}^{N_p \times 3}$, s.t. sampled points obey uniform distribution in both 3D space and normal space.

Preprocessing Man-made CAD models might not have design constraints of orientation or positioning and can lie arbitrarily in space. To let the algorithm operate regardless the model positioning, a first step is to align the model to a canonical reference frame and scale. To do this, we first compute the covariance matrix of the vertices $\mathbf{C}$ and $\mathbf{v}$, the normalized eigen-vectors of $\mathbf{C}$: $\mathbf{C}\mathbf{v} = \lambda\mathbf{v}$. We fix an intermediate reference system as $(\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_0 \times \mathbf{v}_1)$. The model rotation aligns this new coordinate frame to the base frame $\mathbf{C}_0$. Next, the oriented bounding box (OBB) is computed as the axis aligned bounding (ABB) box in the transformed frame. Albeit not being a volume-minimizing BB, it is sufficiently accurate for our purposes. Finally, the mesh is rescaled to the ball with diameter $d = \sqrt{2}$. From here on, with the abuse of notation, the model $\mathbf{M}$ will be referring to this normalized mesh.

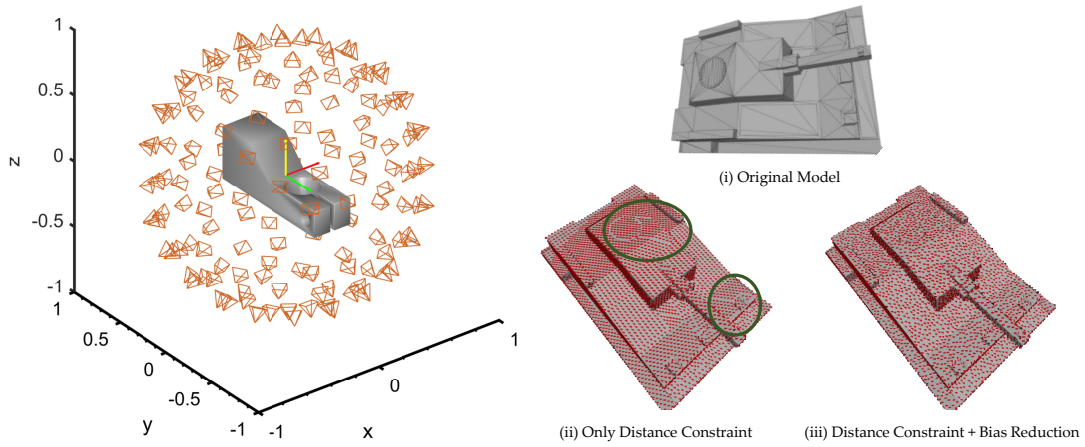
Multi-view setting We generate a set of camera poses (views) $\mathbf{V} = \{V_1 ... V_N\}$, uniformly distributed on a sphere S, obtained by subdividing the faces of a 12-vertex icosahedron into equally spaced N_s vertices as in Fig. 6.3(a). A pose is composed of a rotation matrix $\mathbf{R}$ and a translation vector $\boldsymbol{\xi}$: $\mathbf{Q}_i = [\mathbf{R}_i | \boldsymbol{\xi}_i]$, while $\mathbf{o}_i = -\mathbf{R}_i^T \boldsymbol{\xi}_i$ is the camera center. Moreover, for each pose, we maintain a set of intrinsic camera matrices $\mathbb{K} = \{\mathbf{K}_i\}$, according to the pinhole model. Let $\mathbf{f} = (f_x, f_y)$ be the focal length in pixels, and $\mathbf{c} = (c_x, c_y)$, the principal point. We set $c_x = w/2$, $c_y = h/2$, with (w, h), the desired resolution of the camera. As we synthesize the camera poses $\mathbf{Q}_i$ from sphere S, we are guaranteed to view the entire projection of the model, but we are not guaranteed to utilize the full resolution, unless $\mathbf{f}$ is tuned. To maximally use the viewport, we first set $\mathbf{f}$ to a relaxed initial value $\mathbf{f} = (f_0, f_0)$, and project the model. Given this projection, we compute a tightly fitting 2D bounding box and scale $\mathbf{f}$ accordingly as:

$$f^* = min\big(w/b_w, h/b_h\big) f_0 \qquad \mathbf{c}^* = 2\mathbf{c} - \mathbf{c}_b \tag{6.6}$$

where (b_w, b_h) are the dimensions and $\mathbf{c}_b$ is the center of the 2D bounding box. This way, the area of projection is maximized, while preserving the aspect ratio. Because the projected object silhouette is different in all views, $\mathbf{f}^*$ and $\mathbf{c}^*$ differ for each view, resulting in the set $\mathbb{K} = \{\boldsymbol{K}_i\}$.

Efficient ray-triangle intersection In this next stage, the samples projected on the camera views are backprojected and intersected with the 3D mesh itself.

Let $\mathbf{r} = \{\mathbf{o}, \mathbf{d}\}$ denote the ray with origin $\mathbf{o}$ and a normalized direction vector $\mathbf{d}$. Any point on this ray is then parameterized as $\mathbf{r}(\lambda) = \mathbf{o} + \lambda\mathbf{d}$. We then write the edges of the intersecting triangle as $\mathbf{E} = \{\mathbf{e}_o, \mathbf{e}_1, \mathbf{e}_o\}$ and express a triangle by a point and two edges $\mathbf{t} = (\mathbf{v}, \mathbf{e}_0, \mathbf{e}_1)$. The point of intersection $\mathbf{v}_{int}$ can be described using the 2D Barycentric coefficients (u, v) as $\mathbf{v}_{int} = \mathbf{p} + u\mathbf{e}_0 + v\mathbf{e}_1$. Using the famous Möller Trumbore algorithm [198], λ, u and v are obtained. The normal information of the sample p_{int} is then retrieved as the normal of the face: $\mathbf{n}_{int} = \frac{(\mathbf{p}_1 - \mathbf{p}) \times (\mathbf{p}_2 - \mathbf{p})}{\|(\mathbf{p}_1 - \mathbf{p}) \times (\mathbf{p}_2 - \mathbf{p})\|}$. Note that, since we are using the face normals, we do not have to carry out normal computation for each sample point. Instead, we could pre-compute all the triangle normals and reduce the normal computation to single look-up. If storage is a concern, one could always index the normals during runtime in a hashtable, and ensure single computation per face. The result here is a large sample pool $\mathbf{X}$ covering the CAD model.



(a) Cam Views (b) Illustration of Sampling Bias

Figure 6.3: **(a)** We synthesize camera views around the object as shown. **(b)** Demonstration of Poisson characteristics. Without bias reduction, artifacts caused by view discretization and intersection around the model edges are more visible. To exaggerate the effect, we used 160x120 image resolution.

Weighting samples Due to the viewpoint differences and the numerical accuracy of the ray-triangle intersection, not every selected sample has the same quality. Thanks to the ray casting, for each point $\mathbf{m}_j$ and intersecting triangle $\mathbf{t}_i$, we are able to weigh the samples. We first flip the normals $\{\mathbf{n}_i\}$ to point towards the camera, to get $\{\mathbf{n}_i^v\}$. The weight for the sample i is inversely proportional to the angle between the normal and the cast ray and is defined as:

$$w(\phi) = \left(1 + \exp(-\lambda(\phi - \mu_\phi))\right)^{-1} \qquad \phi = 1 - |\mathbf{n}_i^v \mathbf{d}_j| \tag{6.7}$$

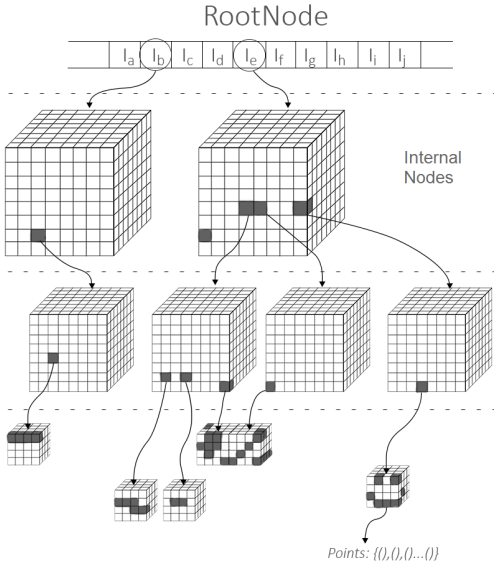
We use $\lambda = 10$ and $\mu_\phi = 0.5$. The maximum weight $w^{max} \approx 1$ is achieved when the vectors $\mathbf{n}_i^v$ and $\mathbf{d}_j$ are parallel ($\phi = 1$), while the minimum $w^{min} \to 0$, is obtained when the angle approaches 90° ($\phi = 0$). Note that, due to camera projection, after ±90° the face is not viewed, or viewed from the other side. Typically, we reject samples if the weight is found to be very low $w_i < \tau_w$. This lets us to choose the samples which are viewed in a fronto-parallel fashion. This procedure can handle open meshes, as the mesh normals provide directional information.

Algorithm 1: Proposed sampling algorithm.

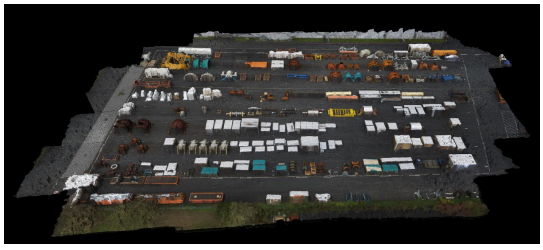
input : Mesh $(\{\mathbf{T}_i\},\{\mathbf{M}_i\})$, Threshold τ, Weight threshold τ_w and # Max Samples N_m
output: Sampled point cloud $\mathbf{D}$ with normals $\mathbf{N_D}$
Normalize and Align $\mathbf{M}$ to canonical frame
Generate camera poses : $\mathbf{V} = \{V_1 ... V_N\}$ $(\mathbf{S},\mathbf{N}) \leftarrow []$
// sample pool generation
for $V_i \in \mathbf{V}$ **do**
    Find best $\mathbf{K}$ via Eq. (2)
    Shoot rays: $\mathbf{r}_i(\lambda) \leftarrow \mathbf{o} + \lambda \mathbf{d}_i$
    $\{(\mathbf{v}^i_{int}, \mathbf{n}^i_{int})\} \leftarrow \{\mathbf{r}_i\} \cap (\mathbf{T},\mathbf{M})$
    Compute $\mathbf{w}(\phi)$ via Eq. (3)
    Exclude vertices with $\mathbf{w}(\phi) < \tau_w$
    $(\mathbf{S},\mathbf{N}) = (\mathbf{S},\mathbf{N}) \cup (\{\mathbf{v}^i_{int}\},\{\mathbf{n}^i_{int}\})$
Randomize$(\mathbf{S},\mathbf{N})$
Compute CDF
$R_d \leftarrow diameter(\mathbf{S})$
$(\mathbf{D},\mathbf{N}_D) \leftarrow []$
$cnt \leftarrow 0$
// prune and merge
for $cnt < N_m$ **do**
    $i \leftarrow$ find$(CDF,$random$(0,N))$
    $(\mathbf{s},\mathbf{n}) \leftarrow (\mathbf{S}_i,\mathbf{N}_i)$
    $d_{min} = \min_{(\mathbf{t}\in\mathbf{D})} |\mathbf{s}-\mathbf{t}|$ **if** $(d_{min} > \tau R_d)$ **then**
        $\mathbf{D} \leftarrow \mathbf{D} \bigcup \mathbf{s}$
        $\mathbf{N}_D \leftarrow \mathbf{N}_D \bigcup \mathbf{n}$
    $cnt \leftarrow cnt + 1$
Apply Lloyd relaxation on $\{\mathbf{D},\mathbf{N}_D,\mathbf{T}\}$

GPU implementation For the systems with graphics support, the ray-triangle intersection as well as the weighting can be implemented by rendering the coordinates of intersection on a 3 channel (RGB) image. This can be computed in Shader. Furthermore, the triangle ID per pixel (sample point) can be stored in alpha channel. This reduces the GPU-to-CPU transfer of entire information to a single RGBA texture.

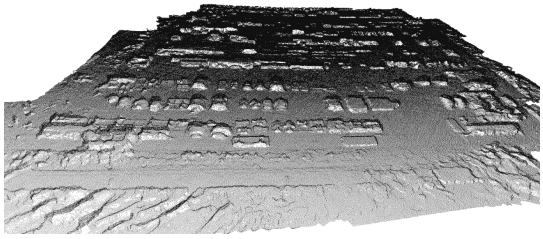
Merging view based samples Given all the ray intersections, we are left with a set of 3D points $\{\mathbf{p}_i\} \in \mathbf{X}$ per each view, which are to be fused into the full 3D sample cloud $\{p_i\} \in \mathbf{P}$. Some of these points $\mathbf{p}_i$ could be duplicates across views, or even if not, they will be found very close (due to quantization errors). Moreover, a satisfactory



(a)



(a) Dense drone reconstruction of ~200m factory with ~26M vertices



(b) Sampled Surface using τ=0.001

(b)

Figure 6.4: (a) Sparse voxel representation. (b) Very large surface mesh and its sampling using the sparse voxels.

distribution of points over the object surface is not yet achieved. Our goal is then to prune this large *sample pool* **X**, subject to certain constraints. Note that, at this stage, the desired task-based 3D sampling characteristics can be also be enforced. Because typical object detection algorithms [35] rely on equidistant/even sampling, as well as local surface characteristics (such as normals), we adapt two strategies: We employ Poisson disk sampling for distance based constraints, and normal-space sampling to enforce a uniform distribution of local surface characteristics. Our approach fuses these into a single sampling strategy, which we will devise below.

Poisson disk sampling (PDS) Recently, PDS is found to be particularly robust for PPF matching [35], due to its generation of even distances and good spectral properties. For our case, it will also help in reducing the bias, caused by discrete sampling on the views. Fig. 6.3(b) illustrates this effect. PDS tries to obtain uniform random points based on a minimum distance criterion between the samples. Formally, it tries to satisfy the following two conditions:

$$\forall \mathbf{p}_i \in \mathbf{P}_i, \forall \Omega \in D_{i-1}, P(\mathbf{p}_i \in \Omega) = Area(\Omega)/Area(D_{i-1}) \tag{6.8}$$

$$\forall(i, j : i \neq j), \|\mathbf{p}_i - \mathbf{p}_j\| > 2r \tag{6.9}$$

where D is the domain of the sample pool and D_{i-1} denotes the available (not-yet-sampled) domain. The first condition (Eq. 6.8 - Poisson Sampling) states a uniformly distributed sample $\mathbf{p}_i$ falls in subdomain Ω with likelihood, proportional to the area of Ω, provided that $D_{i-1} \cap \Omega = \varnothing$. Due to the computational complexity of calculating the region of sampling, we relax the first constraint. We associate each sample $\mathbf{p}_i$ to a sphere centered at $\mathbf{p}_i$ with radius r. r is the same for all the samples. The probability $P(\mathbf{p}_i \in \Omega)$ is then computed as $P(\mathbf{p}_i \in \Omega) = 1 - \sum_{j=1}^{i} \alpha(j) 4/3\pi r^3$, where $\alpha(j)$ influences a poisson process bound in the interval $(0, 1]$, influencing the percentage of the feasible sampling space for the point i. Instead of using $\{\alpha(j)\}$ as in a Poisson process, we simply replace it with its expected value $\bar{\alpha}$, reducing to $P(\mathbf{p}_i \in \Omega) = 1 - 4/3K\bar{\alpha}\pi r^3$. By that, we are allowed to retrieve samples from the pool, sequentially with equal likelihood, i.e. the drawing is uniformly distributed.

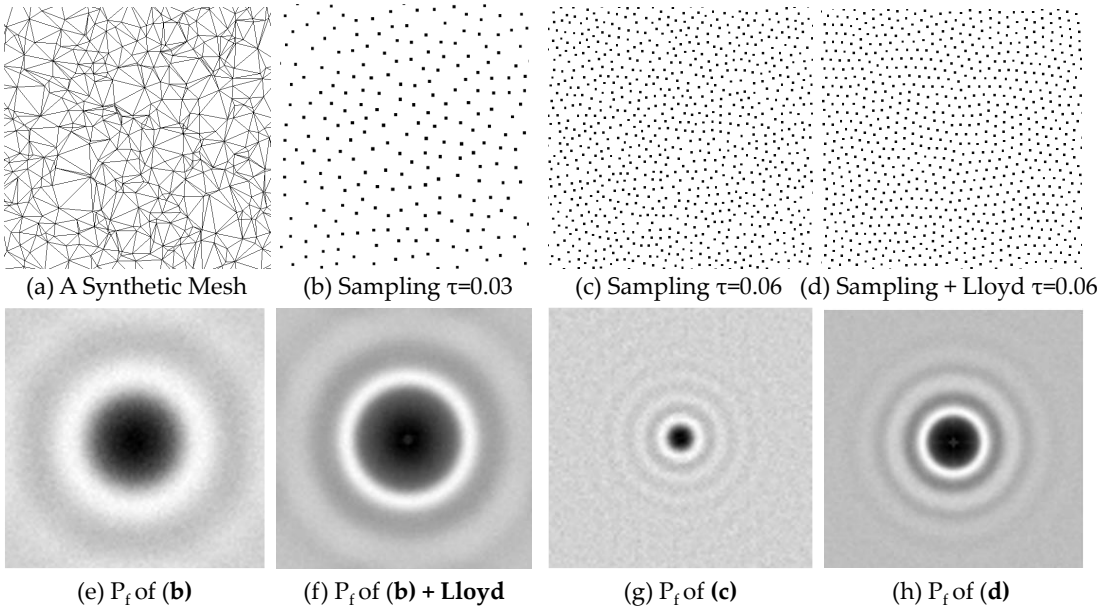
Satisfying the 2^{nd} condition (Eq. 6.9 - Disk Sampling) is more trivial but required to be made efficient for a large number of samples. The main idea is to draw samples from the pool $\boldsymbol{X}$ iteratively and check against the existing samples $\mathbf{P}_{i-1}$, for the violation of Eq. 6.9. If Eq. 6.9 is satisfied, the sample is accepted. A naive implementation involves a search through all the so-far sampled points or marking all the neighbors of a sample as *rejected*. Both are computationally demanding. We take a different path and construct a 3D voxel grid G over the existing samples. We then take a sample $\mathbf{p}_i$ from the pool sequentially and insert it into G. If the sample satisfies Eq. 6.9, it is kept, otherwise rejected. The side length of the grid is tuned such that the search ball remains within 9 voxels, and the query can be done in $O(1)$ time, enabling us to complete the entire sampling in $O(N)$. Even though this procedure is greedy - as it depends on the first sampled point- it is found to generate good distribution in practice.

Sparse voxel representation For large radii (less samples), the search in dense voxel grids will be fast. However small radii, where closer points are sampled are problematic due to the exploding memory of the grid. For that reason, we propose to use a sparser voxel-grid similar to [204]. Our tree structure resembles the one in a B+ tree. By construction, the tree is height-balanced, shallow and wide. This decreases the

number of operations for traversing the tree from root to a leaf node. The structure implemented has 1 root node, 2 internal layers and a leaf layer as shown in Fig. 6.4(a). Point information is stored only at the leaf layer, wheres the internal layers maintain a bitmask for encoding active children based on their spatial coordinates. To increase traversal speed, the whole structure is restricted to powers of two: First layer has a size of 8 units per dimension, second layer 4 and leaf nodes 4. A *Cache* that holds the last visited internal and leaf nodes is also implemented. The resolution of the grid is automatically adjusted to $2r$. This way, given a point, we can answer the question *should this point be sampled?* in constant time and can therefore downsample very large point clouds such as the one in Fig. 6.4(b).

Normal space sampling Rusinkiewicz and Levoy [231] propose to sample points such that the normals are uniformly distributed. We adapt this into our pipeline by altering the order of the considered points. We first quantize all normals into a dense set of bins $\{\bar{n}_j\}$ and assign each normal to the closest bin. We then take the random ordering specified in Section IIIg, and compute the cumulative distribution function $CDF = F(\bar{n}_i)$. After that, a scalar γ is drawn uniformly in the interval $[0, F(\bar{n}_i)]$ and the smallest index i s.t. $F(\bar{n}_i) > \gamma$ is computed, using binary search. Since this can be done subsequent to the random sampling, it has little effect in introducing regularity, while still satisfying the uniform distribution of normals.

Lloyd relaxation Due to the introduced randomness and greediness of the sampling algorithm, the sampled points are not regular. To obtain a good balance between blue noise property and regularity, we conclude our sampling by applying a few Llyod iterations, in which the centers of the points are shifted to the centers of the Voronoi diagram, gradually. While, it is easy to apply this on synthesized samples (samples not on a specific surface), ensuring that the Voronoi centers remain on the surface is difficult. For that, we exploit Restricted Centroidal Voronoi (R-CVD) iterations, efficiently implemented by [298]. R-CVD operates by intersecting (restricting) the VD with the surface. Lloyd scheme is formulated as a variational energy minimization and a quasi-newton approximation is made for fast convergence. Because minimum distance constrained is roughly satisfied in the previous section, this scheme enjoys a good initialization and only a few iterations are enough for pleasing results.



(a) A Synthetic Mesh (b) Sampling τ=0.03 (c) Sampling τ=0.06 (d) Sampling + Lloyd τ=0.06

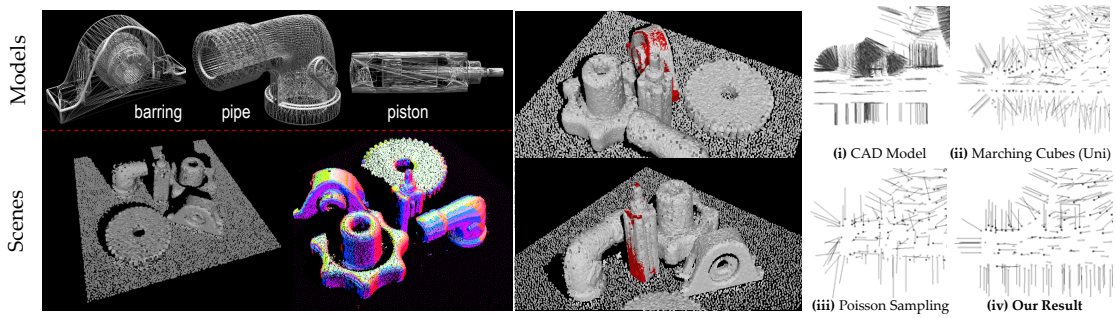
(e) P_f of **(b)** (f) P_f of **(b) + Lloyd** (g) P_f of **(c)** (h) P_f of **(d)**

Figure 6.5: Spectral analysis. **(a)** Original Mesh. **(b,c)** Sampling with different radii. **(d)** Sampling + Lloyd relaxation. **(e)** Power spectrum plot of (b). **(f)** Spectral plot of (b) after 10 Lloyd steps. **(g)** Spectral plot of (c). **(h)** Spectral plot of (d)

6.3.2 Results and Evaluation

Spectral analysis Since our sampling exhibits blue noise characteristics, we use frequency domain analysis to evaluate the quality. Lagae and Dutre [160] standardize this analysis as a power spectrum study. First, we generate 50 different synthetic 2D meshes as shown in Fig. 6.5 and sample them with our method. For each sampling, the periodogram is computed. These periodograms are averaged to estimate the power spectrum $P(f)$. We plot these 2D spectra in Fig. 6.5 with a logarithmic tone map, removing the high-magnitude DC component. $P(f)$ reveals the typical *blue noise* properties: The central DC peak is surrounded by an annulus of low energy, followed by a sharp transition region, a low-frequency cutoff and a flatter high-frequency region. As a result, inter-sample distances follow a certain power law, with high frequencies being more common. Note that our sampling preserves these spectral properties.

Real dataset and parameters As our method best performs with industrial CAD models in mesh forms, we utilize the Toshiba dataset, proposed in [218]. This dataset,



(a) Our Dataset **(b)** Typical Detection Results **(c)** Sampled Normals

Figure 6.6: **(a)** Views of Toshiba dataset. Upper row: Wireframe CAD models and internal structures; Lower row: A scene point cloud and its normals. **(b)** Models detected by [87] using our sampling. **(c)** Normals on sample points compared. Notice our sampling (iv) can generate normals which are closest to their original CAD counterparts (i).

shown in Fig. 6.6a, consists of real life wireframe 3d models and scanned point clouds as scenes. The models have internal structures, elongated triangles and sharp edges, making them hard to work with. We modify this dataset to include normals for 20 scenes, each scene containing multiple objects at different occlusion levels. For all the experiments in this part, a set of fixed parameters are used: $\tau = 0.03$; image resolution is 640x480; the weight function/threshold are shown in Fig. 6.9(a) and we use 2 subdivisions of the sphere, resulting in 162 camera poses viewing the object.

Base methods We compare our approach against 4 other methods, which are capable of converting meshes into point sampled surfaces. Our baseline is set to be the *Monte Carlo (MC)* sampling (a uniform random sampling) within the facets. A more proper method is the *stratified sampling* or triangle subdivision method [7], in which each triangle gets a sample depending on the resolution of an underlying voxel grid. A mainstream and successful approach is *Poisson disk sampling* [69], also used in [35]. Finally, we compare our method against a *uniform mesh resampling*, which consists of building a uniform volumetric signed distance field and applying a marching cubes to get uniformly distributed samples. This is also very similar to the method proposed in [193]. Note that a large family of 3d keypoint extraction methods operate directly on vertices [117, 313] and cannot be applied to Toshiba dataset or this application.

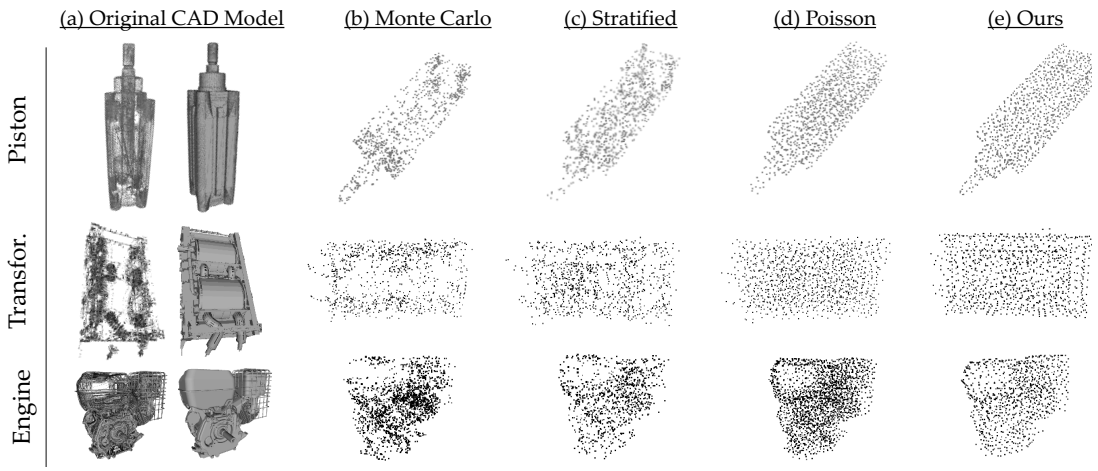


Figure 6.7: Visuals of our point resampling. Columns depict different samplings of the same object, while the rows contain different objects. First two columns show the mesh and the vertices of the original point sets.

Qualitative evaluations We assess the visual quality of our method on Piston object from the Toshiba dataset and everyday objects of chair, glasses and a tree, as well as on two additional industrial CAD models of a gasoline engine and a large transformator. All models have complicated triangle arrangement, hidden faces and structural connections. The vertices are unevenly distributed and clustered on certain support areas. Figures 6.7 and 6.8 presents the results of sampling 1000 points by our algorithm and preceding methods. It is noticeable that our method better preserves the global shape, has even distribution of vertices and can get rid of internal structures. Moreover, thanks to the blue noise characteristics, the discretization artifacts due to the views are not visible. These result in an enhanced perceptual quality of the generated point sets. The closest result to ours is from Poisson sampling, but as it doesn't treat hidden faces or surface normals, our method remains visually superior. We also visualize the normals estimated with different techniques as well as ours in Fig. 6.6c. Here, one should look for the set of directions, which resemble the CAD model's the most. Because we sample directly on the faces, the normal estimation quality of ours is also the closest to the original CAD model among the algorithms we evaluate.

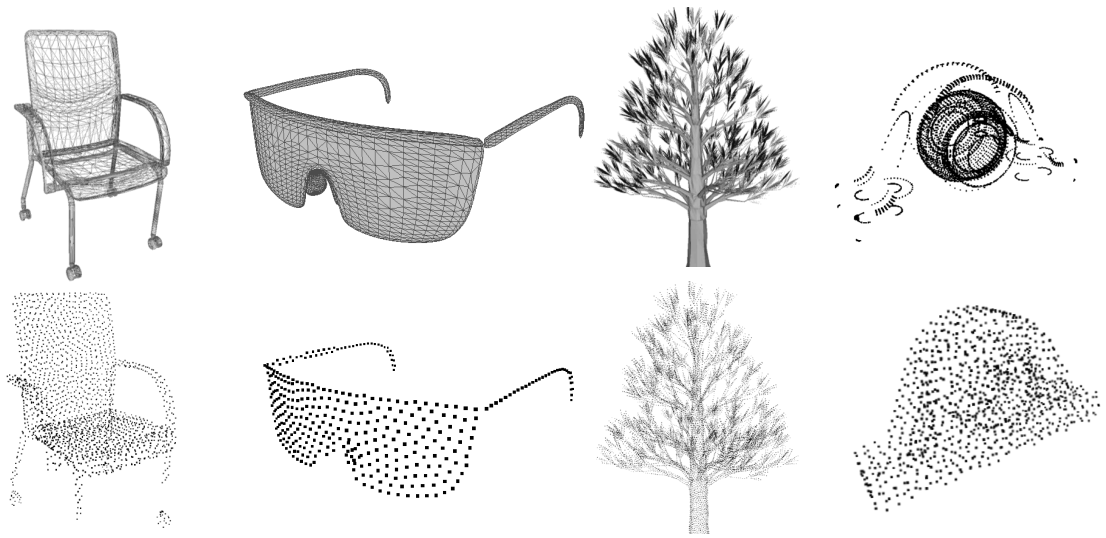
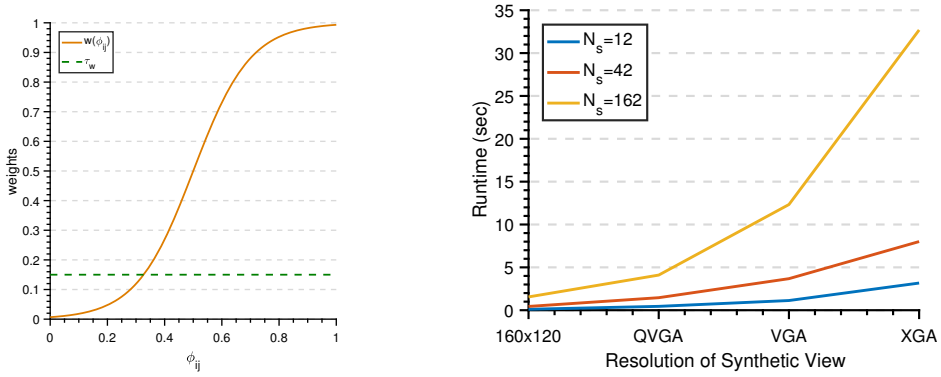


Figure 6.8: Additional visuals of our point resampling. Upper row: The original models, lower row: corresponding sampled models.

Application to object detection We now assess the effectiveness of the method in the challenge of 3D detection and pose estimation, the reason of its design. We use the PPF method of [35, 87] as explained earlier, on the Toshiba dataset [218]. The only modification is that, we replace the weighting scheme of [35] with the weights computed in § 6.3.1. The evaluations are done by substituting the existing sampling of [87] with a different one and comparing the pose estimation results. Our findings for different number of samples and sampling algorithms are plotted in Fig. 6.10. We provide distinct plots for the rotational ($\mathscr{E}_{\mathbf{R}}$) and translational ($\mathscr{E}_{\mathbf{t}}$) error components. A detection is said to be correct if it can be refined to the correct ground truth pose by a subsequent ICP (iterative closest point) alignment. Depending on the pose with which the object lands on the scene, the tolerance of ICP could differ and we use the same setting across all methods. We only accumulate the pose errors when the correct detection is spotted. Fig. 6.6b illustrates these refined poses. We plot the number of correct detection per each sampling in the left-most column of Fig. 6.10. It is consistently visible from this figure that choosing evenly spaced points on the visible surface with uniform distribution of normals helps our algorithm to detect more objects and make less error in the pose when the object is detected. This means that we manage to sample more task suited points which have higher probability



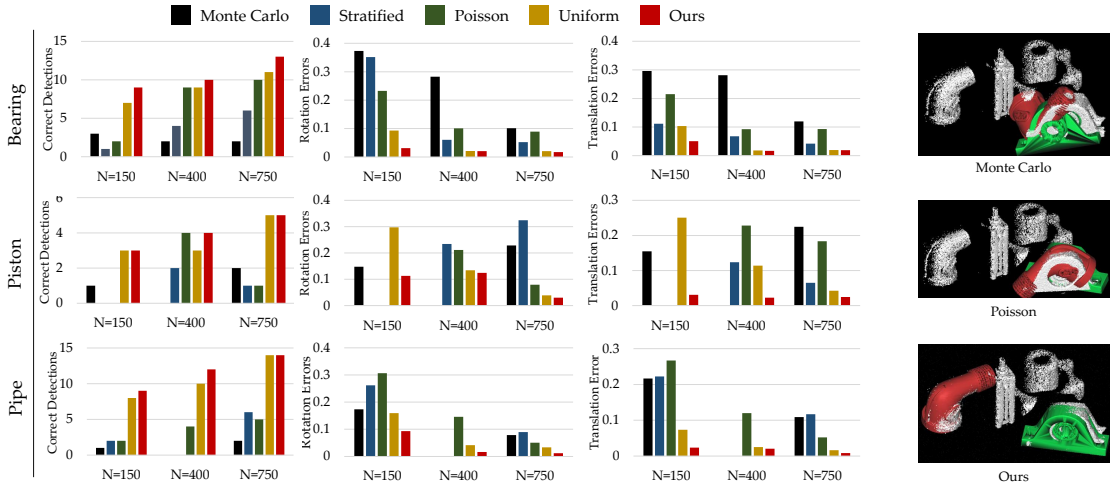
(a) (b)

Figure 6.9: **(a)** Used weighting function (see Section IIId). **(b)** Computational timings of CPU implementation.

of being seen and have a better coverage of the entire surface - as for [87], the pose depends on the point found to be on the surface.

The SDF based sampling of [193] is the closest to ours in terms of pose estimation performance. However, as the number of samples decrease, the performance gap grows. This is because our samples are always located on visible primitives and therefore are better suited to perception/detection tasks. We have also noticed that due to voxel-grid used in Marching Cubes, the memory requirements of [193] become intractable as the model size increases. Our algorithm, on the other hand, doesn't suffer from this and can sample models of arbitrary size (see Fig. 6.4(b)).

Computational time We implemented our algorithm on an Intel i5 2.3GhZ CPU. For ray casting, we use the freely available Intel Embree library [293], carefully optimized for Intel platform. The average timings on our dataset are plotted in Fig. 6.9(b). The most important parameter for runtime is the resolution of the synthetic views as we cast a ray for each pixel of the view. The number of faces of the CAD model has a diminished effect due to the apriori spatial indexing. Our GPU implementation, on the average, could only run twice as fast and is dominated by the transfer overhead of the textures. Note that the view merging is always computed on the CPU, to ease the implementation complexity.



(a) Quantitative Results (b) Detection

Figure 6.10: **(a)** Pose errors, for different number of samples N, computed for correct detections only and are averaged. Our sampling could enable the same detection method to obtain more correct results with lower pose errors, regardless the sample count. **(b)** Scene from Toshiba dataset. Detection results by different sampling algorithms are visualized on this point cloud.

6.4 Further Remarks

We have presented the commonly used 3D sampling algorithms and proposed a new holistic strategy that unites many desired properties for the utilization of CAD models in existing state of the art computer vision pipelines. This algorithm is practical, easy to implement and improves the PPF matching described in Chapter 7. Our algorithm allows us to directly compute a randomized, evenly spaced point sampling of the visible portion of the surface. The normal computation and point quality are well integrated into the schema and are always present in the sampled result. We evaluated our approach qualitatively and quantitatively on a real dataset, and demonstrated the boost in the matching and pose estimation tasks.

Further applications and experiments presented in this thesis utilize the proposed sampling method whenever the CAD models are available. For point sets, we only use the *prune and merge* stage (after line 17 of Alg. 1), that is fast and, albeit a heuristic, still guarantees uniform spatial point distribution.

CHAPTER 7

DETECTION OF CAD MODELS AND THEIR COMPONENTS IN 3D POINT CLOUDS

Many computer vision applications require finding the object of interest in either 2D or 3D scenes. The objects are usually represented with the CAD model or object's 3D reconstruction and the typical task consists of the detection of the particular object instance in the scenes captured with RGB/RGBD/depth cameras. Detection considers determining the location of the object in the input image, usually denoted by the bounding box. However, in many scenarios, this information is not sufficient and a complimentary 6DOF pose (3 degrees of rotation and 3 degrees of translation) is also required. This is typical in robotics and machine vision applications. Consequently, the joint problem of localization and pose estimation is much more challenging due to the high dimensionality of the search space. In addition, objects are often sought in cluttered scenes, under occlusion and illumination changes. In addendum, close to real-time performance is usually required. An alternative to estimating the 6DoF object pose holistically considering the entire object as a rigid body, is to assume that the models are composed of certain basis functions, often referred as 3D primitives. If such primitives are to be detected then they could be assembled together to reconstruct the 3D shape, and if the canonical model is available, a pose can be computed.

In this thesis, we focus on general scenarios where the scenes are described by 3D

point clouds, without color or additional attributes. We assume that one or multiple (not necessarily pinhole) cameras are used in acquisition, and individually, those cameras can output point cloud scenes in their own coordinate frame. Under these non-restrictive assumptions, this chapter covers detecting either instances or bases of CAD models in such scenes. Not incorporating color allows us to alleviate the problem of illumination changes and lets us use a wider range of sensors, such as Lidar only.

3D matching started naturally by applying the know how in the field of 2D computer vision directly to 3D [244]. Later, hand-designed features crafted particularly for 3D, has emerged [63, 247, 267, 307, 313]. These methods operate by extracting a set of 3D keypoints and descriptors from range images or point clouds and matching them to the ones obtained from scenes. For matching either RANSAC-like scenarios, or KD-trees are employed. Unfortunately, while many such local features exist, repeatability is by no means comparable to the one achieved in 2D and thus, the work in this field is far from being complete [151, 268]. This gap has motivated us to design detectors accustomed to our problems, such as reconstruction. Therefore, we first explain our efforts in 6D pose estimation. We approach this problem both through geometric methods and via learning. In the final stage, we will also describe how primitives of these CAD models can be found in cluttered and occluded captures.

7.1 Building upon Point Pair Features and Geometric Hashing

One of the most promising algorithms for matching 3D models to 3D scenes was proposed by Drost *et al.* [87]. In that paper, authors couple the existing idea of point pair features, with an efficient voting scheme to solve for the object pose and location simultaneously. Given the object's 3D model, the method begins by extracting 3D features relating pairs of 3D points and their normals. These features are then quantized and stored in a hash table and used for representing the 3D model for detection. During run-time stage, the same features are extracted from a down-sampled version of a given scene. The hash-table is then queried per extracted/quantized feature and a Hough-like voting is performed to accumulate the estimated pose and location, jointly. In order to overcome complexity of the full 6DOF parametrization, assumption

is made that at least one reference point in the scene belongs to the object. In that case if the correspondence is established between that reference point in the scene and one model point there, and if their normals are aligned, then there is only one degree of freedom, rotation around the normal, to be computed in order to determine the object's pose. Based on this fact, a very efficient voting scheme has been proposed. The great advantage of this technique lies in its robustness in presence of clutter and occlusion. Moreover, it is possible to find multiple instances of the same object, simply by selecting multiple peaks in the Hough space. While operating purely on 3D point clouds, this approach is fast and easy to implement.

Due to its pros on the performance aspect, aforementioned geometric matching method immediately attracted attention of scholars and was plugged into many existing frameworks, often surpassing the learning based counterparts [130]. Moreno *et al.*used it to constrain a SLAM system by detecting multiple repetitive object models [234]. They also devise a strategy towards an efficient GPU implementation. Another immediate industrial application is bin picking, where multiple instances of the CAD model is sought in a pile of objects [131]. Besides, there is a vast number of robotic applications [23, 210] where this method has been applied. The original method also enjoyed a series of add-ons developed. A majority of these works concentrated on augmenting the feature description to incorporate color [66] or visibility context [152]. Choi *et al.*proposed using points or boundaries to exploit the same framework in order to match planar industrial objects [67]. Drost *et al.*modified the pair description to include image gradient information [84]. There are also attempts to boost the accuracy and performance of the matching, without touching the features. Figueiredo *et al.*made use of the special symmetric object properties to speed up the detection by reducing the hash-table size [78]. Tuzel *et al.*proposed a scene specific weighted voting method by learning the distinctiveness of the features as well as the model points using a structured SVM [277].

Unfortunately, despite being well-studied, method of Drost *et al.* [87] is often criticized by high dimensionality of the search space [44], being sensitive to 3D correspondences [171], having performance drops in presence of many outliers, and low density surfaces [197]. Furthermore, the succeeding works report to significantly outperform the technique in many datasets [44, 128]. Yet, these methods work with RGB-D data, cannot handle occlusions and heavily depend on the post-processing

and pose refinement.

In defense of the point pair features, we propose a revised pipeline along with a new perspective on quantization and retrieval. We will first explain our pipeline, where we address the crucial components of the framework. Instead of targeting the specific part of the original method as others, we revise the whole algorithm and draw a more elaborate picture of an improved object detection and pose estimation method. In the next section, we will devise our probabilistic view on the pose retrieval.

7.1.1 PPF Matching Pipeline Revisited

Our approach starts by generating more accurate model representation relaying on PPFs. Since the normals are integral part of the PPFs, we compute them accurately by a second order approximation of the local surface patches. Giving different importance to the PPF is also important in building more reliable model representation. Unlike [277] where scene dependent PPF weighting has been performed, we rely on ambient occlusion maps [194] and associate weights to each model point, obtained via visibility queries over a set of rendered views. This is scene independent and causes a cleaner Hough space, eventually increasing the pose accuracy. During the online operation, the scene (depth map) is first segmented into multiple clusters, in a hierarchical fashion. In our context, coarse-to-fine/hierarchical segmentation refers to a set of partitioning varying from under- to over-segmentation. We detect objects in all segments, separately. Note that, while a variety of methods also segment the 3D model and use the parts [161], we deliberately avoid this, because the proposed matching is already robust to clutter and occlusion, which would be present in distinct clusters. By introducing a hierarchy of depth segment clusters with varying sizes, we deal with the segmentation errors. Processing disjoint segments inherently reduces the clutter and thus, the voting space gets much cleaner. We can then have a better detection rate, with a more accurate pose. Thanks to the same reason, we can detect small objects as well as large ones. This also improves the ability to find multiple objects without cluttering the Hough space. These benefits come with no additional computational cost. In fact, choosing reasonable segment sizes often reduce the run-time.

Our voting scheme makes effective use of the computed model weights and an enhanced Hough voting to achieve further accuracy of poses with more correct detec-

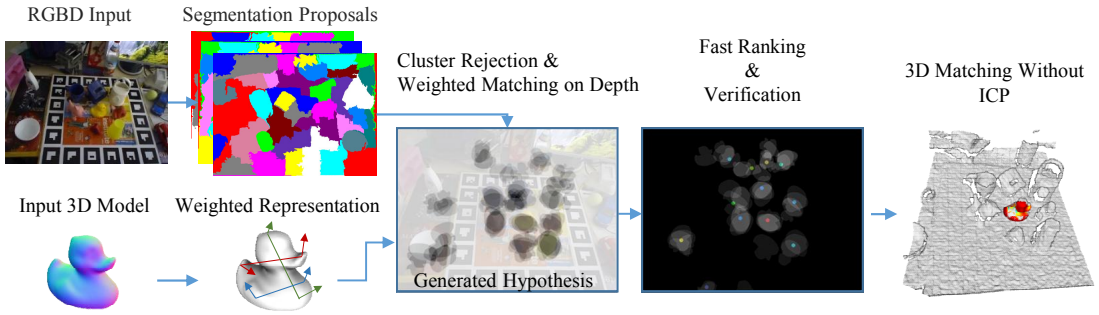


Figure 7.1: Illustration of the proposed pipeline. First input CAD models are trained using ambient occlusion maps as weighting. Each captured scene is first segmented into smaller regions and each region is matched to trained model. Per each segment, we retain many hypothesis and verify using the rendered CAD model. We rank the hypothesis according to the scores and reject the ones with low confidence.

tions. Finally, all the estimated hypotheses in all segments are gathered and checked through an occlusion and clutter aware hypothesis verification. Moreover, thanks to entire procedure, the necessity of ICP pose refinement is minimized, further speeding up the real life applications. To accomplish all this, neither the feature representation nor the matching scheme is altered. This way, all the other methods, benefiting the similar framework can enjoy the contributions.

We evaluate our approach quantitatively and qualitatively on both synthetic and real datasets. We demonstrate the boosted pose accuracy along with the improvements in detection results and compare it to the state of the art. We show that the proposed pipeline yields more accurate poses, an increased detection rate and reduced complexity. The following sections are devoted to the related work, description of the proposed method and experiments.

Weighting model points The original technique of Drost et al [87] treats all sampled points equally. Similar to [277], we argue that not all points carry the equal importance for matching. However, while authors in [277] are performing a scene dependent weighting and learning for a given task, we emphasize the necessity of a scene independent one. Unlike [277] however, our goal is not only to improve the detection rate, but to get better pose accuracy as well. For that, we are trying to focus on the visible surfaces of the object, where the normals are accurate and repeatability is bet-

ter. Consequently, we base our weighting strategy on ambient occlusion maps [194] as explained in §2.8. Based on $A_{\mathbf{p}}$, we propose to weigh the entries of the hashtable. Thus, given the hashtable bins, our weights are nothing but a normalized, geometric mean of $A_{\mathbf{m_r}}$ and $A_{\mathbf{m_i}}$. This way, the likelihood of using a potentially hidden point is reduced. In the experiments section, we show that even though this weighting doesn't necessarily increase the detection rate, it improves the accuracy of the resulting pose.

Global model description Given the extracted PPF, the global description is implemented as a hashtable mapping the feature space to the space of point pairs. To do this, the distances and the angles are sampled in steps of d_{dist} and $d_{angle} = 2\pi / n_{angle}$ respectively. These quantized features are then used as keys to the hashtable. The pair features, which map to the same bucket are grouped together in the same bin, along with the weights. To reduce the computational complexity, a careful downsampling is required at this stage, which would respect the quantization properties. This requires all the points to have at least d_{dist} distances. We found out that using a Poisson Disk Sampling algorithm [69], this is ensured to an acceptable extent. This algorithm consists of generating samples from a uniform random distribution where the minimum distance between each sample is $2r$. This suggests that, a disk of radius r centered on each sample does not overlap any other disk, satisfying our quantization constraint.

Online matching Our input in runtime is only a depth image, typically acquired by a range sensor. First, the required normals are computed using SRI method proposed by Badino *et al.*[21]. This choice is motivated by the grid structure of the range image and the availability of the camera matrix. While not being identical to model normals, they are both accurate and computed quickly. The scene is then downsampled in a similar fashion to model creation. The triangulated depth points are then subject to a voting procedure, over the local coordinates. This section is devoted to the description of a coupled segmentation and voting approach, together with pose clustering and a hypothesis verification.

Having a fixed scene point pair $(\mathbf{s_r}, \mathbf{s_i})$, we seek the optimal model correspondence $(\mathbf{m_r}, \mathbf{m_i})$ to compute the matching and 6DOF pose. Unfortunately, due to quantization, ambiguities and the noise in data, such assignment cannot be found by a simple scan. Instead, a voting mechanism, resembling Generalized Hough Transform is conducted.

While votes can be cast directly on 6DOF pose space, Drost *et al.*[87] proposed an efficient scheme, reducing the voting space to 2D, using local coordinates. Whenever a model pair, corresponding to a scene pair is found, an intermediate coordinate system is established, where $\mathbf{m_i}$ and $\mathbf{s_i}$ are aligned by rotating the object around the normal. The planar rotation angle α_m for the model is precomputed, while the analogous for the scene point α_s is computed online. The resulting planar rotation angle around x-axis is found by a simple subtraction, $\alpha = \alpha_m - \alpha_s$.

An accumulator Acc is 2D voting space composed of $\mathbf{m_r}$ (the model index) and α. It collects votes for each scene reference point. $\mathbf{m_r}$ is already a discrete entity, while α is a continuous one, subject to discretization over the voting space. Unlike original method, we also maintain another accumulator Acc_α retaining the weighted averages of the corresponding α values, for each bin in Acc. This is done for the sake of not sacrificing further accuracy.

Notice that, because the pose parameters and the model correspondence are recovered simultaneously, an incorrect estimation of one, directly corrupts the other. This makes the algorithm sensitive to noisy correspondences. To compensate for these artifacts, we propose to vote with the computed weights in §2.8. Moreover, in presence of significant noise, correct correspondences can fall into neighboring bins, decreasing the evidence. Thus, when voting, the value of each bin is added also to the closest bins. Subsequently, we perform a subpixel maximization over the continuous variable α by fitting a second order polynomial to the k-nn of the discrete maximum and use α_k, obtained from the weighted averaging of the corresponding bins.

Matching disjoint segments Our method employs a pre-segmentation to partition the scene into different meaningful clusters. Each cluster is then processed separately, having distinct Hough domains. This is different than previous works like [161], in which the model is also segmented.

We treat the depth image as an undirected graph $G = \{V, E\}$, with vertices $\mathbf{v}_i \in V$ and edges $(\mathbf{v}_i, \mathbf{v}_j) \in E$. As a dissimilarity measure, each edge has a non-negative weight $w(\mathbf{v}_i, \mathbf{v}_j)$. We then seek to find a set of components $\mathbf{C} \in \mathbf{S}$, where $\mathbf{S}$ is the segmentation. The component-wise similarity is achieved via the weights of the graph. Felzenszwalb and Huttenlocher propose a graph theoretic segmentation algorithm, addressing a similar problem [94]. Their approach is designed for RGB images, whereas we adapt it

to depth images. The algorithm uses a pair-wise comparison predicate (P), which is defined as:

$$P(\mathbf{C}_1, \mathbf{C}_2) = \begin{cases} 1, & \text{if } d(\mathbf{C}_1, \mathbf{C}_2) > M_{int}(\mathbf{C}_1, \mathbf{C}_2) \leq 0 \\ 0, & \text{otherwise} \end{cases} \tag{7.1}$$

Here, $d(\mathbf{C}_1, \mathbf{C}_2)$ is the difference between components and defined as the minimum weight edge:

$$d(\mathbf{C}_1, \mathbf{C}_2) = \min_{\mathbf{v}_i \in \mathbf{C}_1, \mathbf{v}_j \in \mathbf{C}_2, (\mathbf{v}_i, \mathbf{v}_j) \in E} w(\mathbf{v}_i, \mathbf{v}_j) \tag{7.2}$$

where the minimum internal difference M_{int} equals:

$$M_{int}(\mathbf{C}_1, \mathbf{C}_2) = \min(\text{Int}(\mathbf{C}_1) + \tau(\mathbf{C}_1), \text{Int}(\mathbf{C}_2) + \tau(\mathbf{C}_2)) \tag{7.3}$$

with $\text{Int}(\mathbf{C}) = \max_{e \in MST(\mathbf{C}, E)} w(e)$ and MST being the minimum spanning tree of the graph. The threshold function $\tau(\mathbf{C}) = k/|\mathbf{C}|$ exists to compensate for small components with k being a constant and $|\mathbf{C}|$, the cardinality of $\mathbf{C}$. Note that, smaller components are allowed when there is a sufficiently large difference between neighboring components. The segmentation $\mathbf{S} = \{\mathbf{C}_1 \dots \mathbf{C}_s\}$ can be efficiently found by union-find algorithm. The adaptation to depth images is done by designing the weights. We use the local smoothness of the surface normals along with the proximity of the neighboring points. Segmentation weights are defined as:

$$w(\mathbf{v}_i, \mathbf{v}_j) = \|\mathbf{v}_i^n - \mathbf{v}_j^n\| \angle(\mathbf{n}_i, \mathbf{n}_j) \tag{7.4}$$

where $(\mathbf{v}_i^n, \mathbf{v}_j^n)$ is the edge in the normalized coordinates.

While this approach generates a descent segmentation, we do not need to process every segment. In fact, many of these segments might lack sufficient geometry, or can be very small / large, or be coplanar. For that, we apply a filtering. We first remove segments which have a lot of undefined depth values. Then, the segments not obeying the size constraints are filtered out. Finally, we evaluate the linearity of the segments. Because, we have the set of normals $\{\mathbf{N}_j^i\}$ defined for each point $\mathbf{n}_j$ of cluster i, this procedure is simply applying a threshold τ_c over the deviation from the mean normal, computed as:

$$\sigma(\mathbf{N}^i) = \frac{1}{|\mathbf{N}^i|} \sum_{j=1}^{|\mathbf{N}^i|} (\angle(\mathbf{n}_{ij} \bar{\mathbf{n}}_i))^2 \tag{7.5}$$

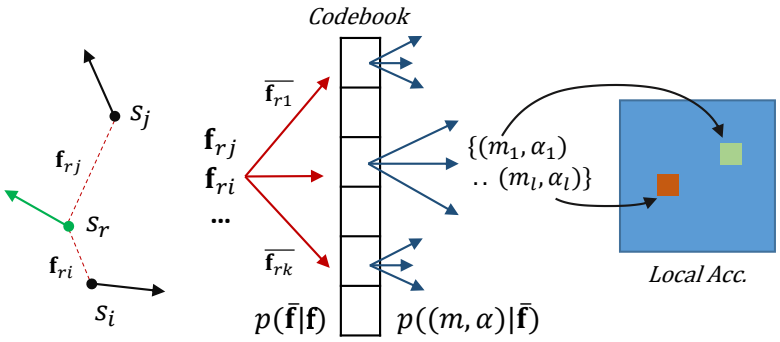


Figure 7.2: Local implicit voting: Given multiple scene point pairs, tied to a common reference $\mathbf{s}_r$, we generate features $\mathbf{f}_{ri}$, activating different codebook buckets (middle). Each bucket casts votes for multiple (m, α) pairs in the local voting space of $\mathbf{s}_r$.

where $\{\bar{\mathbf{n}}^i\}$ is the set of mean cluster normals (see Chapter 2). The clusters which satisfy the condition $\sigma(\mathbf{N}^i) < \tau_c$ are early-rejected. In our experiments, we use a coarse-to-fine (under to over) set of segmentations. Worst case resorts to using the whole scene, while difficult cases, such as small objects are found in coarser levels. Each segment is processed disjointly. We then verify the detected poses as in §7.2. This reduces the clutter and decreases the number of scene points sought. Because less clutter implies more relevant votes, it *demystifies* the Hough space and eases the maximization. Then, the accuracy of the resulting pose, as well as the detection rate increases. Besides, reduction in computational cost comes as a by-product. We will discuss this more in § 7.3.1.

7.1.2 Probabilistic Point Pair Retrieval

A codebook view of model training In the first stage, we generate a pose invariant *codebook* encoding all possible semi-global structures that could be found on the CAD model. A semi-global information in this regard, is obtained by a point pair feature $\mathbf{f}$. We use our codebook to relate a feature (key) to a set of oriented references points $\{(\mathbf{m}_i, \mathbf{m}_j)\}$ (stored in buckets) and build the global model description as an inverted file i.e. a hashtable $\mathbf{H}$. Thus, each bucket in the codebook contains self-similar point pairs extracted from the CAD model. We now cast the detection and pose estimation problem as retrieving/matching the correspondence of an oriented scene point pair $\{\mathbf{s}_r, \mathbf{s}_i\}$ to the one on the model $\{\mathbf{m}_r, \mathbf{m}_i\}$. Whenever a pair from the scene is matched

to one in the model, their normals at the reference points are aligned. Then, the full pose of the object can be obtained once the rotation angle α around the normal is known. This can be done by aligning Local Coordinate Frames (LCF) constructed from matched pairs. Thus, instead of storing the full PPF, we store only this local parameterization $\{\mathbf{m}_r, \alpha\}$ composed of the model reference point $\mathbf{m}_r$ and rotation angle α. A pair correspondence resolves the full 6DOF pose and what is left is to retrieve the matching pairs $\{\mathbf{s}_r, \mathbf{s}_i\}$ and $\{\mathbf{m}_r, \mathbf{m}_i\}$. We now give a novel way to do this.

Probabilistic formulation During detection, a new point cloud scene $\mathbf{S}$ is encountered and downsampled to a set of points $\mathbf{S}_D = \{\mathbf{s}_r\}$, some of which are assumed to lie on the object. The sampling also enforces spatial uniformity (see our suppl. material). We fix a reference point $\mathbf{s}_r$ and pair it with all the other samples $\{\mathbf{s}_i\}$. Each pair makes up a PPF $\mathbf{f}_{ri}$. The original method [87] associates $\mathbf{f}_{ri}$ to a unique key and can not account for the quantization errors that inevitably happen due to the noise. To circumvent these quantization artifacts, resulting from the hard assignment in [87], we quantize $\mathbf{f}_{ri}$ to K different bins ($K > 1$), activating different codebook entries as in ISM. This soft quantization results in possibly matching buckets $\bar{\mathbf{F}}_{ri} = \{\bar{\mathbf{f}}_1..\bar{\mathbf{f}}_K\}$. $\bar{\mathbf{F}}_{ri}$ indexes the buckets of $\mathbf{H}$, with weights $p(\bar{\mathbf{f}}_k|\mathbf{f}_{ri})$. For each matching bucket, we collect the valid interpretations $p(m, \alpha|\bar{\mathbf{f}})$, inversely proportional to the size of the bucket N_b, denoting the probabilities of particular pose configuration, given the quantized feature. Formally:

$$p(\mathbf{m}, \alpha|\mathbf{s}_r, \mathbf{s}_i) = p(\mathbf{m}, \alpha|\mathbf{f}) = \sum_k p(\mathbf{m}, \alpha|\bar{\mathbf{f}}_k)\, p(\bar{\mathbf{f}}_k|\mathbf{f}) \tag{7.6}$$

with $p(\bar{\mathbf{f}}_k|\mathbf{f}) = \frac{1}{K}$ and $p(m, \alpha|\bar{\mathbf{f}}) = \frac{1}{N_b}$ being uniformly distributed. This probability is actually the prior on the PPF of the particular object and can be computed differently accounting for the nature of the object geometry using a suited distribution. At this point, the gathered pair representations for a particular scene reference point are sufficient to recover for the object pose. However, due to outliers, some of these matches will be erroneous. Therefore, a 2D voting scheme is employed, locally for each scene reference point $\mathbf{s}_r$. The voting space is composed of the alignment of the LCF α as well as the model point correspondence $\mathbf{m}$:

$$V(\mathbf{m}, \alpha) = \sum_i p(\mathbf{m}, \alpha|\mathbf{s}_r, \mathbf{s}_i) \tag{7.7}$$

For each $\mathbf{s}_r$, there is a voting space $V_r(\mathbf{m}, \alpha)$, from which the best alignment is extracted as:

$$(\mathbf{m}_r^*, \alpha_r^*) = \arg\max_{\mathbf{m},\alpha} V_r(\mathbf{m}, \alpha) \tag{7.8}$$

Each such $(\mathbf{m}_r^*, \alpha_r^*)$ corresponds to a pose hypothesis. This is similar to performing Generalized Hough Transform (GHT) on reference point level locally and is the reason why we attribute the name Local ISM to our method. After all pose hypotheses are extracted, as the maxima in the local spaces, the poses are clustered together to assemble the final consensus, further boosting the final confidence.

7.2 Hypothesis Verification

Hypotheses verification for depth images Our method generates a set of hypotheses per each object, with reasonable pose accuracy. Yet, such a huge set of hypotheses demands an efficient verification scheme. Typical strategies, such as Hinterstoister *et al.*[128], either put ICP in the loop, whereas, for our method, the pose accuracy is sufficient for ICP-less evaluations.

Whenever a depth image is available as a dense projection of the 3D content, in order to verify and rank the collected hypothesis, we categorize the visible space into 3: Clutter (outlier) S_c, occluders S_o and points on the model S_m according to the following projection error function:

$$E_h(\mathbf{p}, m) = D_{\mathbf{p}} - \Phi(\mathbf{p}|\mathbf{M}, \mathbf{\Theta}_h, \mathbf{K}) \tag{7.9}$$

Φ selects the projection of the model points $\mathbf{M}$ corresponding to pixel $\mathbf{p}$, given a camera matrix $\mathbf{K}$ and the pose parameters $\mathbf{\Theta}_h$ for hypothesis h. The classification for a given valid point $\mathbf{p}$ is then conducted as:

$$\mathbf{p} \in \begin{cases} S_m, & \text{if } |E_h(\mathbf{p}, m)| \leq \tau_m \\ S_o, & \text{if } -E_h(\mathbf{p}, m) \geq \tau_o \\ S_c, & \text{otherwise} \end{cases} \tag{7.10}$$

subsequently, the score for a given hypothesis is:

$$S_h = \left(1 - \frac{|\mathbf{p} \in S_o|}{N_m}\right) \cdot \frac{|\mathbf{p} \in S_m|}{N_m - |S_o|} \tag{7.11}$$

where N_m is the number of model points on valid region of the projection $\Phi(p|\mathbf{M},\mathbf{\Theta}_h,\mathbf{K})$. The thresholds τ_m and τ_o depend on the sensor and are relaxed, due to the missing points not acquired by the sensor. Similarly, we include the check for coinciding normals. Luckily, these scores can be computed very efficiently using vertex buffers and Z-buffering on the GPU. Instead of transforming the model with the given pose, we use the $\mathbf{\Theta}^{-1}$ to update the current camera view. Thanks to the accuracy in pose estimation, the ICP is not a strict requirement of this stage. In fact, frequently, the verification is ICP-free. Retrieving the top N_{best} poses finalizes our object detection pipeline.

This metric favors less occluded and less cluttered matches, having more model points with consistent normals. Yet, in our experiments, we found that use of filtered clusters increases the chances of hypothesizing a descent pose, which is only at seldom missed by the verification.

Hypotheses verification and rejection for 3D data Devised matching theoretically generates a pose hypothesis for each scene reference point, which is assumed to be found on the model. There are typically $\sim 400-1000$ such points, reducing to 50 poses after the clustering, where the close-by poses are grouped together and averaged. Still, as many hypotheses as the number of clusters remain to be verified and the best pose is expected to be refined. In our problem of instance reconstruction it is critical that no false positive pose hypotheses survives. For this reason we introduce a rigorous hypothesis verification scheme. The effective verification requires fine registration, while efficient registration requires as few poses as possible. This creates a chicken and egg problem. We address this issue via a multi-level registration approach. In the first stage, sparsely sampled scan points are finely registered to the model using the efficient LM-ICP [97] variant of Iterative Closest Point (ICP) registration [28]. We also build a 3D distance transform for fast nearest neighbor access. Our sparse LM-ICP requires only 1ms per hypothesis, allowing us to verify all the hypotheses. We define the hypothesis score to be:

$$\Xi(\boldsymbol{\theta}_i) = \frac{1}{N_M}\sum_{j}^{N_M}\begin{cases}1, & \|\boldsymbol{\theta}_i^{-1}\circ\mathbf{m}_j - \mathbf{s}_k(j))\| < \tau_\theta \\ 0, & otherwise\end{cases} \tag{7.12}$$

where $\boldsymbol{\theta}_i$ is the pose hypothesis and $\mathbf{s}_k(j)$ the closest sampled scene point to transformed model point $\boldsymbol{\theta}_i^{-1}\circ\mathbf{m}_j$. Intuitively, this score reflects the percentage of visible

model points. The surviving poses are then sorted, taken to the next level and densely refined. This coarse to fine scheme is repeated for 3 levels of the pyramid. Finally, a dense registration is performed to accurately obtain the final pose.

Until this stage the surface normals are excluded from the fine registration process. We do this intentionally, to use them as a verification tool. Following registration, we check the surface consistency between the scene and the model. To do so for each scene point, the surface normal of the closest model point is retrieved. A scan is only accepted if a majority of the normals agree with the model. While this procedure can result in potentially good detections being removed (due to scene deviations), it does not allow false positives to survive as shown in §5.3.

7.3 Evaluating Object Detection and Pose Estimation

7.3.1 Evaluating Improved Pipeline

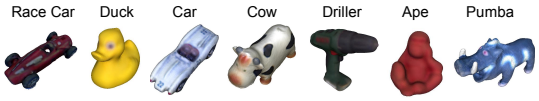


Figure 7.4: 3D models of some of the objects used in our experiments.

The detection performance of a basic variant of our method has already been proven to be robust on this dataset [87]. We now evaluate our method quantitatively and qualitatively on synthetic and real datasets. For real sets, we use the common LineMOD dataset [128] and the Mian dataset [193]. LineMOD dataset [128] is now a standard in object detection and many early works evaluated their methods on it. The package includes 15 non-symmetric objects appearing in ~ 1100 scenes per object. Each scene of an object is cluttered with the other objects. In none of the scenes, the objects are subject to heavy occlusion. We use only a subset of the models in our experiments as shown in 7.4. Specifically, our objects are *phone*, *ape*, *duck*, *iron*, *driller*, *car* and *benchwise*. The chosen models cover a variety of geometrical structures. We select a subset due to either lack of accurate CAD models or large performance drops for LineMod [128] (which would be unfair to show). For all experiments, the points are downsampled with a distance of 3% times the diameter. Normal orientation is sampled for $n_{angle} = 45$.

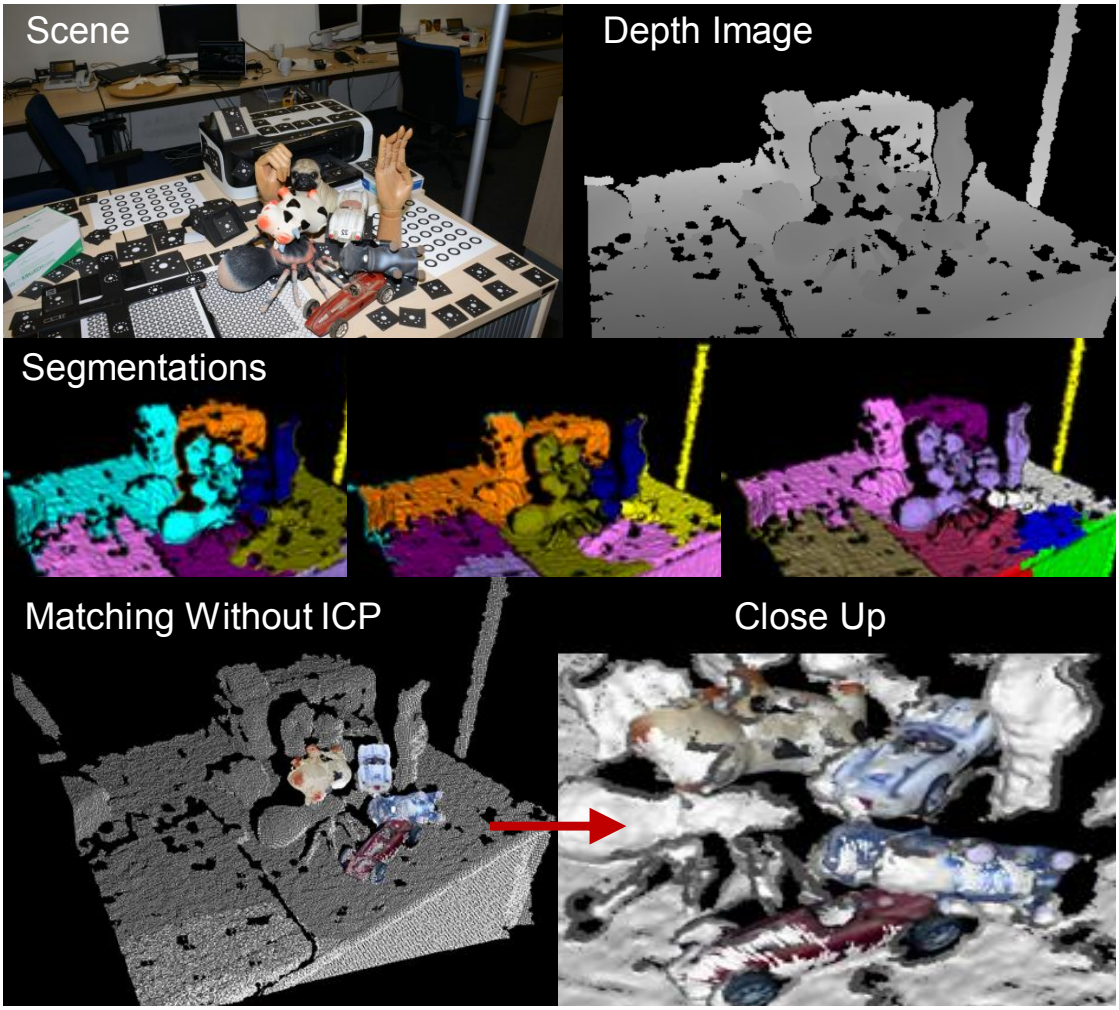


Figure 7.3: Outputs from our approach. Our segmentation aided matching has improved detection rates along with accurate poses.

Synthetic experiments First, we synthetically evaluate the accuracy of our pose estimation. To do that, we virtually render multiple CAD models in 3D scenes along with artificial clutter, also generated by other CAD models. To match the reality, our models are the reconstructions of real objects taken from LineMOD dataset [128]. We synthesize 162 camera poses over the full sphere for each object. This corresponds to 1134 point cloud scenes, all of which had apriori additive Gaussian noise. Because at this point we are concerned for the pose accuracy, no segmentation is applied and we record the rotational and translational errors for correct detections. At this stage,

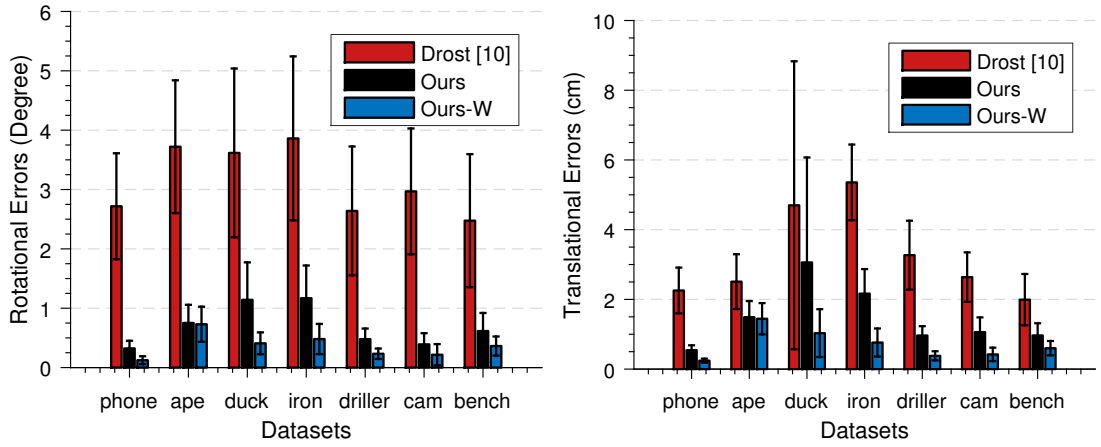


Figure 7.5: Comparison of the voting strategies (Ours-W is the weighted variant). **a)** Rotational errors. **b)** Translational errors.

an object is marked detected if the resulting pose is close to the ground truth pose. We set the threshold to 10% of the object diameter for detection and 10° for rotation. Fig. 7.5(a) and 7.5(b) depict the results obtained from pose estimation. It is seen that, for many of the objects, our rotational component is twice as more accurate as Drost et al [87]. Since the translation is also computed from rotations and the matching model component, there is also similar refinement in translational accuracy.

Evaluations on real data We compare our method against LineMod [128], and Drost *et al.*[87], that are the baselines for our dataset and method respectively. Yet, it is worth paying attention to the following: LineMod [128] is designed for multi-modal features and incorporates color information. Yet, we favor a fair comparison to our method by only using depth cues. Naturally, this has a negative impact on LineMod's performance, as it relies significantly on the color information. We, nevertheless, include LineMod as a baseline. Unlike the experiments in [128], we do not tune the parameters for each object or scene when using our method. While carefully tuning parameters sacrifices computational time for detection accuracy, it is cumbersome and unfair. LineMod without ICP has very low detection rates, as ICP is an integral part of LineMOD pipeline. Thus, we use ICP in the detection stage for LineMOD. Yet, the reported poses are solely obtained by template matching and not ICP. In the

Table 7.1: Detection results on ACCV3D for different objects.

	LineMod	Drost *et al.*	Ours
ape	42.88%	65.54%	**81.95%**
cam	68.78%	84.92%	**91.00%**
cat	35.62%	87.30%	**95.76%**
driller	51.52%	81.06%	**81.22%**
iron	35.22%	87.06%	**93.92%**
Average	46.80%	81.18%	**88.77%**
Avg. Runtime	119ms	6.3s	2.9s

experiments, original implementation of LineMod was used, but only with depth information and without post-processing. Unlike LineMod [128], Drost *et al.*[87] do not necessarily require the refinement. For this reason, neither our poses nor the poses for Drost *et al.*are refined and re-scored. We find this strategy inevitable to reason about our pose results and not the results of ICP. For all these reasons, our results will differ from the original ones, but will be consistent along the experimentation.

The detection rates are shown in Tab. 7.1. For our datasets, only 2-3 segmentation levels per scene were sufficient. In harder cases, one might use more maps to cover for larger variation. It is clearly seen that our method outperforms both methods. Our detection rates never fall below Drost *et al.*[87], as the worst case converges to full matching. Thanks to meaningful segments, using all the points as a single cluster is highly unlikely. On the average, we get 7% more, although this dataset is not cut for our method. It is noteworthy that our improvements in the detection are more significant for small objects (which are hard to spot in clutter and occlusion), while the pose accuracy is more significant in large objects with varying surface characteristics. Nevertheless, we realize increased accuracy in both pose and detection rate regardless of the object size. Also note that, LineMod [128] uses only a hemisphere, whereas we recover the full pose (see Fig. 7.3).

Next, we evaluate the pose accuracy, on the same dataset. To do that, we use the two-metric as explained in § 2.3.6. Our new error function, which is free of the point correspondences but rather depends directly on the pose parameters is defined as:$\mathbf{err}_i = \mathbf{d}_\theta(\mathbf{M}^0_{marker}(\mathbf{M}^i_{marker})^{-1}\mathbf{M}^i_{obj}, \mathbf{M}^0_{obj})$. $\mathbf{d}_\theta$ is a function returning an error vector **err** with angular and translational components. $\mathbf{M}^i_{obj}$ is the object pose at frame

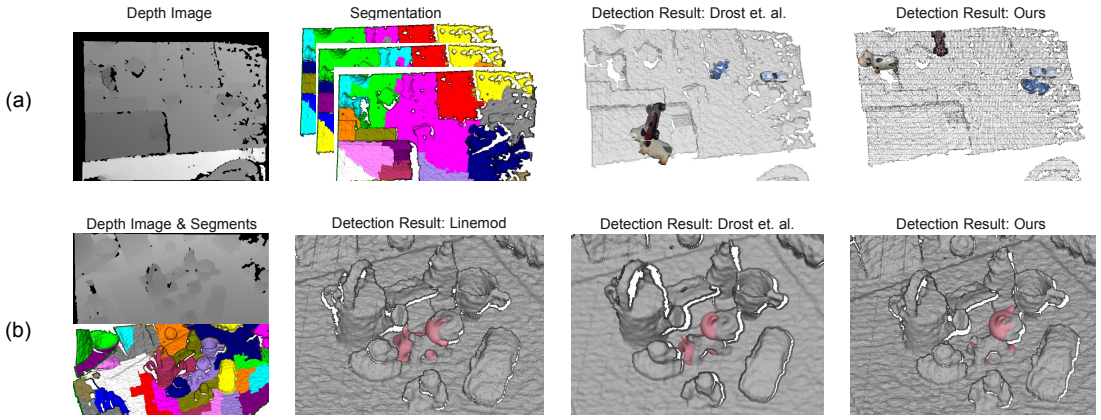


Figure 7.6: Qualitative results. **a)** Detection results in our data, with presence of small objects in long-range Kinect scans. **b)** Pose estimation results on ACCV3D dataset [128]. The accuracy in our poses is even visually distinguishable.

i, where as $\mathbf{M}^{i}_{marker}$ is the pose of the marker board for the same frame. The overall error per object is simply reported as the average pose error in all test frames, i.e average of the set $\{\mathbf{err}_i\}$. This metric transfers each estimated pose to the first frame and computes a pose error between detected object pose transformed to the first frame and ground truth pose in the first frame. We evaluate the error on CAT, DUCK and CAM objects. After transferring each to the initial frame, we perform an ICP and report in Fig. 7.7 $\{\mathbf{err}_i\}$ convergence from our detected pose. After the same number of ICP iterations we have lower rotation/translation error and also because of the better detected pose we need less iterations to converge. Finally, Fig. 7.6 visualizes the results of our method both on a self built setup (Fig. 7.6(a)) and on ACCV3D dataset using CAT object (Fig. 7.6(b)). We show that both the detections and the pose accuracy is visually better than the antecedents.

Complexity analysis and performance One drawback of PPF matching is the combinatorial pairing approach. One way to overcome this problem is by pairing the scene points very sparsely, which is suboptimal. Let M be the number of scene reference points $\mathbf{S_r}$, and N denote all the paired points in the scene. The pairing $(\mathbf{S_r}, \mathbf{S_i})$ then creates a complexity of $O(MN) = O(N_x^2 N_y^2)$, where (N_x, N_y) denotes the dimensions (invisible points and hash-table search are excluded).

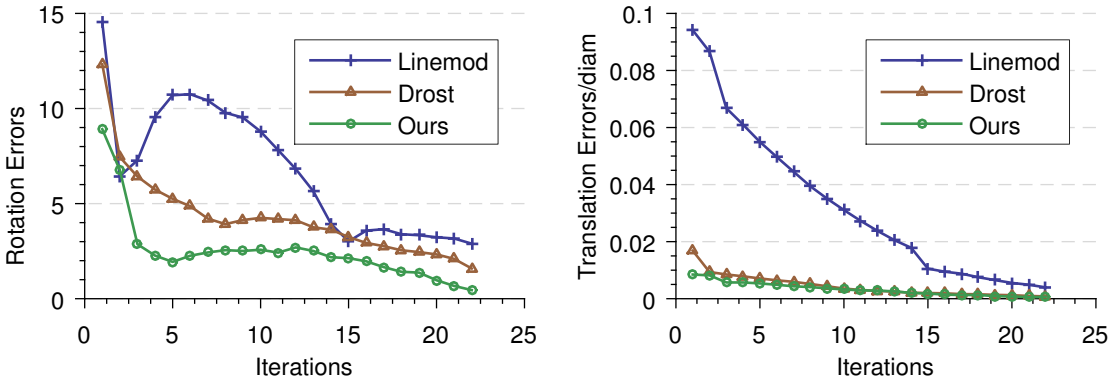


Figure 7.7: Pose errors on ACCV3D dataset as ICP iterates. Rotations are in degrees, while for translation, y-axes are normalized with model diameter.

If we are to segment the image into K clusters, the average number of $\mathbf{S_r}$ per cluster as well as the number of paired points are reduced to $\frac{M}{K}$ and $\frac{N}{K}$, respectively, resulting in an overall complexity of $O(\frac{MN}{K^2})$ per cluster. The overall average time complexity is reduced to $O(\frac{MN}{K})$. If we agree to keep the same complexity, we can now vote for more points. Instead, we prefer to use a set of segmentation with different segment sizes, resulting in more clusters.

We first report the runtime of our algorithm on ACCV3D dataset in Tab. 7.1. Even though we get 2× speed-up over Drost *et al.*[87], this is less than the theoretical possibility. In fact, this is due to the trade-off of obtaining superior detection rate and accuracy by using a sequence of segmentation and more scene points w.r.t. [87].

As explained, the performance is largely affected by the size and the number of the segments. Next experiment targets this effect. We take 3 arbitrary models present in 1000 images, where the objects of interest were *car*, *ape* and *duck*. We sample ~ 900 model points. In each scene, we seek for the minimal number of scene points to pair for a correct match and use that to record the timings. This is in order to make sure that every trial actually results in a correct pose. We plot the segment size vs speed

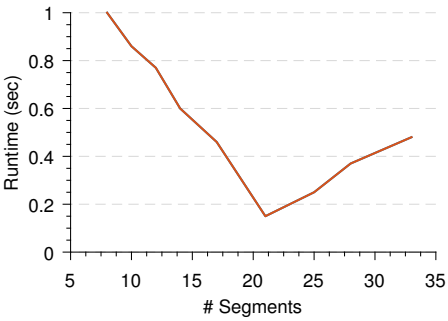
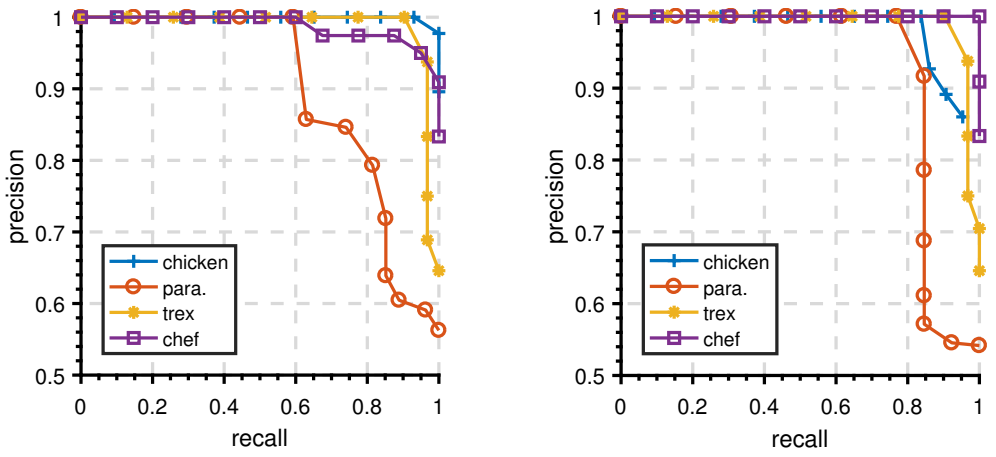


Figure 7.8: Effects of number of segments on runtime.

relation in Fig. 7.8. These timings exclude the data acquisition. Note that there is an optimum point (correct segmentation), which generally depends on the scene. For this experiment, we could reduce the matching time to 170 ms by just using $\frac{1}{50^{th}}$ of the scene points. However, typically, suboptimal choices already allow a descent reduction of computational time, as we do not rely on the precision of the segmentation. This means that, being able to use more clusters, decreases the demand on the sampling and one could use much less scene points to obtain a successful match. Naturally, increasing the segment sizes, reduces the number of clusters and thus the performance converges to that of the original algorithm.

7.3.2 Local Implicit Shape Models (Probabilistic PPF Voting)

Figures 7.9(a) and 7.9(b) provide PR-curves for LISM and the hypothesis verification. Note that although LISM already performs well, our verification clearly improves the distinction between a match vs false positive. Using a simple threshold, we could obtain 100% precision without sacrificing the recall. Thus, our score threshold, combined with the normal consistency check manages to reject all false hypotheses, at the expense of rejecting a small amount of TP.



(a) PR-curve without Verification.

(b) PR-curve with verification.

Figure 7.9: Performance of LISM and verification on Mian dataset.

7.3.3 Limitations

Due to the nature of PPF matching, our approach requires objects with rich geometry. Symmetric objects are also problematic due to ambiguity in pose estimation. Last but not least, currently, there is no mechanism to handle mis-detections. Yet, mis-detections are hardly a problem when the score threshold is reasonably high. This way, we detect in less scenes but avoid mistakes.

7.4 Deeply Learned Features for Object Detection

A natural successor of the geometric approaches presented in the previous sections would be the learned features, that can describe the local 3D structure. With the advent of deep learning, many areas in computer vision shifted from hand crafted labor towards a problem specific end-to-end learning. Local features are of course no exception. Already in 2D, learned descriptors significantly outperform their engineered counterparts [211, 304]. Thus, it was only natural for the scholars to tackle the task of 3D local feature extraction employing similar approaches [150, 308]. However, due to the inherent ambiguities and less informative nature of sole geometry, extracting 3D descriptors on point sets still poses an unsolved problem, even for learning-based methods.

In this part of the thesis, we enrich our object detectors with learned features. We propose one supervised (PPFNet) and one unsupervised (PPF-FoldNet) algorithm to test the capabilities of deep learning in the task of correspondence estimation and object detection. Both PPFNet and PPF-FoldNet outperform the state of the art across the standard benchmarks in which severe rotations are avoided. When arbitrary rotations are introduced into the input, PPF-FoldNet continues to outperform related approaches by a large margin.

7.4.1 PPFNet [81]

We initiate our efforts in learning descriptors by a supervised approach that is robust, yet not fully 6DoF invariant.

PPFNet [81] is a state of the art, deeply learned, fast and discriminative 3D local point cloud descriptor with increased invariance. To satisfy its desirable properties, we

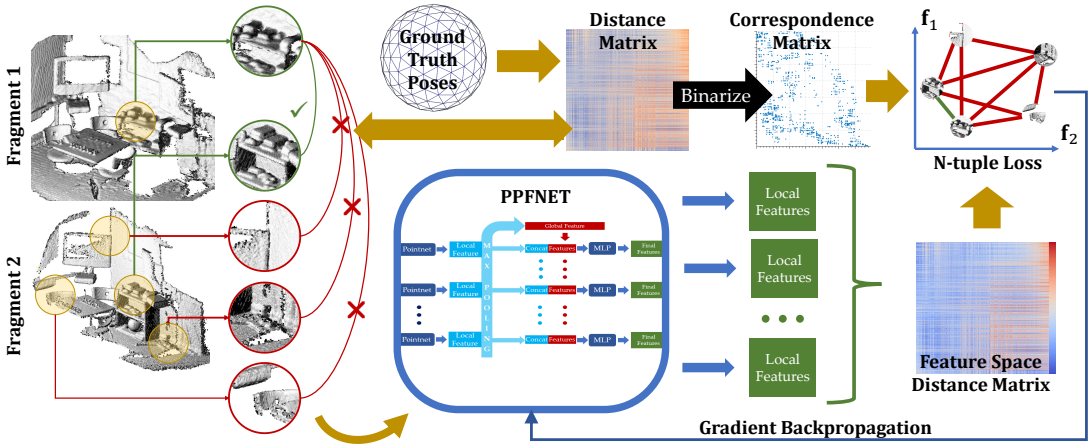


Figure 7.10: PPFNet

first represent the local geometry with an augmented set of simple geometric relationships: points, normals and point pair features (PPF) [87, 232]. We then design a novel loss function, which we term as *N-tuple loss*, to simultaneously embed the matching and non-matching pairs into a Euclidean domain. Our loss resembles the contrastive loss [118], but instead of pairs, we consider an N-combination of the input points within two scene fragments to boost the separability. Thanks to this many-to-many loss function, we are able to also inject the global context into the learning, i.e. PPFNet is aware of the other local features when establishing correspondence for a single one. Also because of such parallel processing, PPFNet is very fast in inference. Finally, we combine all these contributions in a new pipeline, which trains our network from correspondences in 3D fragment pairs. PPFNet extends PointNet [221] and thereby is natural for point clouds and neutral to permutations. The overall architecture is illustrated in Fig. 7.10.

Network architecture The first module of PPFNet is a group of mini-PointNets, extracting features from local patches. Weights and gradients are shared across all PointNets during training. A max pooling layer then aggregates all the local features into a global one, summarizing the distinct local information to the global context of the whole fragment. This global feature is then concatenated to every local feature. A group of MLPs are used to further fuse the global and local features into the final global-context aware local descriptor.

N-tuple loss Our goal is to use PPFNet to extract features for local patches, a process of mapping from a high dimensional non-linear data space into a low dimensional linear feature space. Distinctiveness of the resulting features are closely related to the separability in the embedded space. Ideally, the proximity of neighboring patches in the data space should be preserved in the feature space.

We propose N-tuple loss, an N-to-N contrastive loss, to correctly learn to solve this combinatorial problem by catering for the many-to-many relations. Given the ground truth transformation $\mathbf{T}$, N-tuple loss operates by constructing a correspondence matrix $\mathbf{M} \in \mathbb{R}^{N \times N}$ on the points of the aligned fragments. $\mathbf{M} = (m_{ij})$ where:

$$m_{ij} = \mathbb{1}(\|\mathbf{x}_i - \mathbf{T}\mathbf{y}_j\|_2 < \tau) \tag{7.13}$$

$\mathbb{1}$ is an indicator function. Likewise, we compute a feature-space distance matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$ and $\mathbf{D} = (d_{ij})$ where

$$d_{ij} = \|f(\mathbf{x}_i) - f(\mathbf{y}_j)\|_2 \tag{7.14}$$

The N-tuple loss then functions on the two distance matrices solving the correspondence problem. For simplicity of expression, we define an operation $\sum^*(\cdot)$ to sum up all the elements in a matrix. N-tuple loss can be written as:

$$L = \sum{}^* \left(\frac{\mathbf{M} \circ \mathbf{D}}{\|\mathbf{M}\|_2^2} + \alpha \frac{max(\theta - (1 - \mathbf{M}) \circ \mathbf{D}, 0)}{N^2 - \|\mathbf{M}\|_2^2} \right) \tag{7.15}$$

Here $\circ$ stands for Hadamard Product - element-wise multiplication. α is a hyper-parameter balancing the weight between matching and non-matching pairs and θ is the lower-bound on the expected distance between non-correspondent pairs. We train PPFNet via N-tuple loss, as shown in Fig. 7.10, by drawing random pairs of fragments instead of patches. This also eases the preparation of training data.

7.4.2 Unsupervised Learning with PPF-FoldNet [80]

PPFNet suffers from two unsolved problems: First, full rotation invariance is not achieved, but the network is only tolerant to previously seen rotations. Second, distant patches are defined as non-pairs, an assumption that might not hold for certain local patches lying further away but are still similar. To deal with those, and alleviate the

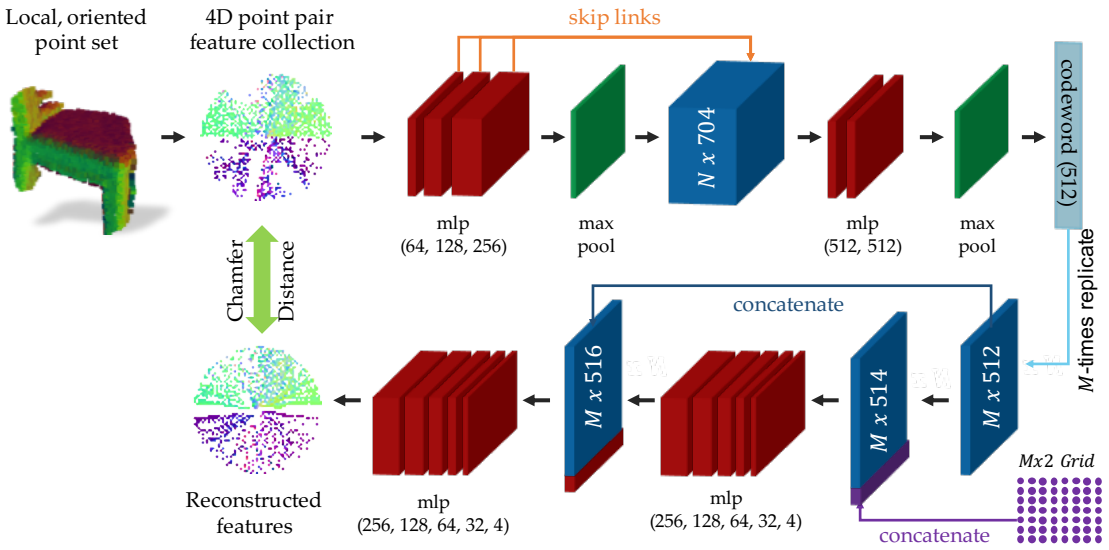


Figure 7.11: PPF-FoldNet

supervision, PPF-FoldNet [80] is proposed, in which, the invariance is achieved by using pure PPF-encoding of the local 3D geometry. As summarized in Fig. 7.11, the collection of the 4D PPFs are sent into a FoldingNet-like end to end auto-encoder (AE) [303], trained to auto-reconstruct the PPFs, using a set distance. This is different to training the network with many possible rotations of the same input and forcing the output to be a canonical reconstruction, which would both be approximate and much harder to learn. Our encoder is simpler than in FoldingNet and for decoding, we propose a similar folding scheme, where a low dimensional 2D grid lattice is folded onto a 4D PPF space. This also allows monitoring the network evolution thanks to our lossless visualization of the PPF space. The latent low dimensional vector of the auto-encoder, *codeword*, is used as the local descriptor attributed to the point around which the patch is extracted. Training of PPF-FoldNet is far easier than training, for example, 3DMatch [308], because it eliminates the necessity of pair information and benefits from linear time complexity in the number of patches.

Encoder The input to our network, and thus to the encoder, is $\mathbf{F}_\Omega$, a local PPF representation, as in §2.7. A three-layer, point-wise MLP (Multi Layer Perceptron) follows the input layer and subsequently a max-pooling is performed to aggregate the individual features into a global one, similar to PointNet. The low level features

are then concatenated with this global feature using skip-links. This results in a more powerful representation. Another two-layer MLP finally redirects these features to a final encoding, the codeword, which is of dimension 512.

Decoder Our decoder tries to reconstruct the whole set of point PPFs using a single codeword, which in return, also forces the codeword to be informative and distill the most distinctive information from the high-dimensional input space. However, inspired by FoldingNet, instead of trying to upsample or interpolate point sets, the decoder will try to deform a low-dimensional grid structure guided by the codeword. Each grid point is concatenated to a replica of the codeword, resulting in an $M \times 514$ vector as input to what is referred as *folding operation* [303]. Folding can be a highly non-linear operation and is thus performed by two consecutive MLPs: the first folding results in a deformed grid, which is appended once again to the codewords and propagates through the second MLP, reconstructing the input PPFs. Moreover, in contrast to FoldingNet, we try to reconstruct a higher dimensional set, 4D vs 3D (2D manifold); we are better off using a deeper MLP - 5-layer as opposed to the 3-layer.

Chamfer loss Note that as size of the grid M, is not necessarily the same as the size of the input N, and the correspondences in 4D PPF space are lost when it comes to evaluating the loss. This requires a distance computation between two unequal cardinality point pair feature sets, which we measure via the well known Chamfer metric:

$$d(\mathbf{F}, \hat{\mathbf{F}}) = \max\left\{ \frac{1}{|\mathbf{F}|} \sum_{\mathbf{f} \in \mathbf{F}} \min_{\hat{\mathbf{f}} \in \hat{\mathbf{F}}} \|\mathbf{f} - \hat{\mathbf{f}}\|_2, \ \frac{1}{|\hat{\mathbf{F}}|} \sum_{\hat{\mathbf{f}} \in \hat{\mathbf{F}}} \min_{\mathbf{f} \in \mathbf{F}} \|\mathbf{f} - \hat{\mathbf{f}}\|_2 \right\} \tag{7.16}$$

where $\hat{}$ operator refers to the reconstructed (estimated) set.

7.4.3 Results and Evaluation

We compare our methods against 3 hand-crafted features (Spin Image [140], SHOT [235], FPFH [232]) and 2 deep-learned features (3DMatch [308], CGF [150]) on 3DMatch Benchmark, identical as the one used in [308]. 3DMatch uses a TSDF input representation and CGF benefits from a hand-crafted histogram and only uses learning

Table 7.2: Our results on the standard 3DMatch benchmark. *Red Kitchen* data is from 7-scenes [248] and the rest imported from SUN3D [296].

	Spin Image	SHOT	FPFH	3DMatch	CGF	PPFNet	PPF-FoldNet
Kitchen	0.1937	0.1779	0.3063	0.5751	0.4605	**0.8972**	0.7352
Home 1	0.3974	0.3718	0.5833	0.7372	0.6154	0.5577	**0.7564**
Home 2	0.3654	0.3365	0.4663	**0.7067**	0.5625	0.5913	0.625
Hotel 1	0.1814	0.208	0.2611	0.5708	0.4469	0.5796	**0.6593**
Hotel 2	0.2019	0.2212	0.3269	0.4423	0.3846	0.5769	**0.6058**
Hotel 3	0.3148	0.3889	0.5000	0.6296	0.5926	0.6111	**0.8889**
Study	0.0548	0.0719	0.1541	0.5616	0.4075	0.5342	**0.5753**
MIT Lab	0.1039	0.1299	0.2727	0.5455	0.3506	**0.6364**	0.5974
Average	0.2267	0.2382	0.3589	0.5961	0.4776	0.6231	**0.6804**

to reduce the dimensionality. The metric R is used to evaluate the qualities of the returned matching results by using different features, the higher the better:

$$R = \frac{1}{M}\sum_{s=1}^{M} \mathbb{1}\left(\left[\frac{1}{|\Omega|}\sum_{(i,j\in\Omega)} \mathbb{1}\left((\mathbf{x}_i - \mathbf{T}\mathbf{y}_j) < \tau_1\right)\right] > \tau_2\right) \tag{7.17}$$

M is the number of ground truth matching fragment pairs, having at least 30% overlap with each other under ground-truth transformation $\mathbf{T}$ and $\tau_1 = 10cm$. (i, j) denotes an element of the found correspondence set Ω. $\mathbf{x}$ and $\mathbf{y}$ respectively come from the first and second fragment under matching. The inlier ratio is set as $\tau_2 = 0.05$.

Table 7.3: Our results on the rotated 3DMatch benchmark. *Red Kitchen* data is from 7-scenes [248] and the rest imported from SUN3D [296].

	Spin Image	SHOT	FPFH	3DMatch	CGF	PPFNet	PPF-FoldNet
Kitchen	0.1779	0.1779	0.2905	0.004	0.4466	0.002	**0.7352**
Home 1	0.4487	0.3526	0.5897	0.0128	0.6667	0.0000	**0.7692**
Home 2	0.3413	0.3365	0.4712	0.0337	0.5288	0.0144	**0.6202**
Hotel 1	0.1814	0.2168	0.3009	0.0044	0.4425	0.0044	**0.6637**
Hotel 2	0.1731	0.2404	0.2981	0.0000	0.4423	0.0000	**0.6058**
Hotel 3	0.3148	0.3333	0.5185	0.0096	0.6296	0.0000	**0.9259**
Study	0.0582	0.0822	0.1575	0.0000	0.4178	0.0000	**0.5616**
MIT Lab	0.1169	0.1299	0.2857	0.026	0.4156	0.0000	**0.6104**
Average	0.2265	0.2337	0.364	0.0113	0.4987	0.0026	**0.6865**

Tab. 7.2 records the matching results of different methods. Both PPFNet and PPF-FoldNet outperforms all the other counterparts in average recall, achieving 62.31% and 68.04% respectively. Superior results attained by our methods validate the strength of our features learned from sparse point-wise input against to volumetric or hand crafted representations. Noteworthy is that, even without supervision signals and using only PPFs as input, PPF-FoldNet obtains a higher recall than PPFNet, as it is free from those wrong pair/non-pair labels which could mislead PPFNet.

To further illustrate the power of PPF-FoldNet under severe rigid transformations, we shuffled the standard 3DMatch benchmark with random rotations and created a rotated benchmark, re-applied all the methods on this rotated benchmark. Results are collected in Tab. 7.3. Due to the coordinates and normals existing in PPFNet, the its performance is harmed drastically. Same degradation could also be seen on 3DMatch. On the contrary, CGF and PPF-FoldNet are not impacted, but our PPF-FoldNet achieves much better recalls against CGF on all the scene sequences.

Sparsity evaluation Thanks to the sparse representation of our input, PPFNet and PPF-FoldNet are also robust in respect of the changes in point cloud density. Fig.7.12 shows the performance of different methods when we gradually decrease the number of points in the fragment from 100% to only 6.25%. PPF-FoldNet is least affected by the reduction in density, while PPFNet is reasonably robust. In particular, when only 6.25% points are left in the fragments, the recall for PPF-FoldNet is still greater than 50% while the other methods almost fail. The results of PPFNet and PPF-FoldNet together demonstrate that PPF representation offers more robustness to input sparsity, which is a common problem for many other encoding.

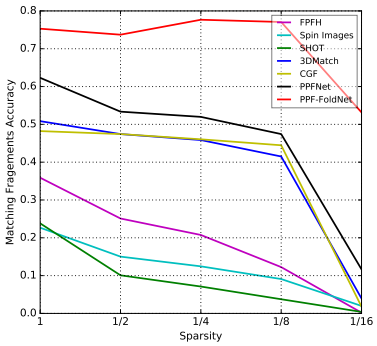


Figure 7.12: Evaluating robustness again point density.

Runtime We run our algorithms on a machine loaded with NVIDIA TitanX Pascal GPU and an Intel Core i7 3.2GHz CPU. We report the runtime of different methods in Tab. 7.4, where input preparation (on CPU) is assessed separately from the total run-

time. Clearly, PPF-FoldNet is also the method of choice regarding the computational costs, while PPFNet is reasonably close in terms of speed.

Table 7.4: Runtime comparisons (reported in seconds) for 2048 local patches.

FPFH	3DMatch			PPFNet			PPF-FoldNet		
Total	Prep.	Inference	Total	Prep.	Inference	Total	Prep.	Inference	Total
31.678	0.634	5.939	6.574	4.587	0.112	4.608	2.616	1.353	**3.969**

7.5 Relaxing the Rigidity and A Bottom-Up Approach: Quadrics

Figure 7.13: *Maurtis Cornelis Escher, Heaven & Hell*

The disc is divided into six sections in which, turn and turn about, the angels on a black background and then the devils on a white one, gain the upper hand. In this way, heaven and hell change place six times. In the intermediate, “earthly” stages, they are equivalent.

Benedict Taschen, "The Graphic Work by M C Escher (introduced and explained by the artist)", 1990

The aforementioned approaches that try to solve six-DoF pose estimation problem are quite successful. However, as the only parameters to discover are rotations and translations, they require a huge number of CAD models to generically represent the real scenarios. To overcome this limitation, inspired by the fact that all CAD models are designed using a similar set of tools, a different line of research attempts to find common bases explaining a broad set of 3D objects, and tries to detect these bases instead of individual models. Such bases that are the common building blocks of our world, and typically termed *geometric primitives*. While the approaches using bases, significantly reduce the database size, usually, the bases undergo higher dimensional transformations compared to, for instance, rigid ones. Examples of the geometric primitives are splines and nurbs surfaces defined by several control points, or *quadrics*,

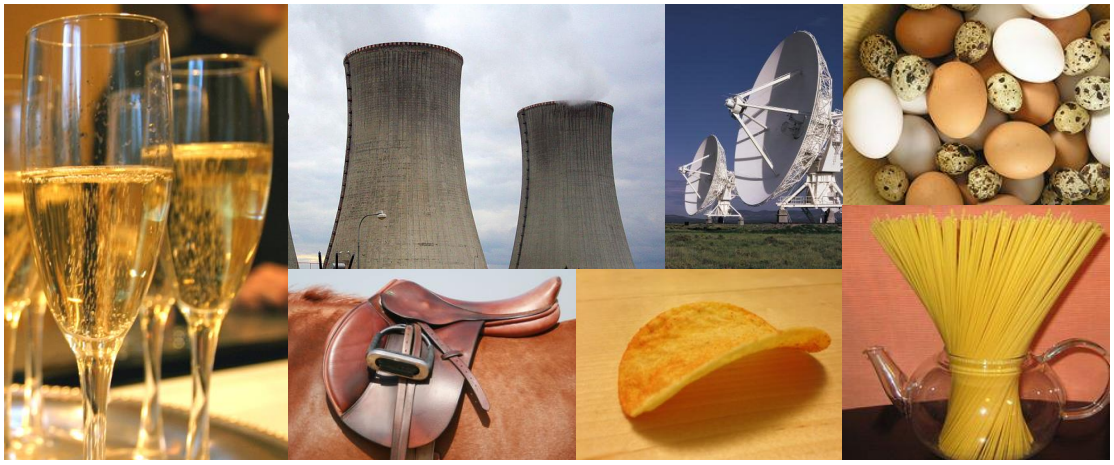


Figure 7.14: Quadric surfaces in real life. *A collection retrieved from Google images.*

the three dimensional, nine-DoF, quadratic forms.

Thanks to their power to embody the most typical geometric primitives, such as planes, spheres, cylinders, cones or ellipsoids, quadrics themselves were of huge interest to vision community since 80s [195]. In fact, Arthur Cayley acknowledges quadrics to be *absolute* and adds: *"There are three absolutes in the real projective line, seven in the real projective plane, and 18 in real projective space. All classical non-euclidean projective spaces as hyperbolic, elliptic, Galilean and Minkowskian and their duals can be defined this way."* Fig. 7.14 portrays the common quadrics present in our every day lives. Some exemplary studies involve recovering 3D quadrics from image projections [71], fitting them to 3D point sets [263], or detecting special cases of the quadratic forms [11]. A majority of those works either put emphasis on fitting to a noisy, but isolated point set [39, 260, 263], or restrict the types of shapes under consideration (thereby reduce the DoF) to devise detectors robust to clutter and occlusions [8, 10, 105].

In this part, our aim is to unite the fitting and detection worlds and present an algorithm that can simultaneously estimate all parameters of a generic nine DoF quadric, which, as shown in Fig. 7.15, resides in a 3D cluttered environment and is viewed potentially from a single 3D sensor, introducing occlusions and partial visibility. We craft this algorithm in three stages: (1) First, we devise a new quadric fit. Unlike its ancestors, this one uses the extra information about the tangent space to increase the number of constraints instead of regularizing the solution. This fit requires only

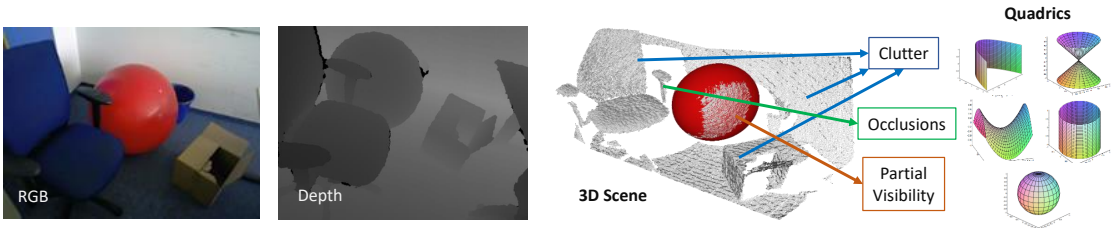


Figure 7.15: Our algorithm quickly detects quadric shapes in point clouds under occlusions, partial visibility and clutter. Note that, the algorithm has no clue whether a sphere exists or not and is able to detect the plausible primitive using only 3D geometry.

four oriented points. We show that such construction also has a regularization effect as a by-product. (2) We then thoroughly analyze its rank properties and devise a novel null-space Hough voting mechanism to reduce the four point case to three. Three points stands out to be the minimalist case developed so far. (3) We propose a variant of RANSAC that operates on our local bases, which are randomly posited. Per each local basis, we show how to make use of the fit and the voting to hypothesize a likely quadric. Finally, we use simple clustering heuristics to group and strengthen the candidate solutions. Our algorithm works purely on 3D point cloud data and does not depend upon any acquisition modality. Moreover, it makes assumptions neither about the type of the quadric that is present in the scene nor how many are visible. Our experimental evaluation demonstrates that we could satisfy all these premises and indeed in certain applications, type-specific detectors can lead to different types, therefore wrong parameters. In those cases resorting to the quadric becomes advantages. Moreover, it is trivial to convert our algorithm to a type specific detector. We show that in those cases, for example for a sphere-fit, we can achieve very robust results with real-time performance. This is thanks to the reduction in basis cardinality.

Our presentation here involves our CVPR publication [32], as well as the following additions:

1. qualitative and quantitative experiments to better grasp the behavior of the proposed fit,
2. algebraic and geometric theoretical analysis of the quadric fit,
3. extensive descriptions of the method as well as accompanying pseudocode.

7.5.1 Related Work

The majority of works related to quadrics can be classified into Primitive Detection, Quadric Fitting and Quadric Detection, where we only review the seminal ones.

Quadrics Quadrics appear in various domains of vision, graphics and robotics. They are found to be one of the best local surface approximators in estimating differential properties [217]. Thus, point cloud normals and curvatures are oftentimes estimated with local quadrics [35, 312]. Yan *et al.*propose an iterative method for mesh segmentation by fitting local quadratic surfaces [299]. Yu presented a quadric-region based method for consistent point cloud segmentation [173]. Kukelova uses quadric intersections to solve minimal problems in computer vision [156]. Uto *et al.* [282] as well as Pas and Platt [216] use quadrics to localize grasp poses and in grasp planning. Quadrics have also been a significant center of attention in projective geometry and reconstruction [71, 102] to estimate algebraic properties of apparent contours. Finally, You and Zhang [306] used them in feature extraction from face data.

Primitive detection Finding primitives in point clouds has kept the vision researchers busy for a lengthy period of time. Works belonging to this category treat the primitive shapes independently [105], giving rise to specific fitting algorithms for planes, spheres, cones, cylinders, etc. Planes, as the simplest forms, are the primary targets of the Hough-family [41]. Yet, detection of more general set of primitives made RANSAC the method of choice as shown by the prosperous Globfit [166]: a relational local to global RANSAC algorithm. Schnabel *et al.* [240] and Tran *et al.* [271] also focused on reliable estimation using RANSAC. An interesting application of primitives is given by Qui *et al.*who extract pipe runs using cylinder fitting [224]. The local Hough transform of Drost and Ilic [85] showed how the detection of primitives can be made efficient by considering the local voting spaces. Authors give sphere, cylinder and plane specific formulations. Lopez *et al.* [175] devise a robust ellipsoid fitting based on iterative nonlinear optimization. Sveier *et al.* [256] suggest a conformal geometric algebra to spot planes, cylinders and spheres. Andrews' approach [11] deals with paraboloids and hyperboloids in CAD models. Even though this is slightly more generalized, paraboloids or hyperboloids are not the only geometric shapes described by quadrics.

Methods in this category are successful in shape detection, yet they handle the primitives separately. This prevents automatic type detection, or generalized modeling.

Quadric fitting Since the 1990s generic quadric fitting is cast as a constrained optimization problem, where the solution is obtained from a Generalized Eigenvalue decomposition of a scatter matrix. Pioneering work has been done by Gabriel Taubin [263] in which a Taylor approximation to the geometric distance is made. This work has then been enhanced by 3L [39], fitting a local, explicit ribbon surface composed of three-level-sets of constant Euclidean distance to the object boundary. This fit implicitly used the local surface information. Later, Tasdizen [260] improved the local surface properties by incorporating the surface normals as regularizers. This allows for a good and stable fit. Recently, Beale *et al.* [22] introduced the use of a Bayesian prior to regularize the fit. All of these methods use at least nine or twelve [22] points. Moreover, they only use surface normals as regularizers - not as additional constraints and are also unable to deal with outliers in data. There are a few other studies [8, 143], improving these standard methods, but they involve either non-linear optimization [300] or share the common drawback of requiring nine independent constraints and no outlier treatment.

Quadric detection Recovering general quadratic forms from cluttered and occluded scenes is a rather unexplored area. A promising direction was to represent quadrics with spline surfaces [200], but such approaches must tackle the increased number of control points, i.e. 8 for spheres, 12 for general quadrics [222, 223]. Segmentation is one way to overcome such difficulties [183, 200]. In fact, Vaskevicius *et al.* [284] developed a noise-model aware quadric fitting and region-growing algorithm for segmented noisy scenes. However, segmentation, due to its nature, decouples the detection problem in two parts and introduces undesired errors especially under occlusions. Other works exploit genetic algorithms [110] but have the obvious drawback of inefficiency. QDEGSAC [98] proposed a six-point hierarchical RANSAC, but the paper misses out an evaluation or method description for a quadric fit. Petitjean [217] stressed the necessity of outlier aware quadric fitting however only ends up suggesting M-estimators for future research.

	# Pri.	# Dual
PD-0	9	0
PD-1	7	1
PD-2	5	2
PD-3	4	3

(a)

	# Pri.	# Dual	VS
Plane	1	1	0
2-Planes	2	2	0
Sphere	2	2	1
Spheroid	2	2	3

(b)

Figure 7.16: **(a)** Number of constraints for a minimal fit in Primal(P) or Dual(D) spaces. PD-i refers to i$^{\text{th}}$ combination. **(b)** Number of minimal constraints and voting space size for various quadrics.

7.5.2 A New Perspective to Quadric Fitting to 3D Data

State of the art direct solvers for quadric fitting rely either solely on point sets [263], or use surface normals as regularizers [260]. Both approaches require at least nine points, posing a strict requirement for practical considerations, i.e. using nine points bounds the possibility for RANSAC-like fitting algorithms as the space of potential samples is N_x^9 where N_x is the number of points. Here, we observe that typical real life point clouds make it easy to compute the surface normals (tangent space) and thus provide an additional cue. With this orientation information, we will now explain a closed form fitting requiring only four oriented points. Similar to gradient-one fitting [260, 261], our idea is to align the gradient vector of the quadric $\nabla\mathbf{Q}(\mathbf{x}_i)$ with the normal of the point cloud $\mathbf{n}_i \in \mathbb{R}^3$. However, unlike $\nabla 1$ [261], we opt to use a linear constraint to increase the rank rather than regularizing the solution. This is seemingly non-trivial as the vector-vector alignment brings a non-linear constraint either of the form:

$$\frac{\nabla\mathbf{Q}(\mathbf{x}_i)}{\|\nabla\mathbf{Q}(\mathbf{x}_i)\|} - \mathbf{n}_i = 0 \quad \text{or} \quad \frac{\nabla\mathbf{Q}(\mathbf{x}_i)}{\|\nabla\mathbf{Q}(\mathbf{x}_i)\|} \cdot \mathbf{n}_i = 1. \tag{7.18}$$

The non-linearity is caused by the normalization as it is hard to know the magnitude and thus the homogeneous scale in advance. We solve this issue by introducing a per normal homogeneous scale α_i among the unknowns and write:

$$\nabla\mathbf{Q}(\mathbf{x}_i) = \nabla\mathbf{v}_i^T\mathbf{q} = \alpha_i\mathbf{n}_i \tag{7.19}$$

Algorithm 2: Quadric fitting, full version.

input : Unit normalized point set $\{\mathbf{x},\mathbf{y},\mathbf{z}\}$, Corresponding surface normals $\{\mathbf{n}_x,\mathbf{n}_y,\mathbf{n}_z\}$, A weight coefficient ω
output : Quadric $\mathbf{q} = [A,B,C,2D,2E,2F,2G,2H,2I,J]^T$, Scale factors $\boldsymbol{\alpha}$
$n = \text{numel}(\mathbf{x})$
$\mathbf{1} = \text{ones}(n,1)$;
$\mathbf{0} = \text{zeros}(n,1)$;
$\mathbf{0}_{nxn} = \text{zeros}(n,n)$;
$\mathbf{X} = [\mathbf{x}^2,\mathbf{y}^2,\mathbf{z}^2,\mathbf{x}*\mathbf{y},\mathbf{x}*\mathbf{z},\mathbf{y}*\mathbf{z},\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{1},\mathbf{0}_{nxn}]$;
$\mathbf{N} = [\text{diag}(\mathbf{n}_x);\text{diag}(\mathbf{n}_y);\text{diag}(\mathbf{n}_z)]$;
$\mathbf{dX} = [2\mathbf{x},\mathbf{0},\mathbf{0},\mathbf{y},\mathbf{z},\mathbf{0},\mathbf{1},\mathbf{0},\mathbf{0},\mathbf{0};\ldots$
$\quad \mathbf{0},2\mathbf{y},\mathbf{0},\mathbf{x},\mathbf{0},\mathbf{z},\mathbf{0},\mathbf{1},\mathbf{0},\mathbf{0};\ldots$
$\quad \mathbf{0},\mathbf{0},2\mathbf{z},\mathbf{0},\mathbf{x},\mathbf{y},\mathbf{0},\mathbf{0},\mathbf{1},\mathbf{0}]$;
$\mathbf{A} = [\mathbf{X};\omega\cdot[\mathbf{dX},-\mathbf{N}]]$;
$[\sim,\sim,\mathbf{V}] = \text{svd}(\mathbf{A})$;
$\mathbf{q} = \mathbf{V}(1:10,n+10)$;
$\boldsymbol{\alpha} = \mathbf{V}(11:(n+10),n+10)$;

Stacking this up for all N points $\mathbf{x}_i$ and normals $\mathbf{n}_i$ leads to:

$$\mathbf{A}'\begin{Bmatrix}\mathbf{M}\\ \mathbf{N}\end{Bmatrix}\begin{bmatrix}\mathbf{v}_1^T & 0 & 0 & \cdots & 0\\ \mathbf{v}_2^T & 0 & 0 & \cdots & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ \mathbf{v}_n^T & 0 & 0 & \cdots & 0\\ \nabla\mathbf{v}_1^T & -\mathbf{n}_1 & \mathbf{0}_3 & \cdots & \mathbf{0}_3\\ \nabla\mathbf{v}_2^T & \mathbf{0}_3 & -\mathbf{n}_2 & \cdots & \mathbf{0}_3\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ \nabla\mathbf{v}_n^T & \mathbf{0}_3 & \mathbf{0}_3 & \cdots & -\mathbf{n}_n\end{bmatrix}\begin{bmatrix}A\\ B\\ \vdots\\ I\\ J\\ \alpha_1\\ \alpha_2\\ \vdots\\ \alpha_n\end{bmatrix} = \mathbf{0} \tag{7.20}$$

where $\nabla\mathbf{v}_i^T = \nabla\mathbf{v}(\mathbf{x}_i)^T$, $\mathbf{0}_3$ is a 3×1 column vector of zeros, $\mathbf{A}'$ is $4N\times(N+10)$ and $\boldsymbol{\alpha} = \{\alpha_i\}$ are the unknown homogeneous scales. The solution containing quadric coefficients and individual scale factors lies in the null-space of $\mathbf{A}'$, and can be obtained accurately via Singular Value Decomposition. Alg. 2 provides a MATLAB implementation of such fit. For a non-degenerate quadric, the following rank (rk) relations hold:

$$N = 1 \Rightarrow \text{rk}(\mathbf{A}) = 4,\ N = 2 \Rightarrow \text{rk}(\mathbf{A}) = 7$$
$$N = 3 \Rightarrow \text{rk}(\mathbf{A}) = 9, \text{ and } N > 3 \Rightarrow \text{rk}(\mathbf{A}) = 10. \tag{7.21}$$

We will now further investigate on this interesting behavior.

Existence of a trivial solution for three-point case The problem of estimating a quadric from three points and associated normals seems initially to be well-posed: when counting constraints and degrees of freedom, one obtains nine on each side (each point gives one constraint, each normal two, whereas a quadric has nine degrees of freedom). Yet, it turns out that our linear equation system always has a trivial solution besides the true one. This is summarized in Eq. 15 by providing the ranks for different cardinalities of bases. We now give further intuition and proof for this behavior:

Theorem 7.1. *Three-oriented point quadric fitting, as formulated, possesses a trivial solution besides the true solution.*

Proof. In the following, let us call **data-plane**, the plane spanned by the three data points (coordinates only, i.e. not considering the associated normals). We illustrate this in Fig. 7.17 (left). As mentioned in § 7.5.2 above, any rank-1 quadric consists of a single plane Π and can be written as $\mathbf{Q} = \Pi\,\Pi^T$. Hence, for any point $\mathbf{U}$ on the plane and thus on the quadric, we have $\mathbf{QU} = \mathbf{0}$. In our formulation of the fitting problem, this amounts to $\mathbf{Nq} = \mathbf{0}$. $\mathbf{N}$ refers to all the gradient-normal correspondence equations, stacked together (lower part of Eq. 7.20). We also have $\mathbf{Mq} = \mathbf{0}$. due to the point lying on the quadric. This means that the following vector is a solution of the equation system: coefficients A to J are those of the rank-1 quadric and the three scalars α_i are zero. In other words, the trivial solution is identified as the rank-1 quadric consisting of the data-plane. Hence, the estimation problem admits at least a one-dimensional linear family of solutions, spanned by the true quadric and the rank-1 quadric of the data-plane. In some cases, the dimension of the family of solutions may be higher (such as when the true quadric is a plane). ■

A geometric explanation of the fact that the three-oriented-point problem is always under-constrained Despite the analytical proof, it is puzzling that nine constraints on nine unknowns are never sufficient in our problem. Moreover, we may wonder if the existence of a trivial solution is due to our linear problem formulation or if it is generic. It turns out that this is generic and can be explained geometrically.

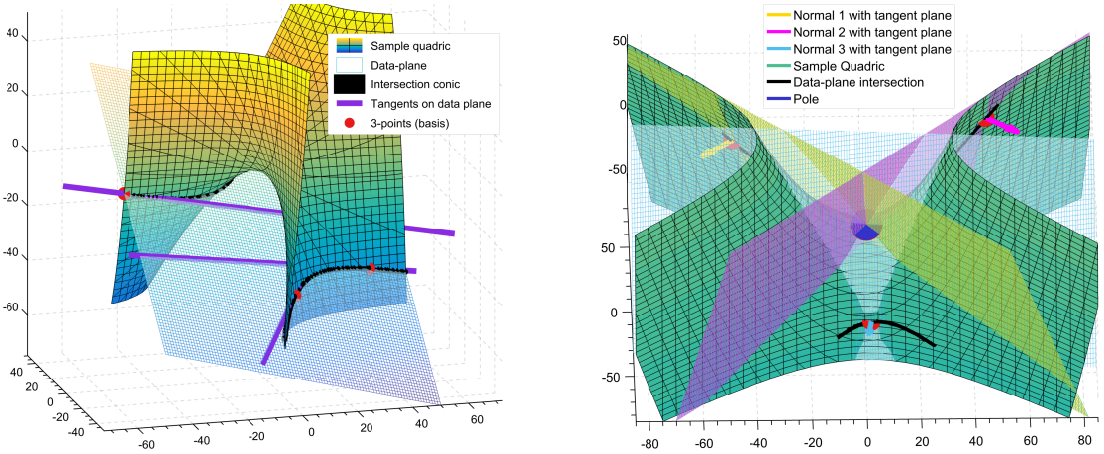


Figure 7.17: Illustration of the geometric intuition. **(left)** Visualizations on a sample quadric: the selected basis (three-oriented points); the data-plane; the conic of intersection between data-plane and the quadric; the lines on the data-plane that are tangent to the quadric. **(right)** Exemplary drawing that shows that the tangent planes to the basis points meet at the pole (see text).

To make it easier to imagine, our description will closely follow the Figures 7.17(a) and 7.17(b). Let us decompose the estimation of the quadric in two parts, the first part being the determination of the quadric's intersection with the data-plane. The intersection of any quadric with any plane is in general always a conic (shown as black curve), be it real or imaginary (the only exception is when the quadric itself contains the data-plane entirely, in which case the "intersection" is the entire plane).

Let us examine which constraints we have at our disposal to estimate the intersection conic. First, the three data points lie on the conic. Second, we know the tangent planes at the data points, to the true quadric. Let us intersect the three tangent planes with the data plane – the resulting three lines (shown in purple) must be tangent to our conic (the only exception occurs when one or more of these tangent planes are identical to the data plane).

Hence, we know three points on the conic and three tangent lines – the problem of estimating the conic is thus in general overdetermined by one DoF. In other words, six of the nine constraints at our disposal are dedicated to estimating the five degrees of freedom of its intersection with the data plane. Hence, the remaining three constraints are not sufficient to complete the estimation of the quadric.

What are these three remaining constraints? They refer to the orientation of the tangent planes: for each tangent plane, an angle expressing the rotation about its intersection line with the data plane. This angle gives one piece of information on the quadric; for three oriented points we thus have our three remaining constraints.

Note that the three tangent planes to the quadric intersect in the quadric's pole to the data plane (see Fig. 7.17(b)). Hence, we can determine this pole which, as shown in appendix, lies on the line joining the centers of the possible solutions for the quadric.

Let us also note that the fact that six pieces of information (three data points and three tangent lines to a conic in the data plane) only constrain five degrees of freedom means that these six pieces of information are not independent from one another: in the absence of noise or other errors, they must satisfy a consistency constraint (the fact that they define a conic). In the presence of noise, the input information will not satisfy this constraint, meaning that a perfect fit will not exist. This is different in most so-called minimal estimation problems in geometric computer vision (such as three-point pose estimation - P3P), where the computed solution is perfectly consistent with the input data. In our case, we can expect that the computed quadric will not satisfy all constraints exactly, i.e. will not necessarily be incident with all data points or be exactly tangent to the given tangent planes.

This gives room to different formulations for the problem, depending on how one quantifies the quality of fit. For instance, one possibility would be to impose that the quadric goes exactly through the data points, but that the tangency is only approximately fulfilled by computing the intersection conic in the data plane and minimizing some cost functions over the tangent lines.

Collinearity of solution centers In the following we show that the centers of the quadrics that are solutions to the 3-oriented-points fitting problem (when the center exists), lie on a straight line, spanned by the center of the true quadric, and the pole of the data-plane to the true quadric. Suppose that the true quadric $\mathbf{Q}$ is central, that is : $\mathbf{Qc} = r\mathbf{e}^{\infty}$ for some scalar $r \in \mathbb{R}$ and $\mathbf{e}^{\infty} \triangleq \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$. $\mathbf{c} \in \mathbb{RP}^3$ is the center. Let Π be the data-plane. As seen, the trivial solution to our problem is the rank-1 quadric given by $\Pi\,\Pi^T$. Let $\mathbf{P}$ be the pole of the plane Π with respect to the true quadric $\mathbf{Q}$. Without loss of generality, assume that the homogeneous coordinates of $\mathbf{P}$ are scaled such that the following pole-polar relation holds exactly and not only up to scale:$\mathbf{QP} = \Pi$. We

now formulate the theorem:

Theorem 7.2. *Consider the family of quadrics which are solutions to our problem, i.e. the quadrics* $\mathbf{Q}'(\lambda) = \mathbf{Q} + \lambda\Pi\,\Pi^T$ *(for a free parameter* λ*). The centers of these quadrics lie on the line spanned by the center* $\mathbf{c}$ *of the true quadric* $\mathbf{Q}$*, and the pole* $\mathbf{P}$ *of plane* Π *with respect to* $\mathbf{Q}$*.*

Proof. The theorem is true if, for every λ, there exists a μ such that $\mathbf{c}+\mu\mathbf{P}$ is the center of $\mathbf{Q}'(\lambda)$. More formally, if:

$$\mathbf{Q}'(\lambda)\left(\mathbf{c}+\mu\mathbf{P}\right) \sim \begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^T \quad .$$

Let us develop the left-hand side:

$$\begin{aligned}\mathbf{Q}'(\lambda)\left(\mathbf{c}+\mu\mathbf{P}\right) &= \mathbf{Q}\mathbf{c} + \mu\mathbf{Q}\mathbf{P} + \lambda\Pi\,\Pi^T\mathbf{c} + \lambda\mu\Pi\,\Pi^T\mathbf{P} \\ &= r\mathbf{e}^\infty + \mu\Pi + \Pi\left(\lambda\Pi^T\mathbf{c}\right) + \Pi\left(\lambda\mu\Pi^T\mathbf{P}\right) \\ &= r\mathbf{e}^\infty + \Pi\left(\mu + \lambda\Pi^T\mathbf{c} + \lambda\mu\Pi^T\mathbf{P}\right) \quad .\end{aligned} \tag{7.22}$$

This is equal (up to scale) to $\mathbf{e}^\infty \triangleq [0 \quad 0 \quad 0 \quad 1]^T$ exactly if $\Pi \sim \begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^T$ or if $\mu + \lambda\Pi^T\mathbf{c} + \lambda\mu\Pi^T\mathbf{P} = 0$. The former means that the plane spanned by the three data points, is the plane at infinity, i.e. all three data points are at infinity. We safely exclude this case in our application. As for the second case, it can be achieved by setting

$$\mu = -\frac{\lambda\Pi^T\mathbf{c}}{1+\lambda\Pi^T\mathbf{P}} \quad . \tag{7.23}$$

Hence, the point $\mathbf{c}+\mu\mathbf{P}$ is indeed the center of $\mathbf{Q}'(\lambda)$. By construction, it is collinear with $\mathbf{c}$ and $\mathbf{P}$, which proves the theorem. A special case occurs when the denominator in the definition of μ is zero, i.e. where $1+\lambda\Pi^T\mathbf{P} = 0$, in which case μ "equals" infinity. This effectively means that $\mathbf{P}$ is the center of $\mathbf{Q}'(\lambda)$, hence the theorem still holds. ■

A special case is when $\Pi^T\mathbf{c} = 0$, i.e. the data-plane contains the center $\mathbf{c}$ of the true quadric $\mathbf{Q}$. Then, μ is always zero, which means that all $\mathbf{Q}'(\lambda)$ have the same center $\mathbf{c}$.

.

Regularizing with gradient norm Quadric fitting problem, like many others (e.g. calibration, projective reconstruction) is intrinsically of non-linear nature, meaning that a "true" MLE or MAP solution, cannot be achieved by a linear fit.[1] However, our main objective in this stage is a sufficiently close and computationally efficient fit, using as few points as possible and upon which we can build our voting scheme. Despite its sparsity, for such purpose, formulation in § 7.5.2 still remains suboptimal since the unknowns in Eq. 7.20 scale linearly with N, leaving a large system to solve. In practice, analogous to gradient-one fitting [261], we could prefer unit-norm polynomial gradients, and thus, can write $\alpha_i = 1$ or equivalently $\alpha_i \leftarrow \tilde{\alpha}$, one common factor. This **soft constraint** will try to force zero set of the polynomial respect the local continuity of the data. Doing so also saves us from solving the sensitive homogeneous system [306], and lets us re-write the system in a more compact form $\mathbf{Aq} = \mathbf{n}$:

$$
\text{(7.24)}\quad \mathbf{A} = \begin{bmatrix}
x_1^2 & y_1^2 & z_1^2 & 2x_1y_1 & 2x_1z_1 & 2y_1z_1 & 2x_1 & 2y_1 & 2z_1 & 1 \\
x_2^2 & y_2^2 & z_2^2 & 2x_2y_2 & 2x_2z_2 & 2y_2z_2 & 2x_2 & 2y_2 & 2z_2 & 1 \\
 & & & & \vdots & & & & & \\
2x_1 & 0 & 0 & 2y_1 & 2z_1 & 0 & 2 & 0 & 0 & 0 \\
0 & 2y_1 & 0 & 2x_1 & 0 & 2z_1 & 0 & 2 & 0 & 0 \\
0 & 0 & 2z_1 & 0 & 2x_1 & 2y_1 & 0 & 0 & 2 & 0 \\
2x_2 & 0 & 0 & 2y_2 & 2z_2 & 0 & 2 & 0 & 0 & 0 \\
0 & 2y_2 & 0 & 2x_2 & 0 & 2z_2 & 0 & 2 & 0 & 0 \\
0 & 0 & 2z_2 & 0 & 2x_2 & 2y_2 & 0 & 0 & 2 & 0 \\
 & & & & \vdots & & & & &
\end{bmatrix}
\quad
\mathbf{n} = \begin{bmatrix} 0 \\ 0 \\ \dots \\ n_x^1 \\ n_y^1 \\ n_z^1 \\ n_x^2 \\ n_y^2 \\ n_z^2 \\ \dots \end{bmatrix}
\quad
\mathbf{q} = \begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \\ I \\ J \end{bmatrix}
$$

$\mathbf{A}$, only $4N \times 10$, is similar to the $\mathbf{A}'$ in § 7.5.2 and gets full rank for four or more oriented points. In fact, it is not hard to show that the equations in rows are linearly dependent, which is why we get diminishing returns when we add further constraints. Note that by removing the scale factors from the solution, we also solve the sign ambiguity problem, i.e. the solution to Eq. 7.20 can result in negated gradient vectors. To balance the contribution of normal induced constraints we introduce a scalar weight w, leading to the ten-liner MATLAB implementation as provided in Alg. 3.

[1] By "true" we mean a fit minimizing a geometric distance.

Algorithm 3: Approximate quadric fitting in 10 lines of MATLAB code.

```
1  input  : Unit normalized point set {x, y, z}, Corresponding surface normals {n_x, n_y, n_z},
            A weight coefficient ω
2  output : Quadric q = [A, B, C, 2D, 2E, 2F, 2G, 2H, 2I, J]^T
3  1 = ones(numel(x), 1);
4  0 = zeros(numel(x), 1);
5  X = [x^2, y^2, z^2, x*y, x*z, y*z, x, y, z, 1];
6  dX = [2x, 0, 0, y, z, 0, 1, 0, 0, 0; ...
7        0, 2y, 0, x, 0, z, 0, 1, 0, 0; ...
8        0, 0, 2z, 0, x, y, 0, 0, 1, 0];
9  A = [X; ω · dX];
10 N = [n_x; n_y; n_z];
11 b = [0; ω · N];
12 q = A/b;
```

In certain cases, to obtain a type-specific fit, a minor redesign of **A** tailored to the desired primitive suffices (see §. 7.5.5.3). If outliers corrupt the point set, a four-point RANSAC could be used. However, below, we present a more efficient way to calculate a solution to Eq. 7.24 rather than using a naive RANSAC on four-tuples by analyzing its solution space. The next section can also be used as a generic method to solve any fitting problem formulated as a linear system, more efficiently.

7.5.3 Quadric Detection in Point Clouds

We now explain our method to detect a generic quadric from point cloud data. We are now ready to use the explained and analyzed fitting procedure, within a new pipeline to detect and localize the quadrics from 3D data, within clutter and occlusions.

Definition 7.5.1. A basis **b** is a subset composed of a fixed number of scene points (b) and hypothesized to lie on the sought surface.

Our algorithm operates by iteratively selecting bases from an input scene. Once a basis is fixed, an under-determined quadric fit parameterizes the solution and attached to this basis, a local accumulator space is formed. All other points in the scene are then paired with this basis to vote for the potential primitive. To discover the optimal basis, we perform RANSAC, iteratively hypothesizing different basis candidates and voting locally for probable shapes. Subsequent to such joint RANSAC and

voting, we verify resulting hypotheses with efficient two-stage clustering and score functions such that multiple quadrics can be detected without repeated executions of the algorithm. We will now describe, in detail, the voting and the bases selection, respectively.

7.5.3.1 Parameterizing the Solution Space

System 7.24 describes an outlier-free closed form fit. To treat the clutter in the scene, a direct RANSAC on nine-DoF quadric appears to be trivial. Yet, it has two drawbacks: 1) evaluating the error function many times is challenging, as it involves a scene-to-quadric overlap calculation in a geometric meaningful way. 2) even with the proposed fitting, selecting random four-tuples from the scene might be slow in practice.

An alternative to RANSAC is Hough voting. However, $\mathbf{q}$ has nine DoFs and is not discretization friendly. The complexity and size of this parameter space makes it hard to construct a voting space. Instead, we will now devise a local search. For this, let $\mathbf{q}$ be a solution to the linear system in (7.24) and $\mathbf{p}$ be a particular solution. $\mathbf{q}$ can be expressed by a linear combination of homogeneous solutions $\boldsymbol{\mu}_i$ as:

$$\mathbf{q} = \mathbf{p} + \sum_{i}^{D} \lambda_i \boldsymbol{\mu}_i = \mathbf{p} + \begin{bmatrix} \boldsymbol{\mu}_1 & \boldsymbol{\mu}_2 & \cdots \end{bmatrix} \begin{bmatrix} \lambda_1 & \lambda_2 & \cdots \end{bmatrix}^T = \mathbf{p} + \mathbf{N}_A \boldsymbol{\lambda}. \tag{7.25}$$

The dimensionality D of the null space $\mathbf{N}_A$ depends on the rank of $\mathbf{A}$, which is directly influenced by the number of points used: $D = 10 - rk(\mathbf{A})$. The exact solution could always be computed by including more points from the scene and validating them, i.e. by a local search. For that reason, the fitting can be split into distinct parts: first a parametric solution is computed, such as in Eq. 7.25, using a subset of points $\mathbf{b} = \{\mathbf{x}_1, ..., \mathbf{x}_m\}$ which lie on a quadric. We refer to subset $\mathbf{b}$ as the *basis*. Next, the coefficients $\boldsymbol{\lambda}$, and thus the solution, can be obtained by searching for other point(s) $(\mathbf{x}_{m+1}, ..., \mathbf{x}_{m+k})$ which lie on the same surface as $\mathbf{b}$.

Proposition 7.5.2. *If two point sets* $\mathbf{b} = (\mathbf{x}_1, ..., \mathbf{x}_m)$ *and* $\mathbf{X} = (\mathbf{x}_{m+1}, ..., \mathbf{x}_{m+k})$ *lie on the same quadric with parameters* $\mathbf{q}$*, then the coefficients* $\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \lambda_2 & \cdots \end{bmatrix}^T$ *of the solution space (Eq. 7.25) are given by the solution of the system:*

$$(\mathbf{A}_k \mathbf{N}_A) \boldsymbol{\lambda} = \mathbf{n}_k - \mathbf{A}_k \mathbf{p} \tag{7.26}$$

where $\mathbf{A}_k$, $\mathbf{n}_k$ *are the linear constraints of the latter set* $\mathbf{b}'$ *in form of (7.24),* $\mathbf{p}$ *is a particular solution and* $\mathbf{N}_A$ *is a stacked null-space basis as in Eq. 7.25, obtained from* $\mathbf{b}$.

Proof. Let $\mathbf{q}$ be a quadric solution for the point set $(\mathbf{x}_1, ..., \mathbf{x}_m)$ and let $(\mathbf{A}_k, \mathbf{n}_k)$ represent the $4k$ quadric constraints for the k points $\mathbf{X} = (\mathbf{x}_{m+1}, ..., \mathbf{x}_{m+k})$ in form of (7.20) with the same parameters $\mathbf{q}$. As $\mathbf{x}_i \in \mathbf{X}$ by definition lies on the same quadric $\mathbf{q}$, it also satisfies $\mathbf{A}_k\mathbf{q} = \mathbf{n}_k$. Inserting Eq. 7.25 into this, we get:

$$\begin{aligned} \mathbf{A}_k(\mathbf{p} + \mathbf{N}_A\boldsymbol{\lambda}) &= \mathbf{n}_k \\ (\mathbf{A}_k\mathbf{N}_A)\boldsymbol{\lambda} &= \mathbf{n}_k - \mathbf{A}_k\mathbf{p} \end{aligned} \tag{7.27}$$

■

Solving Eq. 1 for $\boldsymbol{\lambda}$ requires a multiplication of a $4k \times 10$ matrix with a $10 \times m$ one and ultimately solving a system of $4k$ equations in m unknowns. Once $\mathbf{N}_A$ and $\mathbf{p}$ are precomputed, it is much more efficient to evaluate Eq. 7.26 for $k < m$ rather than re-solving the system (7.24). This resembles updating the solution online for a stream of points. For our case, the amount of streamed points will depend on the size of the basis, as explained below.

7.5.4 Local Voting for Quadric Detection

Given a fixed basis composed of b points ($b > 0$) as in Fig. 7.18, a parametric solution can be described. The actual solution can then be found quickly by using Prop. 7.5.2 by incorporating new points lying on the same quadric as the basis. Thus, the problem of quadric detection is de-coupled into 1) finding a proper basis and 2) searching for compatible scene points. In this section, we assume the basis is correctly found and explain the search by voting. For a fixed basis $\mathbf{b}_i$ on a quadric, we form the null-space decomposition of the under-determined system $\mathbf{A}_i\mathbf{q} = \mathbf{n}_i$. We then sample further points from the scene and compute the required coefficients $\boldsymbol{\lambda}$. Thanks to Prop. 7.5.2, this can be done efficiently. Sample points lying on the same quadric as the basis (inliers) generate the same $\boldsymbol{\lambda}$ whereas outliers will produce different values. Therefore we propose to construct a voting space on $\boldsymbol{\lambda}$ attached to basis $\mathbf{b}_i$ and cast votes to maximize the consensus, only up to the locality of the basis. Fig. 7.18 illustrates this

configuration. The size of the voting space is a design choice and depends on the size of the basis $\mathbf{b}_i$ vs. the DoFs desired to be recovered (see Tab. 7.16(b)).

While many choices for the basis cardinality are possible (and the formulation in § 7.5.3.1 allows for all), we find from Tab. 7.16(a) that using a three-point basis is advantageous for a generic quadric fit due to 1D search space as opposed to two-point vs 3D search.

Efficient computation of voting parameters for a 1D voting space Adding a fourth sample point $\mathbf{x}_4$ completes $rk(\mathbf{A}) = 10$ and a unique solution can be computed, as described above. Yet, as we will select multiple $\mathbf{x}_4$ candidates per basis, hypothesized in a RANSAC loop, an efficient scheme is required, i.e. once again, it is undesirable to re-solve the system in Eq. 7.24 for each incoming $\mathbf{x}_4$ tied to the basis. It turns out that the solution can be obtained directly from Eq. 7.25:

Proposition 7.5.3. *If the null-space is one dimensional (with only 1 unknown) it holds* $\boldsymbol{\lambda}\mathbf{N}_A = \lambda_1\boldsymbol{\mu}_1$ *and the computation in Prop. 7.5.2 reduces to the explicit form:*

$$\lambda_1 = \frac{\mathbf{A}_1\mathbf{N}_A}{\|\mathbf{A}_1\mathbf{N}_A\|^2} \cdot (\mathbf{n}_1 - \mathbf{A}_1\mathbf{p}) \tag{7.28}$$

Proof. Let us re-write Eq. 1 in terms of the null space vectors: $\lambda_1(\mathbf{A}_1\boldsymbol{\mu}_1) = \mathbf{n}_1 - \mathbf{A}_1\mathbf{p}$. A solution λ_1 can be obtained via Moore-Penrose pseudoinverse as $\lambda_1 = (\mathbf{A}_1\boldsymbol{\mu}_1)^+(\mathbf{n}_1 - \mathbf{A}_1\mathbf{p})$. Because for one-dimensional null spaces, $\mathbf{A}_1\boldsymbol{\mu}_1$ is a vector ($\boldsymbol{v}$), for which the $^+$ operator is defined as: $\boldsymbol{v}^+ = \boldsymbol{v}/(\boldsymbol{v}^T\boldsymbol{v})$. Substituting this in Eq. 7.26 gives Eq. 7.28. ■

Prop. 7.5.3 enables a very quick computation of the parameter hypothesis in the case of an additional single oriented point. A MATLAB implementation takes ca. $30\mu s$ per λ. Besides, for a three-point method, inclusion of only a single primal equation is sufficient, letting the normal of the fourth point remain unused and amenable for verification of fit. Thus, we only accept to vote a candidate if the gradient of the fit quadric agrees with the normal at fourth point:

$$\frac{\nabla\mathbf{Q}(\mathbf{x}_4)}{\|\nabla\mathbf{Q}(\mathbf{x}_4)\|} \cdot \mathbf{n}(\mathbf{x}_4) > \tau_n. \tag{7.29}$$

We typically set τ_n to a constant value, $\tau_n \approx 0.85$.

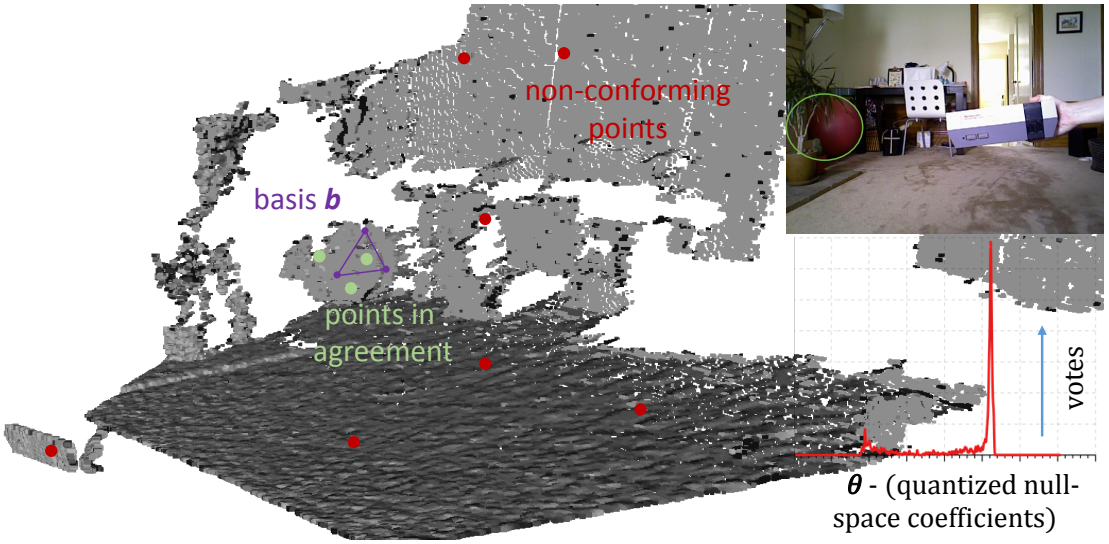


Figure 7.18: Once a basis is randomly hypothesized, we look for the points on the agreeing surface by casting votes on the null-space. The sought pilates ball (likely quadric) is marked on the image and below that lies the corresponding filled accumulator.

Quantizing λ for voting Unfortunately, λ is not quantization-friendly, as it is unbounded and has a non-linear effect on the quadric shape (Fig. 7.19). Thus, we seek to find a geometrically meaningful transformation to a bounded and well behaving space so that quantization would lead to little bias and artifacts. From a geometric perspective, each column of $\mathbf{N}_A$ in Eq. 7.25 is multiplied by the same coefficient λ, corresponding to the slope of a high dimensional line in the solution space. Thus, it could as well be viewed as a rotation. For 1D null-space, we set:

$$\theta = \text{atan2}\Big((y_2 - y_1)/(x_2 - x_1) \Big) \tag{7.30}$$

where $[x_1, y_1, \cdots]^T = \mathbf{p}$ and $[x_2, y_2 \cdots]^T$ is obtained by moving in the direction $\mathbf{N}_A$ from the particular solution $\mathbf{p}$ by an offset λ. Note that simple $tan^{-1}(\lambda)$ could work but would be more limited in the range. This new angle θ is bounded and thus easy to vote for. As the null-space dimension grows, λ starts to represent hyperplanes, still preserving the geometric meaning, i.e. for $d > 1$, different $\boldsymbol{\theta} = \{\theta_i\}$ can be found.

7.5.4.1 Hypotheses Aggregation

Up until now, we have described how to find plausible quadrics given local triplet bases.To discover the basis lying on the surface, we employed RANSAC [96], where

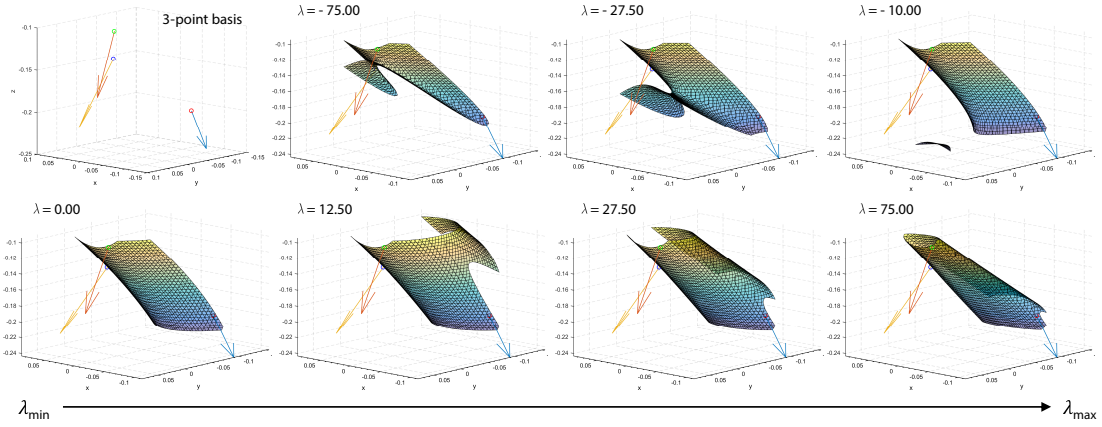


Figure 7.19: Effect of λ on the surface geometry. We compute the null-space for a fixed basis and vary λ from -75 to 75 to generate different solutions $\mathbf{q}$ along the line in the solution space. The plot presents the transition of the surface controlled by λ.

each triplet might generate a hypothesis to be verified, many of which will be similar as well as dissimilar. Thus, the final stage of our algorithm aggregates the potential detections to reduce the number of candidate surfaces and to increase the per quadric confidence. Not to sacrifice further speed, we run an agglomerative clustering similar to [35] in a coarse to fine manner: First a fine (*close*) but fast metric helps to cluster the obvious hypotheses. Second, a coarse (*far*) one is executed on these cluster centers.

Definition 7.5.4. Our distance computation is two-fold: Whenever two quadrics are close, we approximate their distances as in Eq. 7.5.4 (d_{close}), where $\mathbf{I} \in \mathbb{R}^{4\times4}$ is the identity matrix and $\mathbb{1} : \mathbb{R} \rightarrow \{0,1\}$ the indicator function. We use the pseudoinverse just to handle singular configurations. If the shapes are far, such manifold-distance becomes erroneous and we use a globally consistent metric. To do so, we define a more geometric-meaningful distance using the points on the scene itself (d_{far}):

$$
\begin{aligned}
& d_{close}(\mathbf{Q}_1, \mathbf{Q}_2) := \mathbb{1}(\|\mathbf{q}_1 - \mathbf{q}_2\|_1 < \tau) \cdot \|\mathbf{Q}_1\mathbf{Q}_2^{+} - \mathbf{I}\|_F \\
& d_{far}(\mathbf{Q}_1, \mathbf{Q}_2) := \\
& 1 - \frac{1}{K}\sum_{i=1}^{K} \mathbb{1}\left(|\mathbf{x}_i^T\mathbf{Q}_1\mathbf{x}_i| < \tau\right) \cdot \mathbb{1}\left(|\mathbf{x}_i^T\mathbf{Q}_2\mathbf{x}_i| < \tau\right) \cdot \\
& \mathbb{1}\left(1 - \mathbf{n}_i \cdot \nabla\mathbf{Q}_1(\mathbf{x}_i) < \tau_n\right) \cdot \mathbb{1}\left(1 - \mathbf{n}_i \cdot \nabla\mathbf{Q}_2(\mathbf{x}_i) < \tau_n\right).
\end{aligned} \tag{7.31}
$$

$\{\mathbf{x}_i\}$ denote the K scene samples.

Algorithm 4: Combined RANSAC & Local Voting.

input : Unit normalized point set $\mathbf{P}$, Corresponding surface normals $\mathbf{N}$, A weight coefficient ω, Minimum vote threshold s_{min}
output : Quadrics $\mathbf{Q} = \{\mathbf{q}_i\}$
$(\mathbf{S}, \mathbf{N}) \leftarrow$ Sample scene $(\mathbf{P}, \mathbf{N})$.
// seek the best global candidates
while *!satisfied* **do**
  $\mathbf{b}_i \leftarrow$ Pick a random 3 point-basis from $(\mathbf{S}, \mathbf{N})$.
  $(\mathbf{A}, \mathbf{n}) \leftarrow$ Form under-determined system using $\mathbf{b}_i$
  $(\mathbf{p}, \boldsymbol{\mu}) \leftarrow$ Perform null space decomposition
  $\mathbf{V} \leftarrow$ Initial voting space of length # bins
  $\boldsymbol{\Lambda} \leftarrow \{\}$
  // local voting
  for *all* $\mathbf{p}_i$ *in* $\mathbf{P}$ **do**
    Compute λ_i by including $\mathbf{p}_i$
    if $\left(1 - \frac{\nabla\mathbf{Q}(\mathbf{p}_i)}{\|\nabla\mathbf{Q}(\mathbf{p}_i)\|} \cdot \mathbf{n}_i < \tau\right)$ **then**
      Quantize: $\theta \leftarrow tan^{-1}(\lambda_i)$ (using atan2).
      $\mathbf{V}[\theta] + +$ // accumulate
      $\boldsymbol{\Lambda}[\theta] \leftarrow \boldsymbol{\Lambda}[\theta] \cup \lambda_i$ // best candidate in quantized space
  $\theta^* \leftarrow \arg\max_j \mathbf{V}_j$ // best local coefficient
  if $|\boldsymbol{\Lambda}[\theta^*]| > s_{min}$ **then**
    $\lambda_{\text{best}} \leftarrow \sum_k \boldsymbol{\Lambda}[\theta^*][k] \,/\, |\boldsymbol{\Lambda}[\theta^*]|$
    $\mathbf{q} \leftarrow \mathbf{p} + \lambda_{\text{best}}\boldsymbol{\mu}$ // best local solution
    $\mathbf{Q} \leftarrow \{\mathbf{Q}, \mathbf{q}\}$
$\mathbf{Q} \leftarrow$ mean of the clusters in $\mathbf{Q}$ using first distance $\mathbf{d}_{\text{close}}$ then distance $\mathbf{d}_{\text{far}}$
$\mathbf{Q} \leftarrow$ sort(score($\mathbf{Q}$))

Note that, algebraic but efficient d_{close} lacks geometric meaning, while slower d_{far} can, to a certain extent, explain the geometry. Finally, the quadrics are sorted w.r.t. their scores, evaluated pseudo-geometrically by point and normal-gradient compatibility according to definition 7.5.5:

Definition 7.5.5. The score of a quadric is defined to be:

$$S_{\mathbf{Q},\mathbf{X}} = \frac{1}{K}\sum_{i=1}^{K} \mathbb{1}\left(|\mathbf{x}_i^T\mathbf{Q}\mathbf{x}_i| < \tau\right)\mathbb{1}\left(1 - \mathbf{n}_i \cdot \nabla\mathbf{Q}(\mathbf{x}_i) < \tau_n\right) \tag{7.32}$$

Alg. 4 summarizes the entire content of this section.

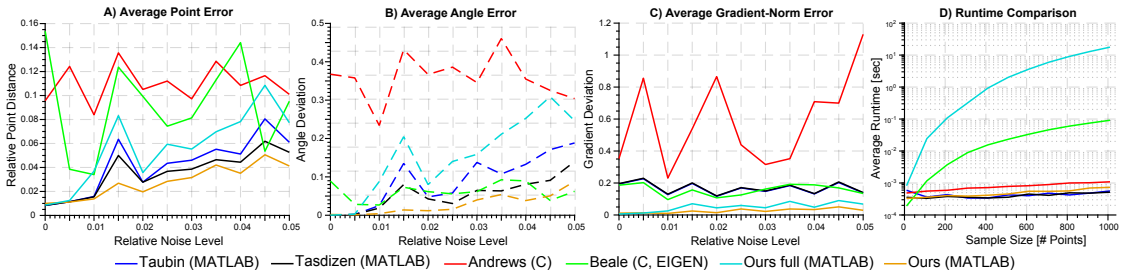


Figure 7.20: Synthetic evaluations. The plot depicts mean geometric errors on points **(a)** and mean angular errors **(b)** for different quadric fitting methods. The per point error is measured as the average point-to-mesh distance from every ground truth vertex to the fitted quadric. The angular error (dashed) is computed as the negated dot product between quadric gradient and the ground truth normal. Moreover, **(c)** shows the average error of the gradient norm compared to the ground truth and **(d)** gives speed and detection rate on synthetic data.

7.5.5 Experimental Evaluation and Discussions

7.5.5.1 Implementation details

Prior to operation, we normalize the point coordinates to lie in a unit ball to increase the numerical stability [122]. Next, we downsample the scene using a spatial voxel-grid enforcing a minimum distance of $\tau_s \cdot diam(\mathbf{X})$ between the samples ($\tau_s = 0.03$) [37]. The required surface normals are computed by the local plane fitting [132]. As planes are singular quadrics and occupy large spaces of 3D scenes, we remove them. To do so, we convert our algorithm to a type specific plane detector, which happens to be a similar algorithm to [85]. Next, influenced by the smoothness of quadrics, we use Difference of Normals (DoN) [138] to prune the points not located on smooth regions. What follows is an iterative selection of triplets to conduct the three-point RANSAC: We first randomly draw the initial point of the basis $\mathbf{x}_1$. Once $\mathbf{x}_1$ is fixed, we query the points in a large enough vicinity, whose normals differ enough to form the three-point basis $\mathbf{b}$. The rest of the points are then randomly selected respecting these criteria. To avoid degenerate configurations, we skip the basis if it does not result in a rank-9 matrix $\mathbf{A}$. In addition, we hash the seen triplets to avoid duplicate processing. This is important in reducing the bias towards such bases.

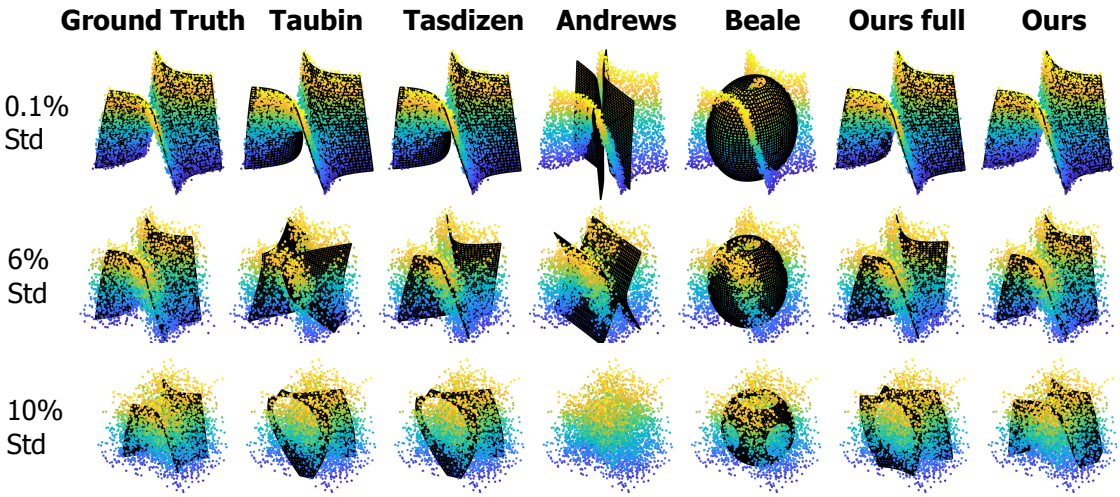


Figure 7.21: Synthetic tests at various noise levels for different fitting methods. Gaussian noise is added to the point coordinates as well as the estimated normal. The standard deviation varies from 0.11% of the visible quadric size to 10%.

7.5.5.2 Synthetic tests of fitting

To asses the accuracy of the proposed fitting, we generate a synthetic test set of multitudes of random quadrics and compare our method with the fitting procedures of Taubin [263], Tasdizen [260], Andrews [11], and Beale [22]. We propose two variants: **Ours full** will refer to Alg. 2, whereas **Ours** is the regularized one in Alg. 3.

Quantitative assessments Prior to run, we add Gaussian noise to the ground-truth vertices with $\sigma = [0\% - 5\%]$ relative to the size s of the quadric. At each noise level, ten random quadrics are tested. We perform not single but twenty fits per set. For the constrained fitting method [11] we pre-specified the type, which might not be possible in a real application. We then record and report the average point-to-mesh distance and the angle deviation as well as the runtime performances in Fig. 7.20. Although, our fit is designed to use a minimal number of points, it also proves robust when more points are added and is among the top fitters for the distance and angle errors. In addition, Fig. 7.20c shows that the errors on the gradient magnitudes obtained by our quadrics. We achieve the least errors, showing that gradient norms align well with the ground truth, favoring the validity of our approximation/regularization. Next, looking at the noise assessments, we see that our full method performs the best on low noise

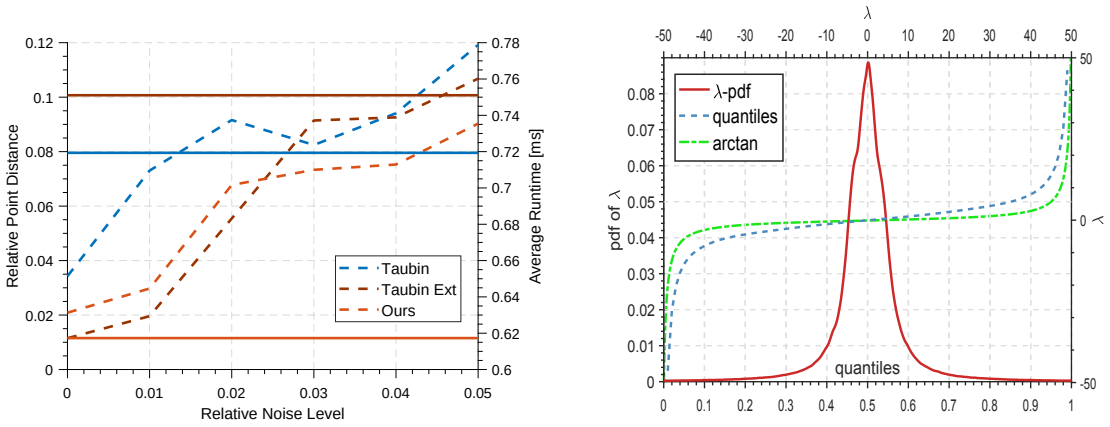


Figure 7.22: **a** (left). Effect of extended point neighborhood to the fitting. **b** (right). Statistical distribution of the solution-space coefficient and our quantization function: PDF (red curve) and inverse CDF (dashed blue-curve) of λ over collected data, and tan^{-1} function (green-line). Note that our quantization function is capable of explaining the empirical data.

levels but quickly destabilizes. This is because the system might be biased to compute correct norms rather than the solution and it has increased parameters. We believe the reason for our compact fit to work well is the soft constraint where the common scale factor acts as a weighted regularizer towards special quadrics. When this constraint cannot be satisfied, the solution settles for a very acceptable shape.

Next, we include the six neighbors of each of seven query points to perform a standard Taubin-fit. We call this *Taubin-42*. Fig. 7.22a shows that while the error of our method is on par with *Taubin-42*, we are more robust at higher noise and more efficient with a runtime advantage of ca. 22%. We also found that for a visually appealing fit, the normal alignment is crucial. Hence, we performed a visual evaluation of the fits regarding both noise robustness and fidelity of the recovered gradients.

Qualitative assessments We synthetically generated a random saddle quadric and performed a random point sampling over its surface. Next, we added Gaussian noise on the sample points and computed the normals. To resolve the sign ambiguity, each normal is flipped in the direction of ground truth gradient. We plot the results of the fitting in Fig. 7.21.

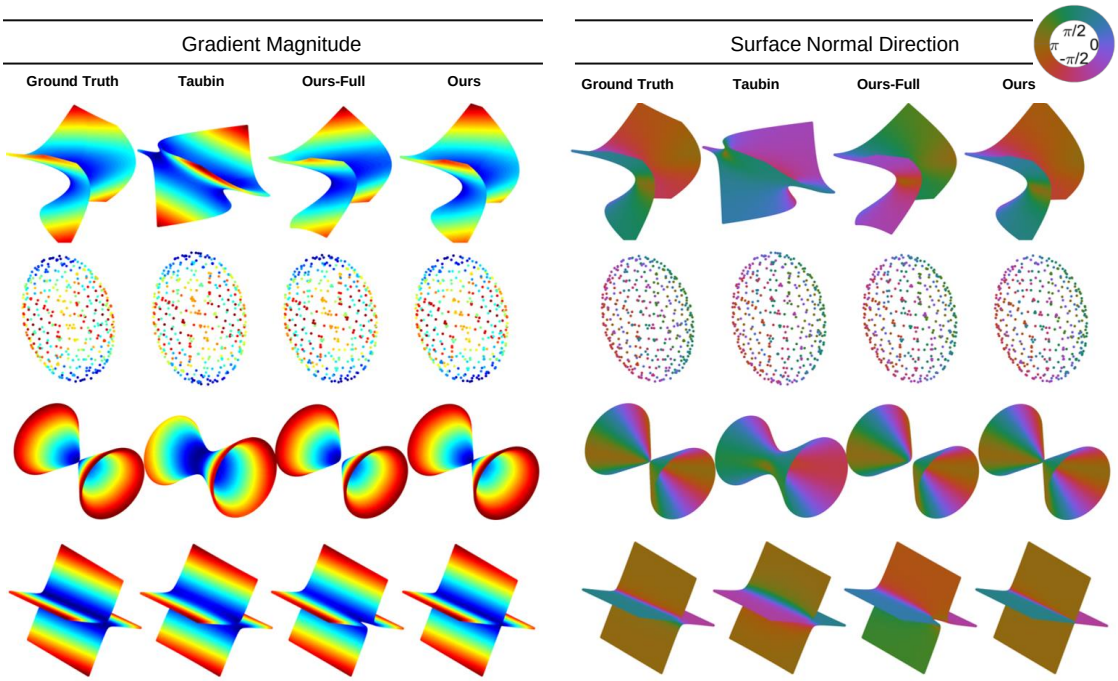


Figure 7.23: Qualitative evaluation of surface normals. Randomly generated quadrics are used as ground truth and fitting is performed. The estimates of gradient magnitude and angle is color coded on the surface. For color selection, we use a jet-like temperature map for the gradient magnitude, where blue denotes the lowest and red denotes the highest magnitudes. For the phase, an angular map as show in the color-bar is used. The ideal case is given in ground-truth against which the methods compete.

It can be seen that even in presence of little noise only some methods fail to estimate the correct geometry either because of the bias towards ellipsoids [22], or the type specific nature [11]. Our approach is able to recover the correct surface even in presence of a severe noise level. Also the effect of our regularization is visible on the last column, which possesses the best visual robustness.

It is of interest to see whether our regularized fit can estimate correct surface normals as well as direction. Thus, a second test was performed to qualitatively observe the gradient properties in more detail. For this, a series of randomly generated quadrics is fitted by Taubin's and our method and the surface gradients are analyzed both in terms of magnitude and phase, as shown in Fig. 7.23.

Due to the explicit integration of the gradients within the formulation of our framework, it can be clearly seen that the gradient direction is recovered better. Moreover,

the right side of Fig. 7.23 also shows that our approximate approach yields the expected results, while the full method could sometimes generate inconsistent gradient signs, as the scale factors are estimated individually. Finally, it is qualitatively visible in Fig. 7.23 that the magnitudes recovered by our method are compatible to the ground truth. Such improvement without sacrificing gradient quality validates the regularizing nature of our approach.

Is atan2 **a valid transformation for λ?** To assess the practical validity of the quantization, we collect a set of 2.5 million oriented point triplets from several scenes and use them as bases to form the underdetermined system **A**. We then sample the fourth point from those scenes, compute λ and establish the probability distribution $p(\lambda)$ for the whole collection to calculate the quantiles, mapping λ to bins via the inverse CDF. A similar procedure has been applied to cross ratios in [34]. We plot the findings together with the atan2 function in Fig. 7.22b and show that the empirical distribution and atan2 follow similar trends, justifying that our quantizer is well suited to the data.

7.5.5.3 Real experiments on quadric detection

Datasets Besides synthetic tests where self evaluation was possible, we assess the quality of generic primitive detection, on 3 real datasets:

1. **Our Dataset** First, because there are no broadly accepted datasets on quadric detection, we opt to collect our own. To do so, we use an accurate phase-shift stereo structured light scanner and capture 35 3D scenes of 5 different objects within clutter and occlusions. Our objects are three bending papers, helmet, paper towel and cylindrical spray bottle. Other objects are included to create clutter. To obtain the ground truth, for each scene, we generated a visually acceptable set of quadrics using 1) [240] when shapes represent known primitives 2) by segmenting the cloud manually and performing a fit, when the quadric type is not available. Each scene then contains 1-3 ground truth quadrics. This dataset has low noise, but a high amount of clutter and partial visibility due to the FOV limitations of the sensor.

2. **Large Objects** Kinect sensor is widely accepted in computer vision community. Thus, it is desirable to see the performance of our generic and type-specific fit

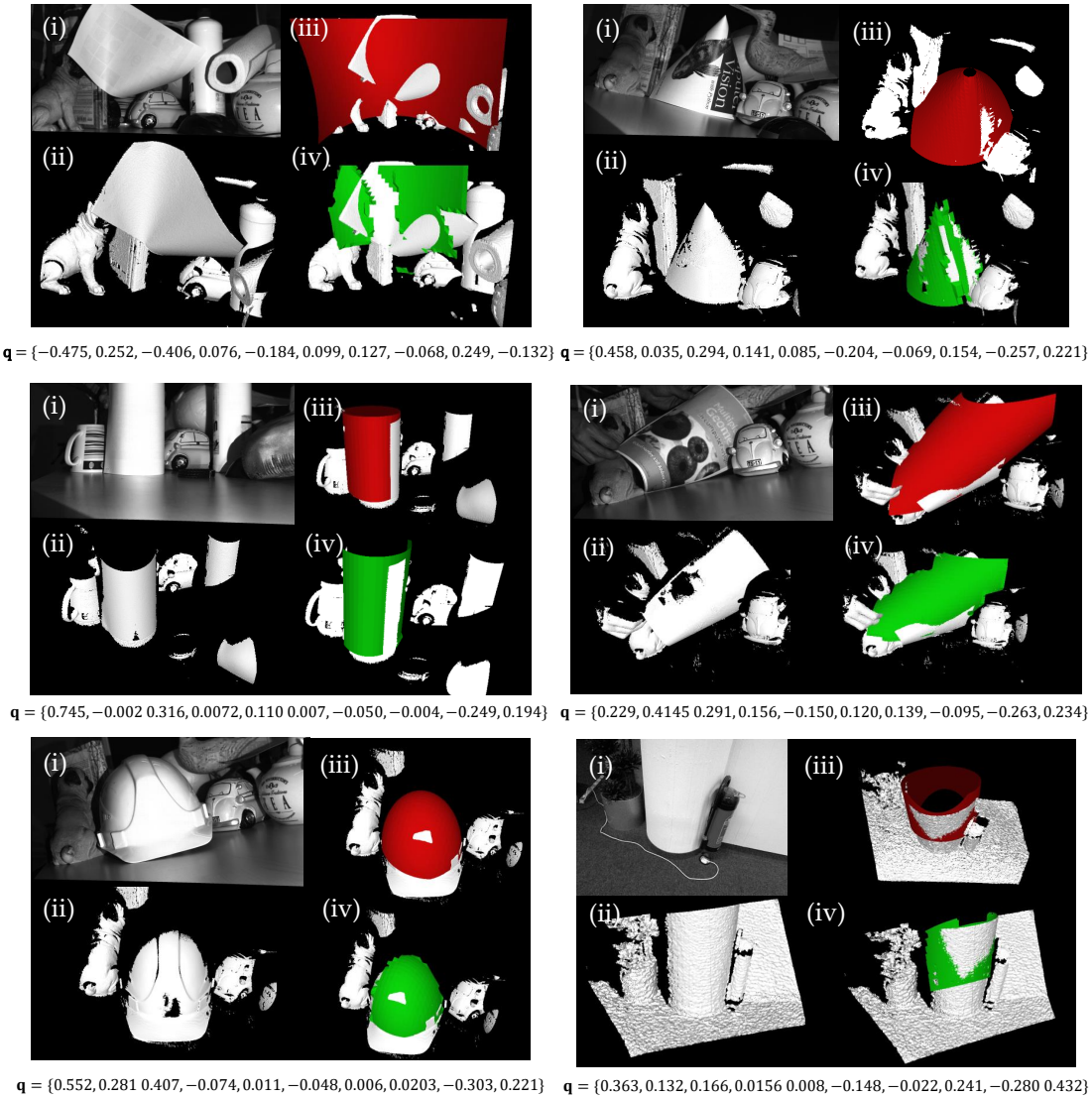
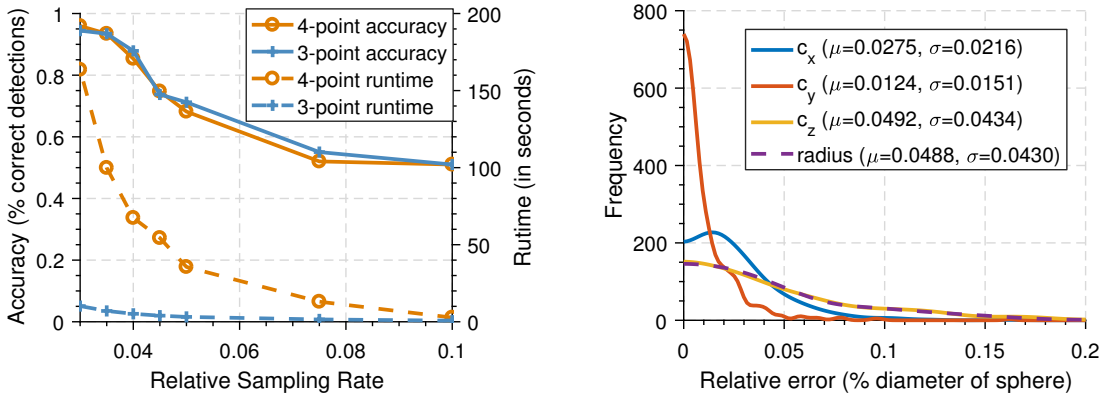


Figure 7.24: **(i)** Images captured by an industrial structured light sensor and Kinect (last image). **(ii)** 3D scene. **(iii)** Detected quadric, shown without clipping. **(iv)** Quadric in (iii) clipped to the points it lies on.

approaches on the Kinect depth images. To this end, we adapt the large objects RGB-D scan dataset of [68]. From this dataset, we sample only the scenes containing objects, that could roughly be explained by geometric primitives. These scenes include apples, globes, footballs, or other small balls. Tab. 7.5 summarizes the objects used. Example detections are also shown in Fig. 7.27. We also



(a) Speed and detection results.

(b) Distribution of errors of sphere fit.

Figure 7.25: Experiments on real datasets. Our three-point variant clearly outperforms the four-point algorithm in terms of speed, while not sacrificing quality. For three-point method, the mean errors indicate that the localization accuracy is on the levels of the sampling rate $\tau \approx 0.05$, agreeing to the success of our algorithm.

augment this dataset with a Pilates Ball sequence that we collect. This sequence involves a lot of partial visibility, clutter and fast motions (see appendix).

3. **Cylinders** Finally, we use a subset of the ITODD dataset [86], designed to evaluate object pose estimators. Our subset, *Cylidners*, includes 14 scenes of varying number of cylinders, from one to ten, as shown in Fig. 7.28a. Again, we use RGB images only to ease the visual perception.

How accurate is the detector? To assess the detection accuracy, we manually count the number of detected quadrics aligning with the ground truth in *Our Dataset*. We compared the four-point and three-point algorithms, both of which we propose. We also tried the naive nine-point RANSAC algorithm (with [263]), but found it to be infeasible when the initial hypotheses of the inlier set is not available. Fig. 7.24 visualizes the detected quadrics on various scenes. Fig. 7.25(a) presents our accuracy over different sampling rates and the runtime performance. Our three-point method is on par with the four-point variant in terms of detection accuracy, while being significantly faster. Next, we also evaluate our detector on the large objects dataset of [68] without further tuning. Tab. 7.5 shows 100% accuracy in locating a frontally appearing ellipsoidal rugby ball over a 1337 frame sequence without type prior. While

	Dataset	# Objects	Type	Occlusion	Accuracy
Pilates Ball 1	Ours	580	Generic	Yes	94.40%
Rugby Ball	[68]	1337	Generic	No	100.00%
Pilates Ball 2	[68]	1412	Sphere	Yes	100.00%
Big Globe	[68]	2612	Sphere	Yes	90.70%
Small Globe	[68]	379	Sphere	Yes	56.90%
Apple	[68]	577	Sphere	Yes	99.60%
Football	[68]	1145	Sphere	Yes	100.00%
Orange Ball	[68]	270	Sphere	Yes	93.30%

Table 7.5: Detection accuracy on real datasets.

such scenes are not particularly difficult, it is noteworthy that we manage to generate the similar quadric repeatedly at each frame within 5% of the quadric diameter.

How fast is it? The speed of our algorithm is influenced by closed form fitting, RANSAC and local voting. Thus, we evaluate the fit and detection separately. Fig. 7.20d shows the runtime of fitting. Our method scales linearly due to the solution of a $4N \times 10$ system, but it is the fastest approach when < 300 points are used. Thus, it is more preferred for a minimal fit. Fig. 7.25(a) then presents the order of magnitude speed gain, when our four-point C++ version is replaced by three-points without accuracy loss. The final runtime is in the range of 1-2 seconds, the fastest known method in segmentation free detection of quadrics.

How accurate is the fit? To evaluate the pose accuracy on real objects, we use closed geometric objects of known size from the aforementioned datasets and report the distribution of the errors. We choose *football* and *pilates ball 1* as it is easy to know their geometric properties (center and radius). We compare the radius to the true value while the center is compared to the one estimated from a non-linear sphere refinement. Our results are depicted in Fig. 7.25(b), where the errors successfully remain about the used sampling rates ($\tau \approx 0.05$). This is as best as we could get. In order to provide an intuition into our dataset, as well as qualitatively assessing our method, we now visualize the detection results on the Pilates Ball sequence, in Fig. 7.26.

Type-specific detection It is remarkably easy to convert our algorithm to a type specific one by re-designing matrix **A**. Here, we propose a sphere-specific detector. Let

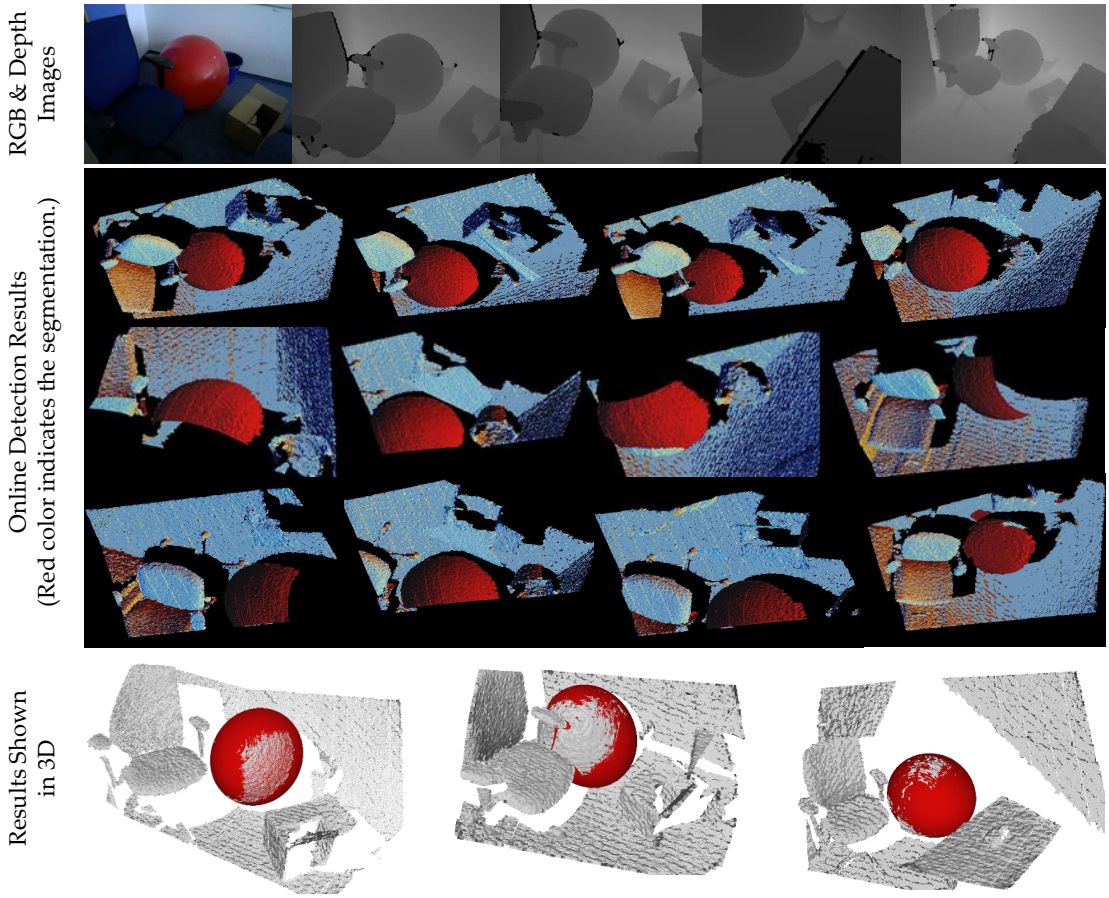


Figure 7.26: Quadric detection in real depth sequence. **(upper row).** An RGB-D camera viewing a pilates ball is moved around the object. RGB image is shown only for perceptual reasons; our method only uses the depth (in fact 3D) information. **(middle rows).** Detection results in different frames of the sequence. The detected shape is segmented by pruning away the points that are not incident with the quadric (due to the distance metrics provided in § 7.5.3). Inlier 3D points are then rendered in red, in real-time. Note that, we do not perform any refinement, and even then, occlusions and clutter are handled gracefully and segmentation mask appears to be decent across the sequence. **bottom row.** Three offline renderings of the detections, shown in pure 3D space. Here, before rendering, the quadric is converted to a mesh representation, by Marching Cubes algorithm [176], and plotted as a meshed surface.

us write any sphere in the following matrix form:

$$\mathbf{Q}_s = \begin{bmatrix} \mathbf{I}_3 & -\mathbf{c} \\ -\mathbf{c}^T & \|\mathbf{c}\| - r^2 \end{bmatrix} \tag{7.33}$$

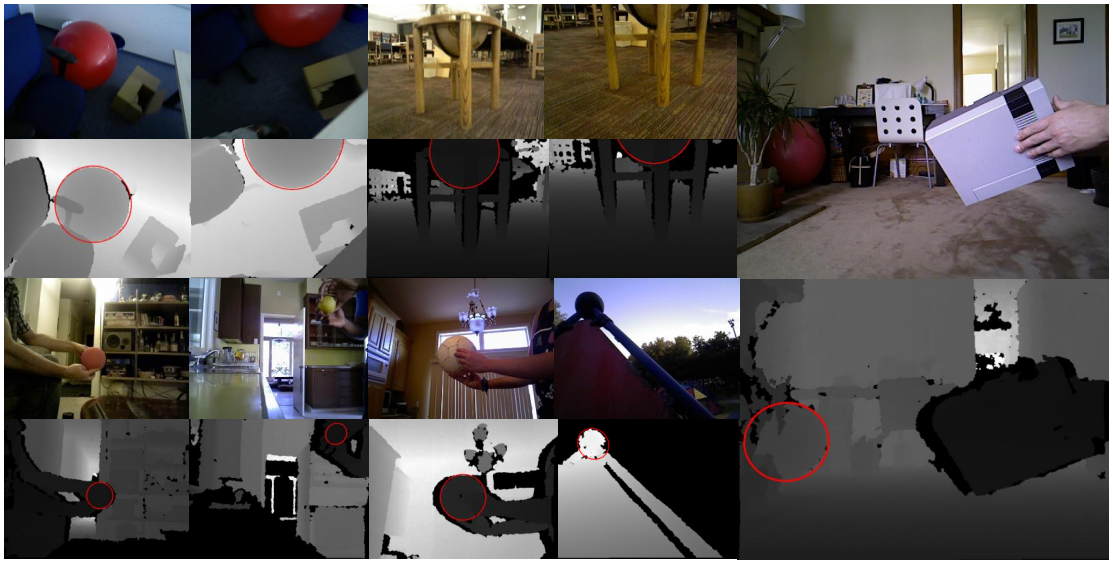


Figure 7.27: Qualitative visualizations of sphere detection in the wild: Our algorithm successfully detects primitives in difficult scenarios including clutter, occlusions and large distances. Note that the sphere is detected in 3D only using the point clouds of depth images and we draw the apparent contour of the quadric. The RGB pictures are also included in the top row to ease the visual perception.

where $\mathbf{c}$ and r are the geometric parameters (center and radius) of the sphere. Rotation does not affect spheres and our $\mathbf{Aq} = \mathbf{b}$ formulation in § 7.5.2 then simplifies to:

$$\mathbf{A} = \begin{bmatrix} \|\mathbf{x}_1\|^2 & 2x_1 & 2y_1 & 2z_1 & 1 \\ \|\mathbf{x}_2\|^2 & 2x_2 & 2y_2 & 2z_2 & 1 \\ & & \vdots & & \\ 2\mathbf{x}_1 & & \mathbf{I}_3 & & \mathbf{0}_3 \\ 2\mathbf{x}_2 & & \mathbf{I}_3 & & \mathbf{0}_3 \\ & & \vdots & & \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \mathbf{n}_1 \\ \mathbf{n}_2 \\ \vdots \end{bmatrix}, \mathbf{q} = \begin{bmatrix} A \\ B \\ C \\ D \\ E \end{bmatrix} \tag{7.34}$$

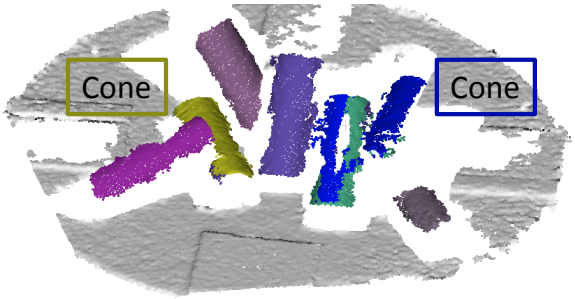
Due to the geometric interpretability, at the scoring phases, we can use the point-to-sphere distance as:

$$d_{\text{point}\to\text{sphere}}(\mathbf{p}, \mathbf{q}) = |(\|\mathbf{p} - \mathbf{c}\|_2 - r)| \tag{7.35}$$

where $\mathbf{p}$ is the point to compute the distance. A sphere to sphere distance (used in clustering) can be obtained by:

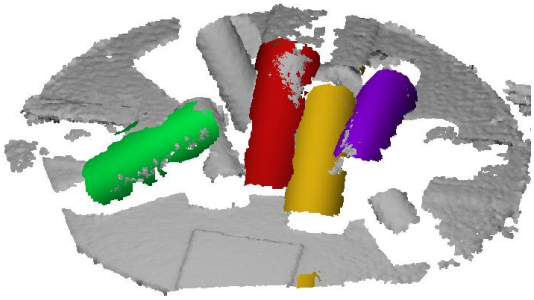
$$d_{\text{sphere}\to\text{sphere}}(\mathbf{q}_1, \mathbf{q}_2) = \frac{1}{2}(|r_1 - r_2| + \|\mathbf{c}_1 - \mathbf{c}_2\|_2) \tag{7.36}$$

(a) An ITODD scene composed of many cylinders



(b) Schnabel et. al. can mis-classify or -segment certain types, thereby fitting the wrong primitives.

(c) 6DoF PPF pose estimation using Cylinder as model [5,11]



(d) Our model-free, unsupervised result
Only the surface parts overlapping the point set are shown.

Figure 7.28: Multiple cylinder detection in clutter and occlusions: Our approach is type agnostic and uninformed about cylinders.

$(\mathbf{c}, r)$ can always be obtained from the quadric form $\mathbf{Q}_s$ as described in Eq. 7.33.

Note that $rk(\mathbf{A}) = 4$ if one point is available, leaving only one free parameter which forms a single dimensional null-space. Geometrically, this means that the radius cannot be resolved from a single point. Yet, by fixing another point, one can vote locally as explained in § 7.5.4. While at this stage Drost and Ilic [85] prefer to vote for radius explicitly, we vote for the null-space coefficient. The difference is that [85] involves trigonometric computations before the voting stage, but vote linearly for the geometric parameter, whereas we keep linearity until the voting stage but vote for the non-linear angle θ corresponding to λ. Our approach evaluates far less trig functions (only one *atan2*). We plug this specific fit into our detector without changing other parts and evaluate it on scenes from [68] which contains spherical everyday objects. Tab. 7.5 summarizes the dataset and reports our accuracy while Fig. 7.27 qualitatively shows that our sphere-specific detector can indeed operate in challenging real scenarios. Our algorithm is able to detect a sphere on many difficult

PPF3D	PPF3D-E	PPF3D-E-2D	S2D [281]	RANSAC [214]	Ours
72%	73%	74%	24%	86%	41.9%

Table 7.6: Results on ITODD [86] cylinders: Even without looking for a cylinder, we outperfom the model based [281].

cases, as long as the sphere is partially visible. We also do not have to specify the radius as unlike many Hough transform based methods. Note that, due to reduced basis size ($b = |\mathbf{b}| = 1$) this type specific fit can meet real-time criteria.

Comparison to model based detectors The literature is overwhelmed by the number of 3d model based pose estimation methods. Hence, we decide to compare our model-free approach to the model based ones. For that, we take the cylinders subset of the recent ITODD dataset [86] and run our generic quadric detector without training or specifying the type. Visuals of different methods are presented in Fig. 7.28 whereas detection performance are reported in Tab. 7.6. Our task is not to explicitly estimate the pose. Thus, we manually accept a hypothesis if ICP [97] converges to a visually pleasing outcome. Note, multiple models are an important source of confusion for us, as we vote on generic quadrics. However, our algorithm outperforms certain detectors, even when we are solving a more generic problem as our shapes are allowed to deform into geometries other than cylinders.

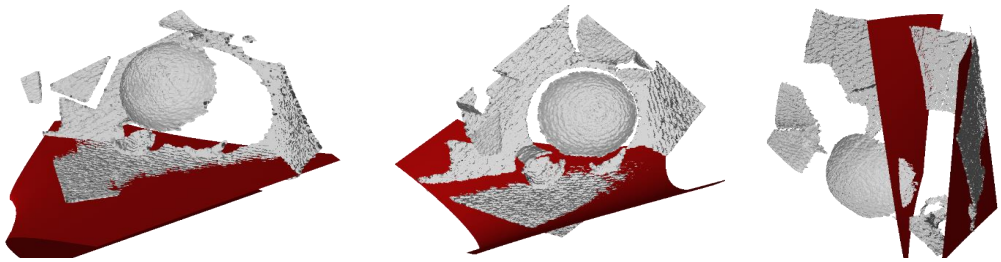


Figure 7.29: When planes gather a majority of the votes, our method approximates them by a close non-degenerate quadric surface. This is similar to representing a large planar football field using Earth's surface.

Limitations Unless made specific, our method is surpassed by type-specific fits in detection rate since solving the generic problem is more difficult. It remains an open

issue to bring the performance to the levels of type-specific fits. Nevertheless, if the design matrix $\boldsymbol{A}$ targets a specific type, we perform even better. Degenerate cases are also difficult for us as shown in Fig. 7.29, but we always find a non-degenerate configuration good-enough to approximate the primitive.

7.6 Discussion

We have proposed several object detectors for finding the orientation and location of 3D objects in noisy and cluttered point clouds. The first stage began with improving the geometric hashing framework of point pair features by multitudes of contributions, including weighted voting, segment-level processing, object and scene dependent database weighting and probabilistic modeling. We have then moved onto learning based modern techniques, that can still operate on the geometric PPF representation. Even though the use of the learned descriptors advances the state of the art in RANSAC-like 3D matching, because of the requirement of real training data or the necessity to fine tune, their appliance to CAD-based computer vision is not yet immediate.

We have then moved onto a different approach, where instead of estimating the pose of a rigid body, we estimate the parameters of bases components that are common to a variety of man-made objects. There, we have given a novel, linear formulation of quadric fitting for oriented point quadruplets. We thoroughly analyzed this fit and devised an efficient null-space voting which uses three pieces of point primitives plus a simple local search instead of a full four oriented point fit. Together, the fitting and voting establish the minimalist cases known up to now - three oriented points, potentially paving the way towards real-time operation. While this detector targets a generic surface, we can, optionally, convert to a type-specific fit to boost speed and accuracy.

CHAPTER 8

POSE GRAPH PROCESSING

"The real voyage of discovery consists not in seeking new landscapes but in having new eyes."

— Marcel Proust, *In Search of Lost Time, 1923*

The final stage of our pipeline involves 3D pose graph optimization. It is noteworthy that PGO is not specific to our problem and finds a variety of use cases in 3D and 2D computer vision. For instance, the ability to reduce drift while navigating autonomously comes with PGO and is now a key technology not only for 3D reconstruction and digitization but also used in self driving cars, unmanned aerial vehicles (UAV), robot guidance, augmented reality, sensory network localization and more. This ubiquitous appliance is due to the fact that vision sensors can provide cues to directly solve 6DoF pose estimation problem and does not necessitate external tracking input, such as imprecise GPS, to ego-localize. The drawbacks of non-vision based solutions coerces many algorithms to operate directly on pose data, that is inherently noisy. Typically, as mentioned in Chapter 5, there are two kinds of scenarios: 1. sequential estimation of rotations and translations resulting in ordered pose data, 2. capturing with random order, where the user or the robot is free to determine the location of the shot. The former creates a temporally related data with edges structured as a chain. This should be treated differently from the latter, where the pose graph can

have arbitrary connections due to the freedom in camera positioning.

In this chapter of this thesis, we develop two tools for processing ordered and unordered pose graphs, respectively. We will first explain our pose graph denoising strategy considering the temporal arrangement of poses. Later on, we will present a generalized statistical pose graph optimization algorithm that can be applied to arbitrary pose graphs, and initialize complex algorithms, such as bundle adjustment. Both of these tools are designed to operate on pose graph directly and avoid 3D structure. Thereby, they are better used as initializers, or smoothers, accompanying a final structure-guided optimization stage.

8.1 Local Geodesic Regression for Filtering Pose Chains

A time-varying, sequential motion can be interpreted on the pose space $\mathbb{DH}_1$ as a high dimensional trajectory. Generally, due to lack of constraints on physical motion or tracking errors, this trajectory is highly non-linear, non-Gaussian and includes outliers. Moreover, it is non-uniformly sampled because of velocity changes or sudden jumps. We take this non-ideal setting into account and propose a non-parametric path smoothing algorithm which is robust, flexible, intuitive and could naturally benefit from the availability of the uncertainties in the pose estimation [50].

Weighted local linear regression Locally, we treat the trajectory as a linear one, and seek to find the linear association of the data points $\mathbf{X}$ (poses) to the responses $\mathbf{Y}$:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \epsilon \tag{8.1}$$

A common way to discover this local relationship is using the *Generalized Least Squares* minimization

$$\boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta}} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T \mathbf{W} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}). \tag{8.2}$$

with diagonal weight matrix $\mathbf{W}$. We will later show how these weights can be adjusted. The solution to Eq. 8.2 is given by weighted least squares regression

$$\boldsymbol{\beta}^* = (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W} \mathbf{Y}. \tag{8.3}$$

Equivalently, when the data is rather unevenly distributed, one likes to prevent distinguishing the predictor and response, rather looking for meaningful linear projections (e.g. maximum variance) instead. As a first step, a hyper-line can be obtained from the first principal component of the data points $\mathbf{X}$. Let $\mathbf{X} = \mathbf{USV}^T$ denote the singular value decomposition (SVD) of $\mathbf{X}$, with right singular vectors $\mathbf{V}$. $\mathbf{V\Delta V}^T$ gives the spectral decomposition of the covariance matrix $\mathbf{XX}^T$ with the non-negative eigenvalues $\mathbf{\Delta} = diag(\lambda_1..\lambda_p)$. Weights are then transferred directly to the covariance matrix by

$$\mathbf{C} = \frac{1}{2\|diag(\mathbf{W})\|}(\mathbf{X}-\boldsymbol{\mu})\mathbf{W}(\mathbf{X}-\boldsymbol{\mu})^T. \tag{8.4}$$

The columns of $\mathbf{V}$ give the orthonormal set of eigenvectors and $\mathbf{Xv}_j$, the j^{th} principal component. On a subsequent step, the local input data can be projected onto the first principal subspace resulting in the principal covariates $\mathbf{XV}_k := \{\mathbf{Xv}_1..\mathbf{Xv}_k\}$. To smooth the trajectory, the central point $\mathbf{c}$ is projected onto the PC-line. Note that, this fit assumes a local Gaussian distribution, while the global distribution can be arbitrary.

Extension to pose chains Our method uses dual quaternions as explained in § 2.3.5 to parameterize camera poses. While PCA schemes hold for the typical Euclidean spaces, they donot generalize to manifold valued functions, such as dual quaternions, because these spaces are not necessarily Euclidean. Buss and Fillmore [53] show that even regressing a great arc on the quaternion hypersphere has ambiguities. Luckily, for the case of local regression, the central point of fitting is known and this enables us to map the immediate neighborhood onto the tangent space of dual quaternions $\mathbb{T}_{\mathbf{x}}\mathbb{G}$, thereby circumventing the non-Euclidean nature. Thanks to the manifold structure, locally, the tangent space behaves Euclidean and we can perform the fit. To smooth the curve, the central point is projected onto

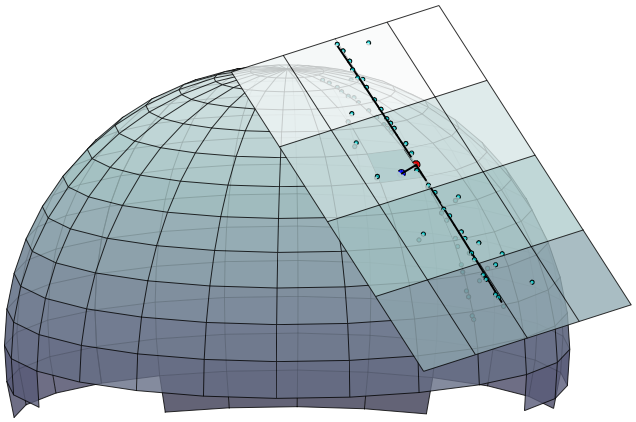


Figure 8.1: Robust PC-regression on tangent space (the concept illustrated in 3D, rather than 8D).

Algorithm 5: irls_wpca : IRLS for Weighted PCA.

input : Local set of dual quaternions $\mathbf{X} = \{\mathbf{X}_i\}$, # Iterations N, Prior weights $\mathbf{w}^0 = \{w_i\}$
output: PCA line $\mathbf{l}$ with projections $\mathbf{X}_{fit}$
$\mathbf{w} \leftarrow \mathbf{w}_0$
for $i = 1 : N$ **do**
    $\{\mathbf{X}_{proj}, \mathbf{l}\} \leftarrow$ weighted_pca($\mathbf{X}$,$\mathbf{w}$)
    Update $\mathbf{w}$ using (8.6)
    $\mathbf{w} \leftarrow \mathbf{w} \cdot \mathbf{w}^0 / \|\mathbf{w} \cdot \mathbf{w}^0\|$ // dampen the estimates

Algorithm 6: Manifold PC-Local Regression.

input : Set of dual quaternions $\mathbf{X} = \{\mathbf{X}_i\}$, Kernel size K, Prior weights $\mathbf{w}^0 = \{w_i\}$ for local window
output: Filtered poses $\mathbf{X}^f = \{\mathbf{X}_i^f\}$
$\mathbf{X}^f \leftarrow []$
for $\boldsymbol{x}_i \in \boldsymbol{X}$ **do**
    $\mathbf{X}_\Omega \leftarrow \{\mathbf{x}_k\} \in \Omega_i$
    $\mathbf{X}_\Omega^t \leftarrow \log(\mathbf{X}_\Omega)$
    $\mathbf{X}_\Omega^{proj} \leftarrow$ irls_wpca($\mathbf{X}_\Omega^t, \mathbf{w}_0$)
    $\mathbf{x}_i^f \leftarrow \exp(\mathbf{X}_\Omega^{proj}(i))$
    $\mathbf{X}^f \leftarrow \mathbf{X}^f \cup \mathbf{x}_i^f$

the regressed 8D-line and mapped back onto the manifold using the exponential map. Naturally, the 8D data points, which are closer to the center of the local model $\mathbf{c}$ are more relevant for the fit, as the linearity decreases with the extent. Therefore we multiply each data point by a Gaussian prior function to downweight the points based on their relative position:

$$w_i^0 = \exp\Big(-\frac{1}{2}(\mathbf{x}_i - \mathbf{c})^T \mathbf{D} (\mathbf{x}_i - \mathbf{c})\Big), \tag{8.5}$$

where $\mathbf{D}$ is a positive semi-definite distance metric explaining the region of influence, a.k.a. *locality*. Moreover, oftentimes the pose space could contain outliers, which a naive fit cannot deal with. Therefore, we introduce to use a re-weighted procedure, in which the residuals of the current fit is used to update the weights for the next iteration. This is commonly referred as *iteratively reweighted least squares (IRLS)*. We use a simple distance based weight update:

$$\mathbf{w}_{i+1} = 1/\max\Big(\delta, \frac{1}{K}\sum_{k=1}^{K} |\mathbf{r}_i^k|\Big), \tag{8.6}$$

where δ is a small number, preventing division by zero and $\{\mathbf{r}_i^k\}$ are the residuals at iteration i. Alg. 6 summarizes our final implementation which is illustrated in Fig. 8.1. It is once again worth reminding that such a denoising algorithm restricts the acquisition order. In the following, we will be moving onto a more generic scenario, where no particular trajectory is assumed.

8.2 TG-MCMC: Bayesian Initialization for Pose Graph Optimization via Bingham Distributions

Many of the problems in the domains of 3D vision can now be addressed by tailor-made pipelines such as SLAM (Simultaneous Localization and Mapping), SfM (Structure From Motion) or multi robot localization (MRL) [57, 153]. Nowadays, thanks to the resulting reliable estimates of rotations and translations, many of these pipelines rely on some form of an optimization, such as bundle adjustment (BA) [272] or 3D global registration [36, 135], that can globally consider the acquired measurements. Holistically, these methods belong to the family of *pose graph optimization* (PGO) [158]. Unfortunately, many of PGO post-processing stages, which take in to account both camera poses and 3D structure, are too costly for online or even soft-realtime operation. This bottleneck demands good solutions for PGO initialization, that can relieve the burden of the joint optimization.

We now address the particular problem of initializing PGO, in which multiple local measurements are fused into a globally consistent estimate, without resorting to the costly bundle adjustment or optimization that uses structure. In specifics, let us consider a finite simple directed graph $G = (V, E)$, where vertices correspond to reference frames and edges to the available relative measurements as shown in Figures 8.2(a), 8.2(b). Both vertices and edges are labeled with rigid motions representing absolute and relative poses, respectively. Each absolute pose is described by a homogeneous transformation matrix $\{\mathbf{M}_i \in SE(3)\}_{i=1}^n$. Similarly, each relative pose is expressed as the transformation between frames i and j, $\mathbf{M}_{ij}$, where $(i, j) \in E \subset [n] \times [n]$. The labeling of the edges is such that if $(i, j) \in E$, then $(j, i) \in E$ and $\mathbf{M}_{ij} = \mathbf{M}_{ji}^{-1}$. Hence, we consider G to be undirected. With a convention as shown in Fig. 8.2(c), the link between absolute

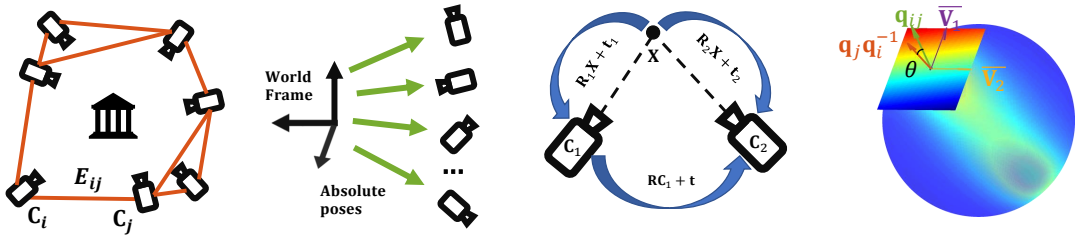


Figure 8.2: From left to right: **(a)** Initial pose graph of relative poses. **(b)** Absolute poses w.r.t. common frame. **(c)** Convention used to describe the pairwise relationships. **(d)** A sample Bingham distribution and the rotational components.

and relative transformations is encoded by the *compatibility constraint*:

$$\mathbf{M}_{ij} \approx \mathbf{M}_j \mathbf{M}_i^{-1}, \forall i \neq j \tag{8.7}$$

Primarily motivated by Govindu *et al.* [111], *rigid-motion synchronization* initializes PGO by computing an estimate of the vertex labels $\mathbf{M}_i$ (absolute poses) given enough measurements of the ratios $\mathbf{M}_{ij}$. In other words, it tries to find the absolute poses that best fit the relative pairwise measurements. Typically, in order to remove the gauge freedom, one of the poses is set to identity $\mathbf{M}_0 = \mathbf{I}$ and the problem reduces to recovering $n-1$ absolute poses. The solution is the state of the art method to initialize, say SfM [58, 153, 274] thanks to the good quality of the estimates.

The pose graph optimization problem is often formed as non-convex optimization problems, opening up room for different formulations and approaches. Direct methods try to compute a good initial solution [13, 14, 58, 99], which are then refined by iterative techniques [112, 269]. Robustness to outlier relative pose estimates is also crucial for a better solution [57, 59, 60, 120, 274]. The structure of our peculiar problem allows for global optimization, when isotropic noise is assumed and under reasonable noise levels as well as well connected graph structures [47, 48, 90, 99, 230, 290]. It is also noteworthy that, even though the problem has been previously handled with statistical approaches [273], up until now, to the best of our knowledge, estimation of uncertainties in PGO initialization are never truly considered.

In this part of our work, we look at the graph optimization problem from a probabilistic point of view. We begin by representing the problem on the Cartesian product of the true Riemannian manifold of quaternions and Euclidean manifold of translations. We model rotations with Bingham distributions [29] and translation with

Gaussians. The probabilistic framework provides two important features: (i) we can align the modes of the data (relative motions) with the posterior parameters, (ii) we can quantify the uncertainty of our estimates by using the posterior predictive distributions. In order to achieve these goals, we come up with efficient algorithms both for maximum a-posteriori (MAP) estimation and posterior sampling: 'tempered' geodesic Markov Chain Monte Carlo (TG-MCMC). Controlled by a single parameter, TG-MCMC can either work as a standard MCMC algorithm that can generate samples from a Bayesian posterior, whose entropy, or covariance, as well as the samples themselves, provide necessary cues for uncertainty estimation - both on camera poses and possibly on the 3D structure, or it can work as an optimization algorithm that is able to generate samples around *the global optimum* of the MAP estimation problem. In this perspective, TG-MCMC bridges the gap between Bayesian geodesic Markov Chain Monte Carlo (gMCMC) and non-convex optimization, as we will theoretically present.

In a nutshell, our contributions are as follows:

- Novel probabilistic model using Bingham distributions in pose averaging for the first time,
- Tempered gMCMC: Novel tempered MCMC algorithm for global optimization and sampling on the manifolds using the known geodesic flow,
- Theoretical understanding and convergence guarantees for the devised algorithm,
- Strong experimental results justifying the validity of the approach.

8.2.1 The Proposed Model

In this section, we will describe our proposed model for PGO initialization. We consider the situation where we observe a set of noisy pairwise pose estimations $\mathbf{M}_{ij}$, represented by *augmented quaternions* as $\{\mathbf{q}_{ij} \in \mathbb{S}^3 \subset \mathbb{R}^4, \mathbf{t}_{ij} \in \mathbb{R}^3\}$. The indices $(i, j) \in E$ run over the edges the graph. We assume that the observations $\{\mathbf{q}_{ij}, \mathbf{t}_{ij}\}_{(i,j)\in E}$ are generated by a probabilistic model that has the following hierarchical structure:

$$\mathbf{q}_i \sim p(\mathbf{q}_i), \qquad \mathbf{t}_i \sim p(\mathbf{t}_i), \qquad \mathbf{q}_{ij}|\cdot \sim p(\mathbf{q}_{ij}|\mathbf{q}_i, \mathbf{q}_j), \qquad \mathbf{t}_{ij}|\cdot \sim p(\mathbf{t}_{ij}|\mathbf{q}_i, \mathbf{q}_j, \mathbf{t}_i, \mathbf{t}_j), \tag{8.8}$$

where the *latent variables* $\{\mathbf{q}_i \in \mathbb{S}^3\}_{i=1}^n$ and $\{\mathbf{t}_i \in \mathbb{R}^3\}_{i=1}^n$ denote the true values of the *absolute poses* and *translations* with respect to a common origin, corresponding to $\mathbf{M}_i$ of Eq. 8.7. Here, $p(\mathbf{q}_i)$ and $p(\mathbf{t}_i)$ denote the *prior distributions* of the latent variables, and the product of the densities $p(\mathbf{q}_{ij}|\cdot)$ and $p(\mathbf{t}_{ij}|\cdot)$ forms the *likelihood* function.

By respecting the natural manifolds of the latent variables, we choose the following prior model:$\mathbf{q}_i \sim \mathscr{B}(\Lambda_p, \mathbf{V}_p)$, $\mathbf{t}_i \sim \mathscr{N}(\mathbf{0}, \sigma_p^2\mathbf{I})$ where Λ_p, $\mathbf{V}_p$, and σ_p^2 are the prior model parameters, which are assumed to be known. We then choose the following model for the observed variables:

$$\mathbf{q}_{ij}|\mathbf{q}_i, \mathbf{q}_j \sim \mathscr{B}(\Lambda, \mathbf{V}(\mathbf{q}_j\bar{\mathbf{q}}_i)), \qquad \mathbf{t}_{ij}|\mathbf{q}_i, \mathbf{q}_j, \mathbf{t}_i, \mathbf{t}_j \sim \mathscr{N}(\boldsymbol{\mu}_{ij}, \sigma^2\mathbf{I}), \tag{8.9}$$

where Λ is a fixed, $\mathbf{V}$ is a matrix-valued function that will be defined in the sequel; $\boldsymbol{\mu}_{ij}$ denotes the expected value of $\mathbf{t}_{ij}$ provided that the values of the relevant latent variables $\mathbf{q}_i$ $\mathbf{q}_j$, $\mathbf{t}_i$, $\mathbf{t}_j$ are known, and has the form:

$$\boldsymbol{\mu}_{ij} \triangleq \mathbf{t}_j - (\mathbf{q}_j\bar{\mathbf{q}}_i)\mathbf{t}_i(\mathbf{q}_i\bar{\mathbf{q}}_j). \tag{8.10}$$

With this modeling strategy, we are expecting that $\mathbf{t}_{ij}$ would be close to the true translation $\boldsymbol{\mu}_{ij}$ that is a deterministic function of the absolute poses. Our strategy also lets $\mathbf{t}_{ij}$ differ from $\boldsymbol{\mu}_{ij}$ and the level of this flexibility is determined by σ^2.

Constructing Bingham distribution on any given mode $\mathbf{q} \in \mathbb{S}^3$ requires finding a frame bundle $\mathbb{S}^3 \to \mathscr{F}\mathbb{S}^3$ composed of the unit vector (the mode) and its orthonormals. Being *parallelizable* ($d = 1, 2, 4$ or 8), manifold of unit quaternions enjoys an injective homomorphism to the orthonormal matrix ring composed of the orthonormal basis [251]. Thus, we define $\mathbf{V} : \mathbb{S}^3 \mapsto \mathbb{R}^{4\times4}$ as follows:

$$\mathbf{V}(\mathbf{q}) \triangleq \begin{bmatrix} q_1 & -q_2 & -q_3 & q_4 \\ q_2 & q_1 & q_4 & q_3 \\ q_3 & -q_4 & q_1 & -q_2 \\ q_4 & q_3 & -q_2 & -q_1 \end{bmatrix}. \tag{8.11}$$

It is easy to verify that $\mathbf{V}(\mathbf{q})$ is orthonormal for every $\mathbf{q} \in \mathbb{S}^3$. $\mathbf{V}(\mathbf{q})$ further gives a convenient notation for representing quaternions as matrices paving the way to linear operations, such as quaternion multiplication or orthonormalization without pesky Gram-Schmidt processes. By using the definition of $\mathbf{V}(\mathbf{q})$ and assuming that the

diagonal entries of Λ are sorted in decreasing order, we have the following property:

$$\arg\max_{\mathbf{q}_{ij}} \left\{ p(\mathbf{q}_{ij}|\mathbf{q}_i, \mathbf{q}_j) = \mathscr{B}(\Lambda, \mathbf{V}(\mathbf{q}_j \bar{\mathbf{q}}_i)) \right\} = \mathbf{q}_j \bar{\mathbf{q}}_i. \tag{8.12}$$

Similar to the proposed observation model for the relative translations, given the true poses $\mathbf{q}_i, \mathbf{q}_j$, this modeling strategy sets the most likely value of the relative pose to the deterministic value $\mathbf{q}_j \bar{\mathbf{q}}_i$, and also lets $\mathbf{q}_{ij}$ differ from this value up to the extent determined by Λ. This configuration is illustrated in Fig 8.2(d).

Representing $SE(3)$ in the form of a quaternion-translation parameterization, we can now formulate the motion-synchronization problem as a probabilistic inference problem. In particular we are interested in the following two quantities:

1. The maximum a-posteriori (MAP) estimate: $(\mathbf{Q}^\star, \mathbf{T}^\star) = \arg\max_{\mathbf{Q},\mathbf{T}} p(\mathbf{Q}, \mathbf{T}|\mathscr{D}) =$

$$\arg\max_{\mathbf{Q},\mathbf{T}} \Big(\sum_{(i,j)\in E} \left\{ \log p(\mathbf{q}_{ij}|\mathbf{Q}, \mathbf{T}) + \log p(\mathbf{t}_{ij}|\mathbf{Q}, \mathbf{T}) \right\} + \sum_i \log p(\mathbf{q}_i) + \sum_i \log p(\mathbf{t}_i) \Big), \tag{8.13}$$

   where $\mathscr{D} \equiv \{\mathbf{q}_{ij}, \mathbf{t}_{ij}\}_{(i,j)\in E}$ denotes the observations, $\mathbf{Q} \equiv \{\mathbf{q}_i\}_{i=1}^n$ and $\mathbf{T} \equiv \{\mathbf{t}_i\}_{i=1}^n$.

2. The full posterior distribution: $p(\mathbf{Q}, \mathbf{T}|\mathscr{D}) \propto p(\mathscr{D}|\mathbf{Q}, \mathbf{T}) \times p(\mathbf{Q}) \times p(\mathbf{T})$.

Both of these problems are very challenging and cannot be directly addressed by standard methods such as gradient descent (problem 1) or standard MCMC methods (problem 2). The difficulty in these problems is mainly originated by the fact that the posterior density is non-log-concave (i.e. the negative log-posterior is non-convex) and any algorithm that aims at solving one of these problems should be able to operate in the particular manifold of this problem, that is $(\mathbb{S}^3)^n \times \mathbb{R}^{3n} \subset \mathbb{R}^{7n}$.

8.2.2 Tempered Geodesic Monte Carlo for PGO

8.2.2.1 Connection between sampling and optimization

In a recent study [169], Liu *et al.*proposed the stochastic gradient geodesic Monte Carlo (SG-GMC) as an extension to [54] and provided a practical posterior sampling algorithm for the problems that are defined on manifolds whose geodesic flows are analytically available. Since our augmented quaternions form such a manifold[1], we

[1]The manifold $(\mathbb{S}^3)^n \times \mathbb{R}^{3n}$ can be expressed as a product of the manifolds $\mathbb{S}^3$ (n times) and $\mathbb{R}^{3n}$. Therefore, its geodesic flow is the combination of the geodesic flows of individual manifolds. Since the geodesic flows in $\mathbb{S}^{d-1}$ and $\mathbb{R}^d$ are analytically available, so is the flow of the product manifold [54].

can use this algorithm for generating (approximate) samples from the posterior distribution, which would address the second problem defined in § 8.2.1.

Recent studies have shown that SG-MCMC techniques are closely related to optimization and they indeed have a strong potential in non-convex problems due to their randomized nature. In particular, it has been recently shown that, a simple variant of SG-MCMC is guaranteed to converge to a point near a local optimum in polynomial time [278, 310] and eventually converge to a point near the global optimum [226], even in non-convex settings. Even though these recent results illustrated the advantages of SG-MCMC in optimization, it is not clear how to develop an SG-MCMC-based optimization algorithm that can operate on manifolds. In this section, we will extend the SG-GMC algorithm in this vein to obtain a *parametric* algorithm, which is able to both sample from the posterior distribution and perform optimization for obtaining the MAP estimates depending on the choice of the practitioner. In other words, the algorithm should be able to address both problems that we defined in § 8.2.1 with theoretical guarantees.

We start by defining a more compact notation that will facilitate the presentation of the algorithm. We define the variable $\mathbf{x} \in \mathscr{X}$, such that $\mathbf{x} \triangleq [\mathbf{q}_1^\top, \dots, \mathbf{q}_n^\top, \mathbf{t}_1^\top, \dots, \mathbf{t}_n^\top]^\top$ and $\mathscr{X} \triangleq (\mathbb{S}^3)^n \times \mathbb{R}^{3n}$. The posterior density of interest then has the form $\pi_{\mathscr{H}}(\mathbf{x}) \triangleq p(\mathbf{x}|\mathscr{D}) \propto \exp(-U(\mathbf{x}))$ with respect to the Hausdorff measure, where U is called the *potential* energy. We define a *smooth embedding* $\xi : \mathbb{R}^{6n} \mapsto \mathscr{X}$ such that $\xi(\tilde{\mathbf{x}}) = \mathbf{x}$. If we consider the embedded posterior density $\pi_\lambda(\tilde{\mathbf{x}}) \triangleq p(\tilde{\mathbf{x}}|\mathscr{D})$ with respect to the Lebesgue measure, then by the area formula (cf. Theorem 1 in [82]), we have the following key property: $\pi_{\mathscr{H}}(\mathbf{x}) = \pi_\lambda(\tilde{\mathbf{x}}) / \sqrt{|\mathbf{G}(\tilde{\mathbf{x}})|}$, where $|\mathbf{G}|$ denotes the determinant of the Riemann metric tensor:

$$[\mathbf{G}(\tilde{\mathbf{x}})]_{i,j} \triangleq \sum_{l=1}^{7n} \frac{\partial x_l}{\partial \tilde{x}_i} \frac{\partial x_l}{\partial \tilde{x}_j} \quad \text{for all} \quad i, j \in \{1, \dots, 6n\}. \tag{8.14}$$

The main idea in our approach is to introduce an *inverse temperature* variable $\beta \in \mathbb{R}_+$ and consider the *tempered* posterior distributions whose density is proportional to $\exp(-\beta U(\mathbf{x}))$. When $\beta = 1$, this density coincides with the original posterior; however, as β goes to infinity, the tempered density concentrates near the global minimum of the potential U [104, 137]. This important property implies that, for large enough β, a random sample that is drawn from the tempered posterior would be close to the global optimum and can therefore be used as a MAP estimate.

8.2.2.2 Construction of the algorithm

We will now construct the proposed algorithm. In particular, we will first extend the continuous-time Markov process proposed in [169] and develop a process whose marginal stationary distribution has a density proportional to $\exp(-\beta U(\mathbf{x}))$ for any given $\beta > 0$. Then we will develop practical algorithms for generating samples from this tempered posterior.

We propose the following stochastic differential equation (SDE) in the Euclidean space by making use of the embedding ξ:

(8.15)

$$d\tilde{\mathbf{x}}_t = \mathbf{G}(\tilde{\mathbf{x}}_t)^{-1}\mathbf{p}_t dt$$

$$d\mathbf{p}_t = -\Big(\nabla_{\tilde{\mathbf{x}}} U_\lambda(\tilde{\mathbf{x}}_t) + \frac{1}{2}\nabla_{\tilde{\mathbf{x}}}\log|\mathbf{G}| + c\mathbf{p}_t + \frac{1}{2}\nabla_{\tilde{\mathbf{x}}}(\mathbf{p}_t^\top \mathbf{G}^{-1}\mathbf{p}_t)dt + \sqrt{(2c/\beta)\mathbf{M}^\top\mathbf{M}}\, dW_t,$$

where $\nabla_{\tilde{\mathbf{x}}} U_\lambda \triangleq -\nabla_{\tilde{\mathbf{x}}}\log\pi_\lambda$, $\mathbf{G}$ and $\mathbf{M}$ are short-hand notations for $\mathbf{G}(\tilde{\mathbf{x}}_t)$ and $[\mathbf{M}(\tilde{\mathbf{x}}_t)]_{ij} \triangleq \partial\mathbf{x}_i/\partial\tilde{\mathbf{x}}_j$, respectively, $\mathbf{p}_t \in \mathbb{R}^{6n}$ is called the *momentum* variable, $c > 0$ is called the *friction*, and W_t denotes the standard Brownian motion in $\mathbb{R}^{6n}$.

We will first analyze the invariant measure of the SDE (8.2.2.2).

Proposition 8.2.1. *Let $\boldsymbol{\varphi}_t = [\tilde{\mathbf{x}}_t, \mathbf{p}_t^\top]^\top \in \mathbb{R}^{12n}$ and $(\boldsymbol{\varphi}_t)_{t\geq 0}$ be a Markov process that is a solution of the SDE* (8.2.2.2)*. Then $(\boldsymbol{\varphi}_t)_{t\geq 0}$ has an invariant measure μ_φ, whose density with respect to the Lebesgue measure is proportional to* $\exp(-\mathscr{E}_\lambda(\boldsymbol{\varphi}))$ *, where $\mathscr{E}_\lambda$ is is defined as follows:*

$$\mathscr{E}_\lambda(\boldsymbol{\varphi}) \triangleq \beta U_\lambda(\tilde{\mathbf{x}}) + \frac{\beta}{2}\log|\mathbf{G}(\tilde{\mathbf{x}})| + \frac{\beta}{2}\mathbf{p}^\top\mathbf{G}(\tilde{\mathbf{x}})^{-1}\mathbf{p}. \tag{8.16}$$

All the proofs are given in the supplementary document. By using the area formula and the definitions of $\mathbf{G}$ and $\mathbf{M}$, one can show that the density of μ_φ can also be written with respect to the Hausdorff measure, as follows: (see § 3.2 in [54] for details) $\mathscr{E}_{\mathscr{H}}(\mathbf{x},\mathbf{v}) \triangleq \beta U_{\mathscr{H}} + \frac{\beta}{2}\mathbf{v}^\top\mathbf{v}$, where $\mathbf{v} = \mathbf{M}(\mathbf{M}^\top\mathbf{M})^{-1}\mathbf{p}$. This result shows that, if we could exactly simulate the SDE (8.2.2.2), then the *marginal* distribution of the sample paths would converge to a measure $\pi_{\mathbf{x}}$ on $\mathscr{X}$ whose density is proportional to $\exp(-\beta U(\mathbf{x}))$. Therefore, for $\beta = 1$ we would be sampling from $\pi_{\mathscr{H}}$ (i.e. we recover SG-GMC), and for large β, we would be sampling near the global optimum of U.

Numerical integration We will now develop an algorithm for simulating (8.2.2.2) in discrete-time. We follow the approach given in [54, 169], where we split (8.2.2.2) into three disjoint parts and solve those parts analytically in an iterative fashion. The split SDE is given as follows:

$$\text{A:}\begin{cases} d\tilde{\mathbf{x}}_t = \mathbf{G}^{-1}\mathbf{p}_t dt \\ d\mathbf{p}_t = \frac{1}{2}\nabla(\mathbf{p}_t^\top \mathbf{G}^{-1}\mathbf{p}_t) dt \end{cases} \quad \text{B:}\begin{cases} d\tilde{\mathbf{x}}_t = 0 \\ d\mathbf{p}_t = -c\mathbf{p}_t dt \end{cases} \quad \text{O:}\begin{cases} d\tilde{\mathbf{x}}_t = 0 \\ d\mathbf{p}_t = -(\nabla U_\lambda(\tilde{\mathbf{x}}_t) + \frac{1}{2}\nabla \log|\mathbf{G}|) dt \\ \qquad\qquad + \sqrt{\frac{2c}{\beta}\mathbf{M}^\top \mathbf{M}}\, dW_t. \end{cases}$$

The nice property of these (stochastic) differential equations is that, each of them can be analytically simulated directly on the manifold $\mathscr{X}$, by using the identity $\mathbf{x} = \xi(\tilde{\mathbf{x}})$ and the definitions of $\mathbf{G}$, $\mathbf{M}$, and $\mathbf{v}$. In practice, one first needs to determine a sequence for the A, B, O steps, set a step-size h for integration along the time-axis t, and solve those steps one by one in an iterative fashion [62, 162]. In our applications, we have emprically observed that the sequence BOA provides better results among several other combinations, including the ABOBA scheme that was used in [169].

Once the gradients with respect to the latent variables are computed, i.e.:

$$\nabla_{\mathbf{x}} U(\mathbf{x}) \equiv \{\nabla_{\mathbf{q}_1} U(\mathbf{x}), \dots \nabla_{\mathbf{q}_n} U(\mathbf{x}), \nabla_{\mathbf{t}_1} U(\mathbf{x}), \dots \nabla_{\mathbf{t}_n} U(\mathbf{x})\}, \tag{8.17}$$

we can update each of the variables $\mathbf{q}_1, \dots, \mathbf{q}_n, \mathbf{t}_1, \dots, \mathbf{t}_n$ independently from each other, meaning that, the split integration steps, A, B, O can be applied to each of these variables independently. The operations A, B, O will differ depending on the manifold of the particular variable, therefore we will define these operations both on $\mathbb{S}^3$ and $\mathbb{R}^3$ for the two sets of variables $\{\mathbf{q}_i\}_{i=1}^n$ and $\{\mathbf{t}_i\}_{i=1}^n$, respectively. As we split $\mathbf{x}$ into $\{\mathbf{q}_i\}_{i=1}^n$ and $\{\mathbf{t}_i\}_{i=1}^n$, we similarly split the variable $\mathbf{v}$ into $\{\mathbf{v}_i^{\mathbf{q}}\}_{i=1}^n$ and $\{\mathbf{v}_i^{\mathbf{t}}\}_{i=1}^n$ in order to facilitate the presentation. Below we provide the necessary update equations.

Update equations for the rotation components Set a step-size h. For each $\{\mathbf{q}_i, \mathbf{v}_i^{\mathbf{q}}\}$ pairs, the operations A, B, O have the following analytical form:

Step A**:** Set $\alpha = \|\mathbf{v}_i^{\mathbf{q}}\|$, $\mathbf{q}' \leftarrow \mathbf{q}_i$ and $\mathbf{v}' \leftarrow \mathbf{v}_i^{\mathbf{q}}$.

$$\begin{aligned} \mathbf{q}_i &\leftarrow \mathbf{q}' \cos(\alpha h) + (\mathbf{v}'/\alpha)\sin(\alpha h) \\ \mathbf{v}_i^{\mathbf{q}} &\leftarrow -\alpha \mathbf{q}' \sin(\alpha h) + \mathbf{v}' \cos(\alpha h) \end{aligned} \tag{8.18}$$

Step B: $\mathbf{v}_i^{\mathbf{q}} \leftarrow \exp(-ch)\mathbf{v}_i^{\mathbf{q}}$

Step O: Set $\mathbf{v}' \leftarrow \mathbf{v}_i^{\mathbf{q}}$ and $\mathbf{g} \leftarrow \nabla_{\mathbf{q}_i} U(\mathbf{x})$

$$\mathbf{v}_i^{\mathbf{q}} \leftarrow \mathbf{v}' + P(\mathbf{q}_i)\big(-h\mathbf{g} + \sqrt{2c/\beta}\,\mathbf{z}_i^{\mathbf{q}}\big), \tag{8.19}$$

where $P(\mathbf{q}) = (\mathbf{I} - \mathbf{q}\mathbf{q}^\top)$ denotes the projector and $\mathbf{z}_i^{\mathbf{q}}$ denotes a standard Gaussian random variable on $\mathbb{R}^4$.

Update equations for the translation components Set a step-size h. For each $\{\mathbf{t}_i, \mathbf{v}_i^{\mathbf{t}}\}$ pairs, the operations A, B, O have the following analytical form:

Step A: $\mathbf{t}_i \leftarrow \mathbf{t}_i + h\mathbf{v}_i^{\mathbf{t}}$

Step B: $\mathbf{v}_i^{\mathbf{t}} \leftarrow \exp(-ch)\mathbf{v}_i^{\mathbf{t}}$

Step O: Set $\mathbf{v}' \leftarrow \mathbf{v}_i^{\mathbf{t}}$ and $\mathbf{g} \leftarrow \nabla_{\mathbf{t}_i} U(\mathbf{x})$

$$\mathbf{v}_i^{\mathbf{t}} \leftarrow \mathbf{v}' + \big(-h\mathbf{g} + \sqrt{2c/\beta}\,\mathbf{z}_i^{\mathbf{t}}\big), \tag{8.20}$$

where $\mathbf{z}_i^{\mathbf{t}}$ denotes a standard Gaussian random variable on $\mathbb{R}^3$. We illustrate the overall algorithm in Algorithm 7

Algorithm 7: TG-MCMC

input : $\mathbf{x}_0 = \{\mathbf{q}_1, \ldots, \mathbf{q}_n, \mathbf{t}_1, \ldots, \mathbf{t}_n\}$, $\mathbf{v} = \{\mathbf{v}_1^{\mathbf{q}}, \ldots, \mathbf{v}_n^{\mathbf{q}}, \mathbf{v}_1^{\mathbf{t}}, \ldots, \mathbf{v}_n^{\mathbf{t}}\}$, β, c , h

for $i = 1, \ldots, N$ **do**

  Compute the gradient $\nabla_{\mathbf{x}} U(\mathbf{x}_i)$

  `// Update the rotation components`

  for $j = 1, \ldots, n$ **do**

    Run the B, O, A steps (in this order) on $\mathbf{q}_j, \mathbf{v}_j^{\mathbf{q}}$

  `// Update the translation components`

  for $j = 1, \ldots, n$ **do**

    Run the B, O, A steps (in this order) on $\mathbf{t}_j, \mathbf{v}_j^{\mathbf{t}}$

8.2.2.3 Theoretical analysis

We will now provide non-asymptotic results for the proposed algorithm. Let us denote the output of the algorithm $\{\mathbf{x}_k\}_{k=1}^N$, where k denotes the iterations and N denotes the number of iterations. In the MAP estimation problem, we are interested in finding $\mathbf{x}^\star \triangleq$

$\arg\min_{\mathbf{x}} U(\mathbf{x})$, whereas for full Bayesian inference, we are interested in approximating posterior expectations with finite sample averages, i.e. $\bar{\phi} \triangleq \int_{\mathscr{X}} \phi(\mathbf{x})\pi_{\mathscr{H}}(\mathbf{x})\, d\mathbf{x} \approx \hat{\phi} \triangleq (1/N)\sum_{k=1}^{N} \phi(\mathbf{x}_k)$, where ϕ is a test function.

As briefly discussed in [169], the convergence behavior of the SG-GMC algorithm can be directly analyzed within the theoretical framework presented in [62]. In a nutshell, the theory in [62] suggests that, with the BOA integration scheme, the bias $|\mathbb{E}\hat{\phi} - \phi|$ is of order $\mathscr{O}(N^{-1/2})$.

In this study, we focus on the MAP estimation problem and analyze the *ergodic* error $\mathbb{E}[\hat{U}_N - U^\star]$, where $\hat{U}_N \triangleq (1/N)\sum_{k=1}^{N} U(\mathbf{x}_k)$ and $U^\star \triangleq U(\mathbf{x}^\star)$. This error resembles the bias where the test function ϕ is chosen as the potential U; however, on the contrary, it directly relates the sample average to the global optimum. Similar ergodic error notions have already been considered in non-convex optimization [61, 167].

We present our main result in the Thm. 8.1, but before, we lay down our assumptions that allow us to formulate the theorem:

H1. The gradient of the potential is Lipschitz continuous, i.e. there exists $L < \infty$, such that $\|\nabla_{\mathbf{x}} U(\mathbf{x}) - \nabla_{\mathbf{x}} U(\mathbf{x}')\| \leq L\mathrm{d}_{\mathscr{X}}(\mathbf{x},\mathbf{x}')$, $\forall \mathbf{x},\mathbf{x}' \in \mathscr{X}$, where $\mathrm{d}_{\mathscr{X}}$ denotes the geodesic distance on $\mathscr{X}$.

H2. The second-order moments of $\pi_{\mathbf{x}}$ are bounded and satisfies the following inequality: $\int_{\mathbb{R}^d} \|\mathbf{x}\|^2 \pi_{\mathbf{x}}(d\mathbf{x}) \leq \frac{C}{\beta}$, for some $C > 0$.

H3. Let ψ be a functional that is the unique solution of a Poisson equation that is defined as follows:

$$\mathscr{L}_n \psi(\boldsymbol{\varphi}_n) = U(\mathbf{x}_n) - \bar{U}_\beta, \tag{8.21}$$

where $\boldsymbol{\varphi}_n = [\tilde{\mathbf{x}}_n^\top, \mathbf{p}_n^\top]^\top$, $\mathscr{L}_n$ is the generator of (8.2.2.2) at $t = nh$ (see [62] for the definition). The functional ψ and its up to third-order derivatives $\mathscr{D}^k\psi$ are bounded by a function $V(\boldsymbol{\varphi})$, such that $\|\mathscr{D}^k\psi\| \leq C_k V^{r_k}$ for $k = 0,1,2,3$ and $C_k, r_k > 0$. Furthermore, $\sup_n \mathbb{E}V^r(\mathbf{x}_n) < \infty$ and V is smooth such that $\sup_{s\in(0,1)} V^r(s\boldsymbol{\varphi} + (1-s)\boldsymbol{\varphi}') \leq C(V^r(\boldsymbol{\varphi}) + V^r(\boldsymbol{\varphi}'))$ for all $\boldsymbol{\varphi}, \boldsymbol{\varphi}' \in \mathbb{R}^{12n}$, $r \leq \max 2r_k$, and $C > 0$.

Theorem 8.1. *Assume that the conditions given in 1 hold. If the iterates are obtained by using the* BOA *the scheme, then the following bound holds:*

$$\left|\mathbb{E}\hat{U}_N - U^\star\right| = \mathscr{O}\big(\beta/(Nh) + h/\beta + 1/\beta\big), \tag{8.22}$$

for the domain $\mathscr{X} = (\mathbb{S}^3)^n \times \mathbb{R}^{3n}$.

Sketch of the proof. The proof is based on decomposing the error into two terms: $\mathbb{E}[\hat{U}_N - U^\star] = \mathscr{A}_1 + \mathscr{A}_2$, where $\mathscr{A}_1 \triangleq \mathbb{E}[\hat{U}_N - \bar{U}_\beta]$ and $\mathscr{A}_2 \triangleq [\bar{U}_\beta - U^\star] \geq 0$, and $\bar{U}_\beta \triangleq \int_{\mathscr{X}} U(\mathbf{x})\pi_{\mathbf{x}}(d\mathbf{x})$. The term $\mathscr{A}_1$ is the bias term, which we can bounded by using existing results. The rest of the proof deals with bounding $\mathscr{A}_2$, where we incorporate ideas from [226]. The full proof resides in the supplementary. ■

Theorem 8.1 shows that the proposed algorithm will eventually provide samples that are close to the global optimizer $\mathbf{x}^\star$ even when U is non-convex. In other words, the *tempered posterior* will concentrate around the global optimum as we increase β. Hence, for large enough β, the multi-modal structure of the posterior disappears and there will be a single mode of the distribution. This will imply that if $U(\mathbf{x})$ is close to $U^\star$, then $\mathbf{x}$ should be close to $\mathbf{x}^\star$ under the large β regime. Fig. 8.3 illustrates this phenomenon on a simple 2-component Gaussian mixture: when $\beta = 1$ both modes are visible, but when $\beta = 20$ the mode on the right vanishes and the distribution concentrates around the global mode. Our result is stronger than the guarantees for the existing convex optimization algorithms on manifolds [174, 309], and is mainly due to the stochasticity of the algorithm that is introduced by the Brownian motion. However, despite this nice theoretical property, in practice our algorithm will still be affected by the *meta-stability phenomenon*, where it will converge near a local minimum and stay there for an exponential amount of time.

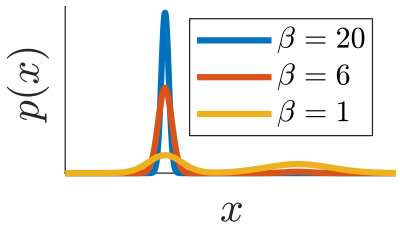


Figure 8.3: Effect of β on the cost function.

We also note that the case $\beta \to \infty$ renders the SDE degenerate and hence, cannot be analyzed by using standard MCMC tools. Thus, our bound, as well as the argumentation of the numerous recent MCMC-based optimization algorithms [32,33,34], ceases to hold. However, this does not limit the applicability of our theoretical results due to the following: 1) Primarily, as long as β is set to a large yet finite value, the algorithm will indeed perform optimization (see also [32,33,34]) and converge near a MAP solution. *β need not be infinite for this.* 2) The upper-bound on β depends on a constant C that appears on Assumption $H2$, which states that the second-order moments of the (tempered) posterior obey the

inequality $\int_{\mathbb{R}^d} \|\mathbf{x}\|^2 \pi_{\mathbf{x}}(d\mathbf{x}) \leq \frac{C}{\beta}$. Here, the constant C can be arbitrarily large. In fact, we expect C to be very large in our applications, since the second-order posterior moments would be very large in the high-dimensional scheme we consider. This also allows us to set β to a large value as confirmed by our experiments. 3) Due to the meta-stability phenomenon (mentioned in line 251), the algorithm already performs similarly either for large β or for $\beta \to \infty$. For all those reasons, in § 8.3.2, we had chosen to phrase that $\beta \to \infty$ to increase clarity.

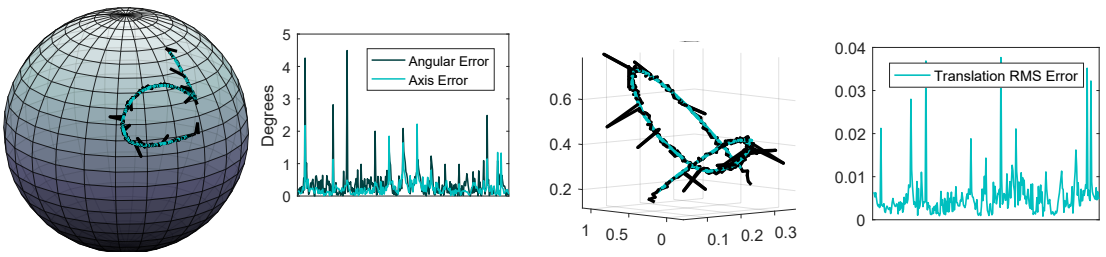
We note that our proof covers only the case where $\mathscr{X} = (\mathbb{S}^3)^n \times \mathbb{R}^{3n}$; however, we believe that it can be easily extended to more general setting. We also note that the gradient computations in our algorithm can be replaced with stochastic gradients in the case of large-scale applications where the number of data points can be prohibitively large, so that computing the gradients at each iteration becomes practically infeasible. The same theoretical results hold as long as the stochastic gradients are unbiased.

8.3 Experiments

8.3.1 Pose Filtering

In the following, the pose filtering methods presented in § 8.1 are evaluated and compared to other approaches in two different scenarios. A first synthetic experiment analyzes the ability of the smoothing algorithms to recover a noisy pose series with outliers while the second evaluation is performed on a real dataset of natural hand movement in a collaborative medical robotic environment where tracking accuracy and robustness to pose outliers is crucial.

Synthetic tests Our first test evaluates the robustness and accuracy of the tangent space regressors. In order to evaluate these properties we generate a synthetic dataset from a ground truth rigid body movement.
A set of five points $\mathbf{v}_i \in \mathbb{R}^3$ together with five values $\theta_i \in [0, 2\pi]$ are chosen as query points representing the rotation axis and angle of the rotations $\mathbf{R}_i$. Five points $\mathbf{t}_i \in [0,1]^3$ represent the translational component of the poses. A cubic spline interpolates both the axes and angles and with (2.14) and (2.28) we get our pose representations in $\mathbb{H}_1 \times \mathbb{R}^3$ and $\mathbb{D}\mathbb{H}_1$. As the space $\mathbb{R}^3$ is already Euclidean we can perform a pose filtering



(a) Effect on Rotations.

(b) Effect on Rotations.

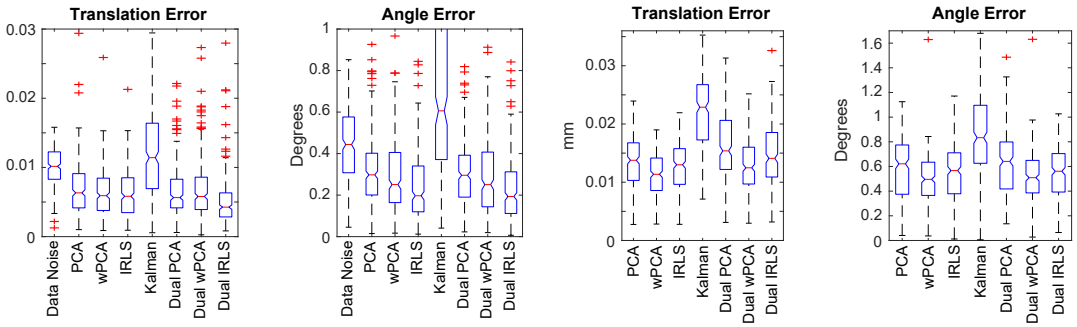
Figure 8.4: Dual IRLS applied to synthetic rigid body movement (black). **(a)** Rotational effect: the intersections of the rotation axis with $\S^2$ are illustrated for both the noisy input (black) and the filtered pose (turqoise), along with the angular and axis errors of the rotation. **(b)** Translational effect: on the left, the smooth trajectory is visualized, while the rightmost figure shows the RMS error on the translation component.

for the 6-DoF pose both in $\mathbb{DH}_1$ and $\mathbb{H}_1 \times \mathbb{R}^3$. For the latter, all methods are applied twice on both spaces independently.

To evaluate the performance of the local regression, we sample the ground truth pose series densely and apply additional uniform noise in range $[-0.02, 0.02]$ to the angle and the axis of rotation as well as to the translation. On top, random outliers for 5 % of the data points are generated with an additional noise of $\sigma = 0.2$.

Then we run PCA, wPCA, IRLS, Dual PCA, Dual wPCA, Dual IRLS as well as a Linear Kalman Filter. We use a window size of 19 and the Kalman implementation [289] of MATLAB [187] with a covariance tuple of $[0.5, 2]$ for the rotation and $[0.2, 1]$ for the translation process noise and measurement noise covariance. The resulting pose set is illustrated together with the results for the Dual IRLS method in Fig. 8.4.

An error quantification for the different methods is given in Fig. 8.5(a). For the Kalman filter, tradeoff values have been chosen which are still able to recover the pose without over-smoothing. However, the method only evaluates the past points and thus information of half the window size for future poses is not included which explains the performance difference. The direct local PCA methods perform equally well in the same error range while the weighting gain in the separate treatment is slightly better with 0.010 ± 0.020 in translation and $0.37 \pm 0.55°$ rotation. The outlier aware IRLS method performs best for the angle while the improvement for the translation is with $2.1 \cdot 10^{-3}$ only minimal for the non-dual quaternions. It can be clearly seen that the treatment of outliers in the dual space $\mathbb{DH}_1$ helps to increase the accuracy



(a) (b)

Figure 8.5: **(a)** Performance tests of different pose filtering methods on synthetic data with the given noise shown in Data Noise. The illustrated Angle Error is the angular difference of the rotation axis to the ground truth. **(b)** Tracking stream refinements for natural hand movement.

to $4.3 \cdot 10^{-3}$ and 0.26° in the median. This can be intuitively understood through the fact that the local neighborhood of the linear regression line in $\mathbb{DH}_1$ is much more restrictive than the joint neighborhoods in $\mathbb{H}_1 \times \mathbb{R}^3$ where the effect of an outlier in the separate parameter spaces is higher.

It is also worth mentioning that the only parameter of the presented methods is the window size which directly reflects the movement speed of the displacements while the parameter adjustments for a filter method such as Kalman are more elaborate. For a visual comparison of the different methods on a synthetic pose stream with noise, please be referred to the supplementary video.

Real data: Tracking stream refinement We compare our methods on the dataset of Busam *et al.* [52] where a robotic arm runs in gravity compensation mode with zero stiffness and a human operator performs a natural hand movement manipulating its end effector. The robot is tracked via a marker based stereo vision system running the tracking algorithm [51] and calibrated such that the forward kinematics of the industrial robotic manipulator provide ground truth poses with a precision of 0.05 mm. The 30 Hz pose stream is fed into our filter pipelines and compared to the absolute poses of the dataset. Fig. 8.5(b) illustrates the results where we use the same naming and parameters as above. It is noteworthy that in this scenario, the wPCA methods

perform best with $11.2 \pm 3.9 \mu m$ (median $11.4 \mu m$) and $0.7 \pm 1.4°$ (median $0.5°$) for the non-dual one while the IRLS methods give only mediocre results between PCA and wPCA both for quaternions ($13.0 \mu m$, $0.6°$ median) and dual quaternions ($14.1 \mu m$, $0.6°$) median. The Kalman filter again gives acceptable results for which heuristical parameter fine-tuning did not show any significant improvements.
The advantage of the dual space robustification - which can yield a significant improvement in the case of outliers (see above) - is not applicable as there are only few outliers in the already quite accurate optical tracking data. The IRLS methods suffer from this problem as the weights for equally important data points are reduced. This results in case of reliable pose data in the fact that the separate treatment of translation and rotation is preferable to the non-dual regressors perform better.

8.3.2 Evaluating Pose Graph Optimization

We run a series of synthetic and real experiments to empirically show the characteristics of our algorithm. In a sequel of evaluations, we will be benchmarking our TG-MCMC against the state of the art methods including subsets of: convex programming of Ozyesil *et al.* [320], Lie algebraic method of Govindu [112], dual quaternions linearization of Torsello *et al.* [269], direct EIG-SE3 method of Arrigoni [13] and R-GODEC [15]. We also include two baseline methods: 1. propagating the pose information along one possible minimum spanning tree (minspan), 2. the chordal averaging [121] (chordal). We used the provided codes whenever possible.

Hyper-parameter selection Throughout all the experiments we set $c \leftarrow 1000$ and during optimization $\beta \leftarrow \infty$. The variance of the Bingham distribution is adjusted in range $\lambda \in [350, 900]$. Likewise, variance of the Gaussian lies in $\sigma^2 \in [0.01, 0.1]$. Typically, the exact value is picked emprically. Note that certain level of noise can also be compensated by the step size, as variance and step size are multiplicative factors. To show that the choice is not critical, in Fig. 8.6, we plot λ, our most sensitive parameter, against the error

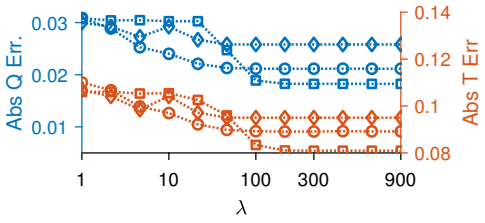


Figure 8.6: Effect of λ on rotational (Q) and translational (T) errors.

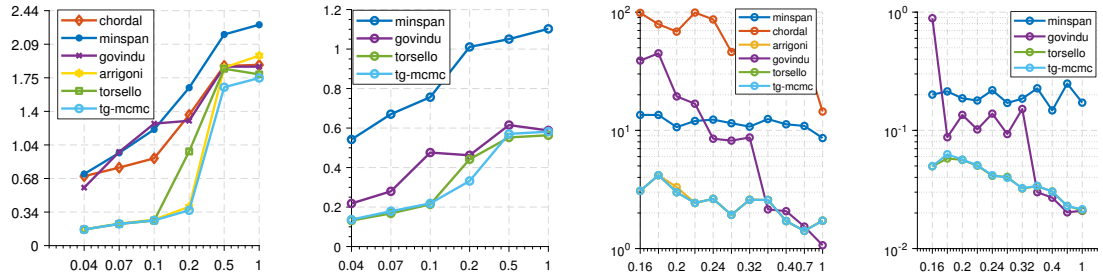


Figure 8.7: Synthetic Evaluations. **(a)** Mean Riemannian (rotational) error vs noise variance. **(b)** Mean Euclidean (translational) error vs noise variance. **(c)** Riemannian error vs E% for $N = 50$. **(d)** Euclidean error vs E% for $N = 50$. E% refers to graph completeness and N to the node count.

attained at convergence for different datasets, including synthetic and real. It is seen that, for a variety of choices where $\lambda > 100$, the solution can safely be found. The step size h varies between $0.001-0.008$ depending on (λ, σ^2) and TG-MCMC runs until convergence. Thus, the number of integration steps varies, typically in range $[350, 800]$. We initialize our algorithm randomly and Govindu [112, 114] from a minimum spanning tree (*minspan*).

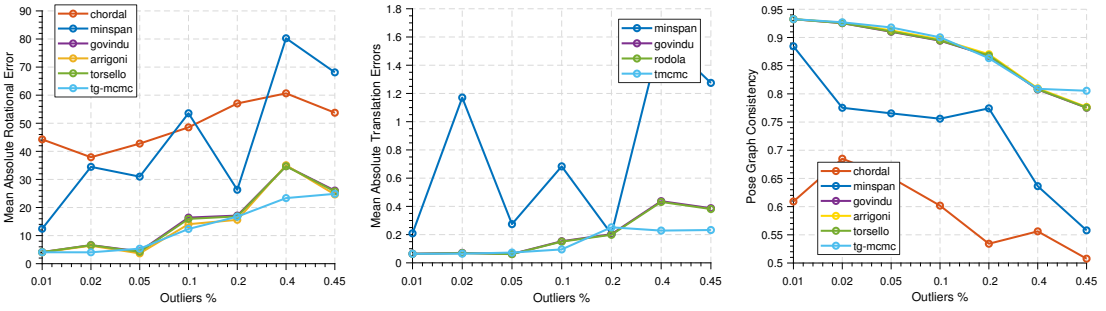
Graph consistency As an intuitive measure of quantifying how well the estimated parameters agree to the input data, we propose *graph consistency* and define it as:

$$g_c = 1 - \frac{1}{\pi |E|} \sum_{(i,j) \in E} 2 \arccos(\mathbf{q}_{ij} (\mathbf{q}_i \bar{\mathbf{q}}_j)) \tag{8.23}$$

In other words, g_c measures how well the relative poses computed from absolute estimates align with the ones given in the data. $g_c = 1$ for the perfect ground truth and $g_c \to 0$ when all estimates are off by π. This measure is easier to interpret than, say, average rotational distance, that is always unit bound.

8.3.2.1 Synthetic Evaluations

We first synthesize random problems with ground truth (GT) to perform controlled evaluations. To do so, we first generate a fully connected graph, and randomly drop e% of the edges, ensuring that the graph is connected. For non-connected graphs, we take



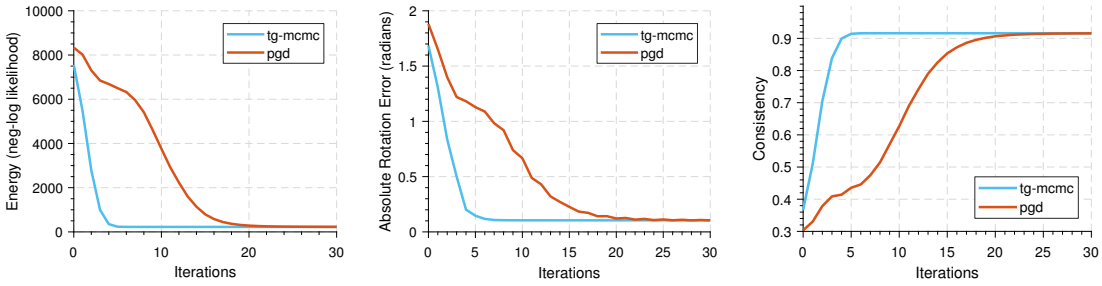
(a) (b) (c)

Figure 8.8: Robustness to outliers. With respect to the outlier percentage, we plot: **(a)** deviations of rotations from ground truth (mean error) **(b)** deviations of translation from ground truth (mean error) **(c)** graph consistency (for definition, see Eq. 8.23).

the largest connected component. Then, we sample n rotations from the Bingham and n translations from the Gaussian distribution as determined by the prior model. This forms the GT absolute poses, from which we further sample the noisy relative poses, i.e., another Bingham and Gaussian distribution pair. On these problems, we run a series of tests including monitoring the gradient steps, noise robustness, tolerance to graph completeness (sparsity) and fidelity w.r.t. ground truth. For each test, we distort the graph for the entity we test, i.e. add noise on nodes if we test the noise resilience.

Robustness to noise The rotational errors are evaluated by the true Riemannian distance, $\|\log(\mathbf{R}^T\hat{\mathbf{R}})\|$, the translations by Euclidean $\|\mathbf{t}-\hat{\mathbf{t}}\|$. Fig. 8.7 plots our findings. It is noticeable that our accuracy is always on par with or better than the state of the art for moderate problems. In presence of significant degradation in the data, such as increased noise (Figures 8.7(a), 8.7(b)) or sparsified graph structure leading to missing data (Figures 8.7(c), 8.7(d)), our method shows clear advantage in both rotational and translational components of the error. This is thanks to our probabilistic formulation and theoretically grounded inference scheme.

Outlier resilience Even though TG-MCMC has no explicit treatment of outliers, it is still of interest to observe the robustness to outliers. We do that synthetically, by following a similar experimentation setup. This time, we increase the outlier ratio in the pose graph by excessively corrupting some of the relative transformation matrices



(a) (b) (c)

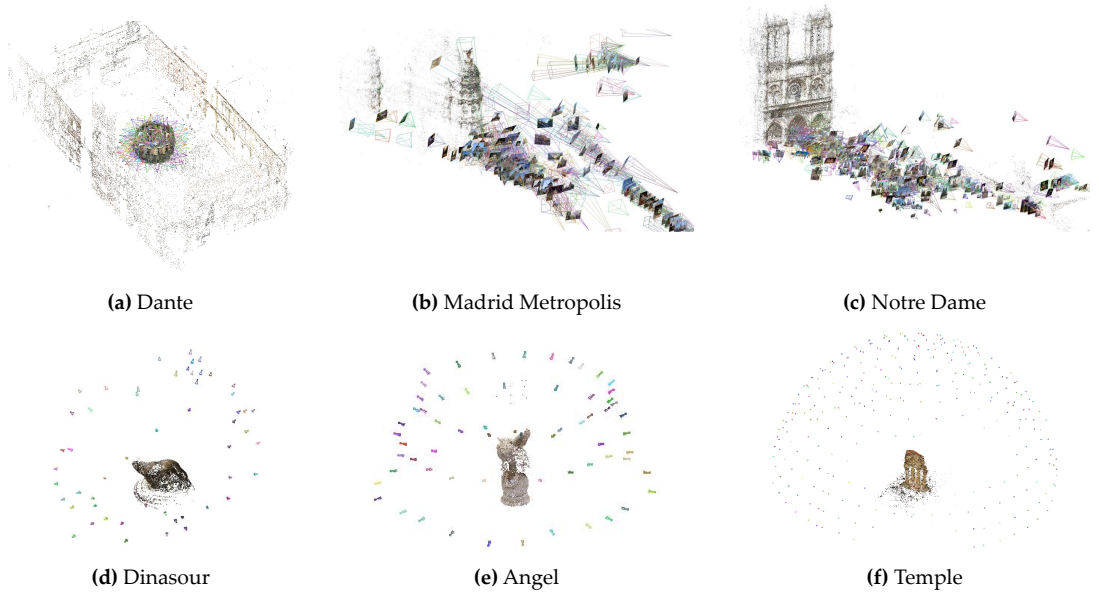
Figure 8.9: Synthetic evaluations against projected gradient descent (PGD). **(a)** Negative log likelihood vs iterations. **(b)** Absolute rotation error of the estimates w.r.t. ground truth vs iterations. **(c)** Consistency (for definition, see Eq. 8.23) vs iterations.

by composing it with random rotations in the range $[60°, 80°]$ around random axes, and random translations between $[0, 1]$. We then run TG-MCMC, as well as the other algorithms under consideration. Our results are depicted in Fig. 8.8. Many state-of-the-art methods that lack outlier handling are similar in performance. However, advantage of TG-MCMC is more apparent as the outlier percentage increases.

Projected gradient descent Next, we compare our method against projected gradient descent (PGD) algorithm, that is heavily used when one avoids the manifold operations of quaternions. This amounts to solving our MAP estimation using a standard first order method and projecting the intermediary solutions back onto the manifold. Using compatible step sizes, Fig. 8.9 plots multiple quantities as iterates progress. It is clearly visible that walking on the manifold is advantageous both in finding quicker solutions (a,b) and in reducing the energy of the cost (c).

8.3.2.2 Results in Real Data

We now evaluate our framework by running SFM on the EPFL Benchmark [254], that provide 8 to 30 images per dataset, along with ground-truth camera transformations. Moreover we utilize the standard SFM datasets [294]. To have a better idea of the nature of these datasets, it is worthwhile to visualize the camera locations as well as the 3D reconstruction following a full bundle adjustment, that optimizes both 3D points (structure) and 3D poses (motion). In Fig. 8.10, we report 6 such visualizations



(a) Dante **(b)** Madrid Metropolis **(c)** Notre Dame

(d) Dinasour **(e)** Angel **(f)** Temple

Figure 8.10: Results of the full bundle adjustment (structure + camera poses) on several datasets.

on 3 outdoor, large scenes, as well as 3 object-scanning scenarios. These plots are not the outcome of our approach, but meant as a reference for the datasets we deal with.

To quantify the accuracy, we match the ambiguous scale of the estimated translations to that of the ground truth, similar to [13]. The mean rotation and translation errors (MRE, MTE) are depicted in Tab. 8.1. Notice that when rotations and translations are combined, our optimization results in superior minimum both for translations and orientations, not to mention the uncertainty information computed as a by-product.

Table 8.1: Evaluations on EPFL Benchmark [254].

	Ozyesil et al		R-GODEC		Govindu		Torsello		EIG-SE(3)		**TG-MCMC**	
	MRE	MTE	MRE	MTE	MRE	MTE	MRE	MTE	MRE	MTE	MRE	MTE
HJ-P8	0.060	0.007	0.040	0.009	0.106	0.015	0.106	0.015	0.040	0.004	0.106	0.015
HJ-P25	0.140	0.065	0.130	0.038	0.081	0.020	0.081	0.020	0.070	0.010	0.081	0.020
Fountain	0.030	0.004	0.030	0.006	0.071	0.004	0.071	0.004	0.030	0.004	0.071	0.004
Entry-P10	0.560	0.203	0.440	0.433	0.101	0.035	0.101	0.035	0.040	0.009	0.090	0.035
Castle-P19	3.690	1.769	1.570	1.493	0.393	0.147	0.393	0.147	1.480	0.709	0.393	0.148
Castle-P30	1.970	1.393	0.780	1.123	0.631	0.323	0.629	0.321	0.530	0.212	0.622	0.285
Average	1.075	0.574	0.498	0.517	0.230	0.091	0.230	0.090	0.365	0.158	**0.227**	**0.085**

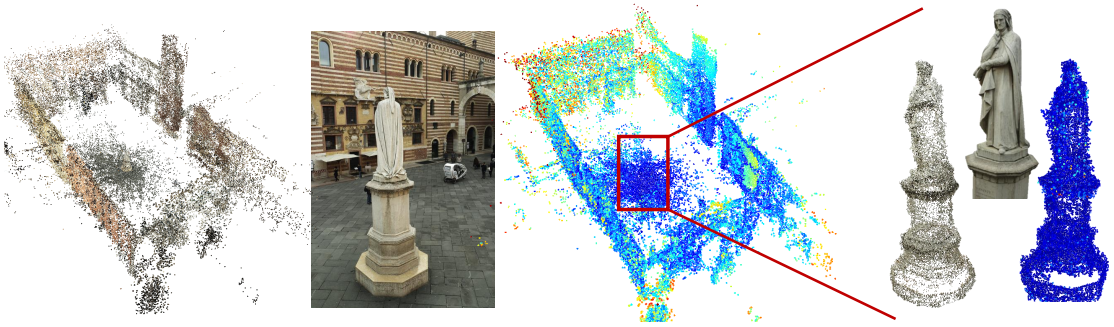
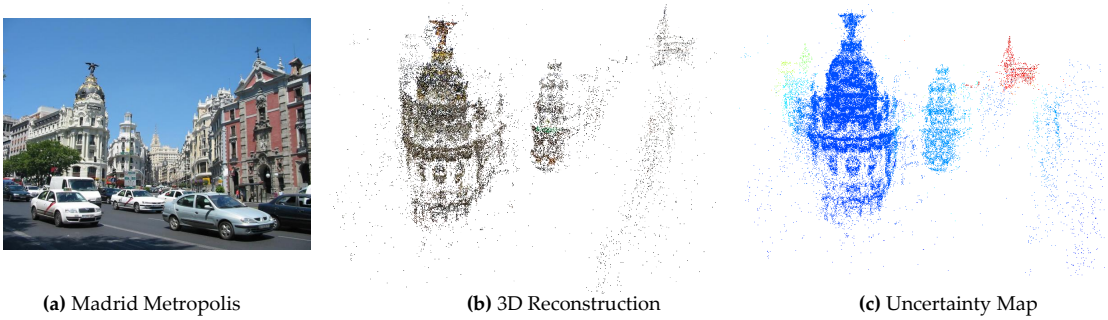


Figure 8.11: Uncertainty estimation in the Dante Square. From left to right: the colored reconstruction (bundle adjustment used in 3D structure only), a sample image from the dataset, reconstructed points colored w.r.t. uncertainty value, a close-up to the center of the square, Dante statue.

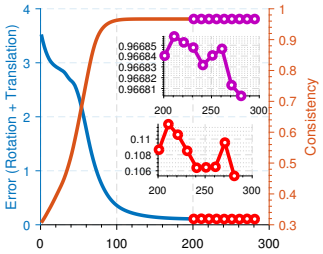


(a) Madrid Metropolis **(b)** 3D Reconstruction **(c)** Uncertainty Map

Figure 8.12: Reconstruction of Madrid Metropolis. Our uncertainty map can reveal the distant structures because the 3D triangulation quality decreases with the distance. Samples produced by TG-MCMC can successfully explain these variations.

While many methods can perform similarly on easy sets, a clear advantage is visible on Castle sequences where severe noise and missing connections are present. There, for instance, EIG-SE(3) also fails to find a good closed form solution.

Uncertainty estimation in real data Estimating uncertainties within the same framework is a differentiating aspect of our method. Hence, we now qualitatively demonstrate the uncertainty estimation capabilities arising on various SFM problems and datasets [254, 291] including 3D Flow showcase - `https://www.3dflow.net`. To do so, as shown in the figure on the right, we first run our opti-



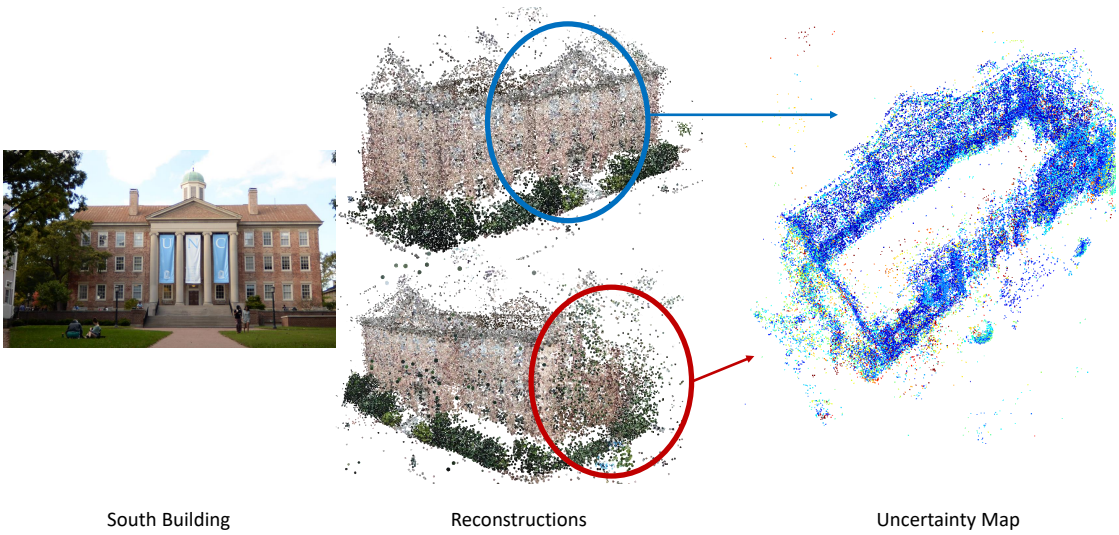


Figure 8.13: Reconstruction of South Building of UNC. Notice that hard-to-reconstruct structure such as vegetation is also marked to be uncertain by our algorithm, whereas rigid structures such as building façades enjoy high certainty.

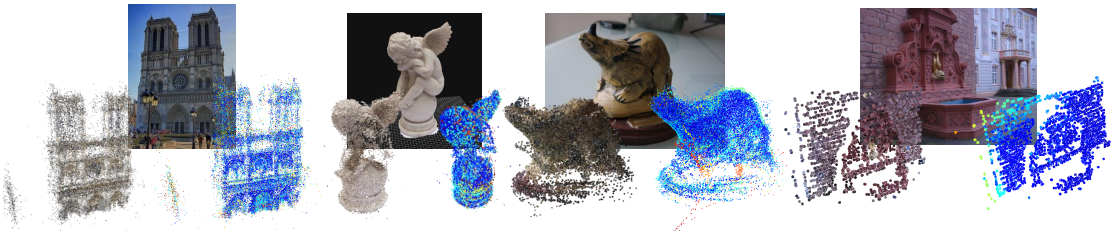


Figure 8.14: Visualization of uncertainty in Notre Dame, Angel, Dinosaur and Fountain datasets.

mizer setting $\beta \leftarrow \infty$ for > 400 iterations. After that point, depending on the dataset, we set β to a finite value (~ 1000), allowing the sampling of posterior for 40 times. For each sample, that is a solution of the problem in Eq. 8.7, we perform a 3D reconstruction, similar to [59]: We first estimate 2D keypoints and relative rotations by running 1) VSFM [294] 2) two-frame bundle adjustment [4] (BA) on image pairs, resulting in pairwise poses, as well as a rough two-view 3D structure. We run our method on these relative poses, computing the absolute estimates. Fixing the estimated poses, a second BA then optimizes for the optimal 3D structure. At the end, we obtain 40 3D scenes per dataset. For each point, we record the mean and variance across different reconstructions, transferring the uncertainty estimation to the 3D cloud of points.

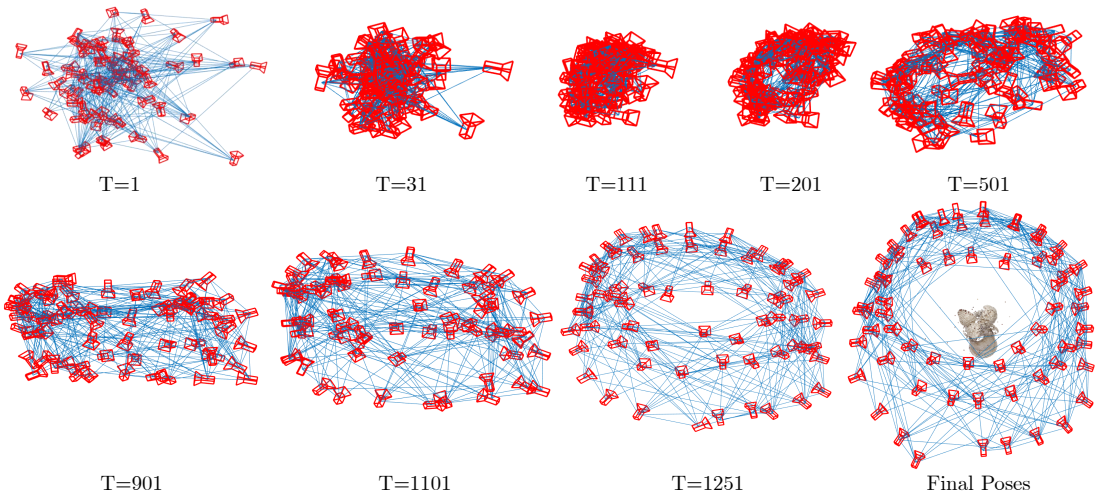
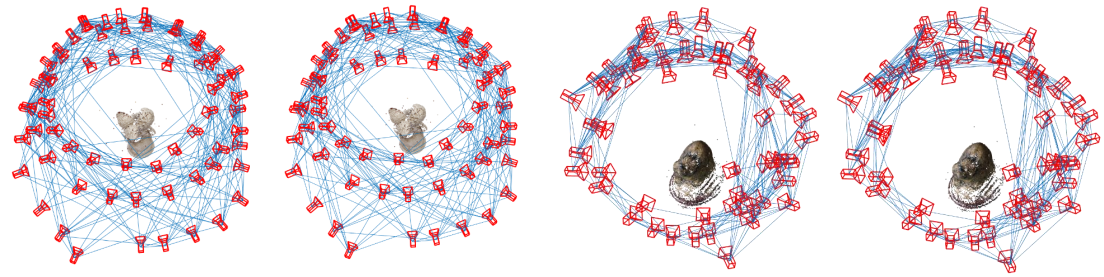


Figure 8.15: Evolution of the graph structure on the Angel object. T denotes iterations. TG-MCMC is initialized by random poses drawn from a Bingham distribution.

In figures 8.11, 8.12, 8.13 and 8.14, we colorize each point by mapping the uncertainty value to RGB space using a jet-colormap (blue lower, red higher), with a scale proportional to the diameter of reconstruction. Fig. 8.12 illustrates the uncertainty mapping on the Madrid Metropolis reconstruction and Fig. 8.13 on the South Building dataset [242, 243]. In the former, distant content, that is intrinsically less accurate to triangulate, appears less certain than the structure nearby. This overlaps well with the findings of stereo vision where baseline-to-distance ratio determines the triangulation accuracy. In the latter, though, we see that hard to match content such as vegetation has more uncertainty. This is also natural, because such image regions render the feature matching difficult. Finally, in Fig. 8.14, we provide four more maps on Notre Dame, Angel, Dinosaur, and Fountain objects. In those, due to the noise and small size, variation is less apparent. Nevertheless, on Angel dataset, our algorithm overall manages to spot the noisy points and mark them with higher uncertainty. On the Fountain, the structure close to the borders of the scene are photographed from a fewer number of cameras, which is what TG-MCMC has discovered and on Notre Dame, and Dinosaur, outliers are picked to be uncertain.

Graph evolution To shade light upon the inner workings of TG-MCMC, we now visualize the evolution of the pose graph as iterations/time proceed(s) on the exemplary



(i) Ground truth pose graph **(ii)** Computed pose graph **(i)** Ground truth pose graph **(ii)** Computed pose graph

(a) Angel **(b)** Dinasour

Figure 8.16: Comparing the resulting pose graph with the ground truth for Angel and Dino objects.

Angel object - see Fig. 8.15. Notice, our algorithm can start from a random initialization and achieve results that are very close to the ground truth. In fact, we also show comparisons of the obtained pose graph against the ground truth poses in Fig. 8.16. Our low numbers in quantitative error well transfers to qualitative evaluations.

Computational cost We now provide, on the real setting of EPFL benchmark, a runtime analysis, evaluating the overhead of our algorithm. Our results are shown in Tab. 8.2. It is common to all PGO initialization algorithms to outperform bundle adjustment (BA) in terms of speed. While we are not the fastest, our runtimes are comparable to those of the state of the art. In addition, we have the component of uncertainty estimation, which cannot be provided by the competing algorithms. It is clearly visible that BA benefits from the TG-MCMC initialization by an order of magnitude (shown in Gains column).

Table 8.2: Runtimes (seconds) of different algorithms. BA-X refers to BA with initialization X. *Points, Poses* refers to optimizing only points or only poses, respectively.

	PGO				BA-MinSpan		BA-Ours		
Dataset	Arrigoni	Torsello	Govindu	Ours	Points	Poses	Points	Poses	Gains
Madrid	0.137	2.255	3.033	5.794	53.82	139.6	33.30	14.54	**9.60x**
South Build.	0.060	0.784	1.213	1.986	46.18	108.0	6.087	4.969	**21.74x**

8.4 Discussion and Summary

We proposed two techniques to process pose chains and graph respectively. The former filter is suitable to filter and denoise camera trajectories while the latter addresses a more general scenario that is order free and allowed to have higher order edges/connections. In particular, we have proposed a new optimizer for this problem, called TG-MCMC, a manifold-aware, tempered rigid motion synchronization algorithm with a novel probabilistic formulation. TG-MCMC enjoys unique properties of trading-off approximately globally optimal solutions with non-asymptotic guarantees, to drawing samples from the posterior distribution, providing uncertainty estimates for the PGO-initialization problem.

TG-MCMC paves the way to a diverse potential future research: First, stochastic gradients can be employed to handle large problems, scaling up to hundreds of thousands of nodes. Next, the uncertainty estimates can be plugged into existing pipelines such as BA or PGO to further improve their quality. We also leave it as a future work to investigate different simulation schemes by altering the order of and combining differently the A, B and O steps. Finally, TG-MCMC can be extended to different problems, still maintaining its nice theoretical properties.

CHAPTER 9

CONCLUSION & FUTURE DIRECTIONS

"He must be very ignorant for he answers every question he is asked."

— François-Marie Arouet, *Voltaire*

While overcoming the drawbacks of 2D image representations, such as light selection, illumination design, perspective projection, and lack of metric scale, 3D vision introduces other challenges such as occlusions, point density variations, lack of appearance information and noise. These open up a whole new set of challenges for automated 3D scene understanding.

Out of all these challenges, this thesis offered new geometric perspectives to 3D reconstruction. First, we have proposed a new pipeline that can make benefit from the available CAD models, instead of relegating their usage to the final stage of deviation analysis. This pipeline has required many building blocks: point cloud and CAD preprocessing, object detection and pose estimation in 3D point clouds and pose graph optimization. We have delved deeply into details of these sub-components and presented our contributions for each of them. We have further presented two works that are suited to ground truth acquisition and sparse reconstruction of 3D coordinates: our specialized fiducials, *X-tag* and an online measurement system that is capable of calibrating non-overlapping cameras.

Once again, we have mainly contributed to the challenging problems of the litera-

ture with the following:

1. *X-tag* markers for robust 3D keypoint and camera pose localization under severe perspective distortions.
2. A new pipeline for online quality inspection of industrial parts. By using sparse point triangulations, this system is capable of calibrating non-overlapping static cameras and use them later in a real-time measurement stage.
3. A new, industry-grade dense 3D reconstruction pipeline that can consume available CAD models as priors to advance upon the reconstruction literature.
4. Algorithms for sampling CAD models and point clouds for better conditioning of the design models for use in computer vision applications.
5. A variety of geometric and learning based 3D object detection methods for 6DoF rigid pose and for parametric quadrics.
6. Pose graph optimization methods for ordered and unordered input. The former is a smoothing filter, whereas the latter optimizes the loop closure constraint with guarantees and can unveil the uncertainty of the poses.

We have validated all of these contributions qualitatively and quantitatively on multitudes of synthetic and real data.

9.1 Limitations

What gives our methods their power is also what curses them: Geometry. We require extensive 3D geometric structure to operate. Even though we have addressed the detection and use of textureless objects, our approaches do not extend to the case of symmetric objects or objects that are geometrically uninteresting. We also skipped touching upon many valuable problems of 3D vision, such as semantic segmentation or tracking. While segmentation would not be a direct necessity for our methods, it is certainly a nice-to-have by-product. Tracking on the other hand, would allow us to better use the ordered camera chains, that are common, for instance, in autonomous driving. We sacrificed tracking for the sake of generality and focused on detection.

9.2 Future Work

Our dense reconstruction framework is based on several building blocks. In this regard, it paves the way for further research in many directions: the distinct components can be improved or replaced completely. In this section, we will give a pinch of future directions we plan to follow. With the basis formed in this thesis, we plan to address the following future problems:

9.2.1 Deep Learning Based 3D Object Detectors

As the trend follows the paradigm shift towards learning based methods, a natural way is to transition our building blocks into learned ones. Because, pose estimation in cluttered data is arguably the most crucial and error-prone block, we think that it can be improved through the use of deep networks. Of course, such approach should respect the data modality. Luckily, recent years have given rise to many sparse, 3D network architectures including but not limited to Multiview CNNs [255], PointNet [221], OctNet [227], Graph CNNs [79] and et cetera., some of which are amenable for pose estimation. A straightforward idea would then be porting the 2D detectors such as SSD6D [147], BB8 [225], iPose [139], PoseCNN [295] and PoseAgent [155] to 3D.

9.2.2 Use of TG-MCMC in More Challenging Problems

TG-MCMC provides a novel framework combining Bayesian reasoning with optimization. Its unique properties are: 1. capability to estimate uncertainty, 2. sampling on the Riemannian manifolds and 3. optimization on the manifolds. For any optimization problem that can be formulated on the manifold, with a known distribution and geodesic flow, TG-MCMC is applicable. Below we provide one such application, where consistency constraint of permutation matrices is optimized.

9.2.2.1 Synchronizing Permutations for Better Correspondence Estimation

A *permutation matrix* is defined as a sparse, square binary matrix, where each column and each row is allowed to have only a single *true (1)* value:

$$\mathbf{P} := \{\mathbf{X} \in \{0,1\}^{n \times n} : \mathbf{X}\mathbf{1}_n = \mathbf{1}_n \,,\, \mathbf{1}_n^T \mathbf{X} = \mathbf{1}_n^T\}. \tag{9.1}$$

where $\mathbf{1}_n$ denotes a n-dimensional ones vector. $\mathbf{P} \in \mathbb{P}^n$ is a *total* permutation matrix and $\mathbf{P}_{(i,j)} = 1$ implies that element i is mapped to element j.

Manifold of permutations Taking a geometric standpoint, the *Birkhoff-von Neumann theorem* states that the convex hull of the set of permutation matrices is the set of doubly-stochastic matrices: non-negative, square matrices whose rows and columns sum to 1. Usually, the convex hull is treated as a continuous relaxation of the original discrete space. The doubly-stochastic matrices live on the Birkhoff Polytope [38], an $(n-1)^2$ dimensional convex submanifold of the ambient $\mathbb{R}^{n\times n}$, defined as:

$$\mathscr{DP}_n = \{\, \mathbf{X} : X_{ij} > 0\,,\ \forall i,j \in 1\cdots n\ \wedge \sum_{i=1}^{n} X_{ij} = 1 \wedge \sum_{j=1}^{n} X_{ij} = 1 \,\} \tag{9.2}$$

This is also referred as the doubly stochastic multinomial manifold [83]. In fact, $\mathbf{P}$ can be considered as an orthogonal subset of $\mathscr{DP}_n$: $\mathbf{P} = \{\mathbf{X} \in \mathscr{DP}_n : \mathbf{X}\mathbf{X}^T = \mathbf{I}\}$, i.e. the discrete set of permutation matrices is the intersection of the convex set of doubly stochastic matrices and the manifold of orthogonal matrices. Douik *et al.* [83] endow $\mathscr{DP}_n$ by the Fisher information metric, resulting in the Riemannian manifold of $\mathscr{DP}_n$, with tangent space:

$$\mathscr{T}_X\mathscr{DP}_n := \{\mathbf{X} \in \mathbb{R}^{p\times p} : \mathbf{X}\mathbf{1}_p = \mathbf{0}_p\,,\ \mathbf{1}_p^T\mathbf{X} = \mathbf{0}_p^T\}. \tag{9.3}$$

Authors also provide operators for projection and retraction map.

Partial permutations We define a *partial permutation matrix*, that allows for zero rows in order to explain the missing data / non-matching elements:

$$\hat{\mathbf{P}} := \{\mathbf{X} \in \{0,1\}^{m\times n} : \mathbf{X}\mathbf{1}_n \leq \mathbf{1}_m\,,\ \mathbf{1}_m^T\mathbf{X} \leq \mathbf{1}_n^T\}. \tag{9.4}$$

This is similar to the *total* counterpart, but allows for different number of correspondences. Partial permutations belong to the so-called *Symmetric Inverse Semigroup*.

Permutation synchronization problem Given n images, with m_i feature points at each image i, let $\hat{\mathbf{P}}_{ij}^{hk} = 1$ when feature k in image j is matched with feature h in image i and $\hat{\mathbf{P}}_{ij}^{hk} = 0$ everywhere else. The partial-ness of the permutation allows for zero

rows and columns, incorporating points (a.k.a. features, which in this context, refer to the different attributes of the same entity) that are not matched in other images.

Now let us assume a universe, that contains all the features in all the images. An *absolute permutation* $\mathbf{P}_i$ encodes the correspondences between image i and the universe. Under ideal circumstances, the partial permutation group is closed under multiplications and thus, the *relative* correspondences between images i and j, $\mathbf{P}_{ij}$, can be equivalently obtained by first computing the matches between image i and the universe, and then between universe and image j, giving rise to the *cycle consistency constraint*: $\mathbf{P}_{ij} = \mathbf{P}_i \mathbf{P}_j^T$. The problem of recovering the absolute permutations given a set of relative ones is the goal of permutation synchronization [16, 27, 186, 215, 238].

9.2.3 Full Pose Graph Optimization

While we have presented TG-MCMC as a global initialization of pose graph optimization, it is still an open question how to best solve the refinement of multiple scans (actual pose graph optimization conditioned on the point cloud data). While the literature is pretty rich, the studies have not been united under a common framework and cannot easily handle the amount of data coming from the modern laser scanners. Moreover, the uncertainty in such a registration scheme is hard to compute, giving rise to future research questions.

Finally, we leave it also as a future work to plug individual building blocks proposed in this work back into the pipeline, with an improved and advanced processing stack.

APPENDIX A

APPENDIX A. POSE GRAPH OPTIMIZATION

A.1 Proof of Proposition 8.2.1

Proof. We start by rewriting the SDE given in (8.2.2.2) as follows:

$$d\boldsymbol{\varphi}_t = \left\{ -\left(\underbrace{\begin{bmatrix} 0 & 0 \\ 0 & \frac{c\mathbf{M}^\top\mathbf{M}}{\beta}\mathbf{I} \end{bmatrix}}_{\mathbf{D}} + \underbrace{\begin{bmatrix} 0 & -\frac{\mathbf{I}}{\beta} \\ \frac{\mathbf{I}}{\beta} & 0 \end{bmatrix}}_{\mathbf{Q}} \right) \underbrace{\begin{bmatrix} \mathscr{A}(\tilde{\mathbf{x}}_t, \mathbf{p}_t, \beta) \\ \beta\mathbf{G}^{-1}\mathbf{p}_t \end{bmatrix}}_{\nabla_{\boldsymbol{\varphi}}\mathscr{E}_\lambda(\boldsymbol{\varphi}_t)} \right\} dt + \sqrt{2\mathbf{D}}\, dW_t. \tag{A.1}$$

where $\mathscr{A}(\tilde{\mathbf{x}}_t, \mathbf{p}_t, \beta) \triangleq \beta\nabla_{\tilde{\mathbf{x}}} U_\lambda(\tilde{\mathbf{x}}_t) + \frac{\beta}{2}\nabla_{\tilde{\mathbf{x}}}\log|\mathbf{G}| + \frac{\beta}{2}\nabla_{\tilde{\mathbf{x}}}(\mathbf{p}_t^\top\mathbf{G}^{-1}\mathbf{p}_t)$. Here, we observe that $\mathbf{D}$ is positive semi-definite, $\mathbf{Q}$ is anti-symmetric. Then, the desired result is a direct consequence of Theorem 1 of [181]. ■

A.2 Proof of Theorem 8.1

Before proving Theorem 8.1, we first prove the following intermediate results, whose proofs are given later in this document.

Corollary A.2.1. *Assume that **H**1 and **H**3 hold. Let* $\{\mathbf{x}_n, \mathbf{v}_n\}$ *be the output our algorithm with* $\beta > 0$. *Define* $\hat{U}_N \triangleq \frac{1}{N}\sum_{n=1}^{N} U(\mathbf{x}_n)$. *Then the following bound holds for the bias:*

$$\left|\mathbb{E}\hat{U}_N - \bar{U}_\beta\right| = \mathscr{O}(\frac{\beta}{Nh} + \frac{h}{\beta}). \tag{A.2}$$

Lemma A.2.2. *Assume that the conditions **H**1 and **H**2 hold. Then, the following bound holds for* $\beta \leq \frac{6}{L\pi^2}\log\frac{CL\pi^3 e}{3\Gamma(3/2)^{2/3}n}$*:*

$$\bar{U}_\beta - U^\star = \mathscr{O}\Big(\frac{1}{\beta}\Big), \tag{A.3}$$

where C *is defined in **H**2.*

A.2.1 Proof of Theorem 8.1

Proof. The proof is a direct application of Corollary A.2.1 and Lemma A.2.2. ■

A.3 Proof of Corollary A.2.1

Proof. From [62][Theorem 2], the bias of a standard SG-MCMC algorithm (i.e. $\beta = 1$) is bounded by

$$\mathscr{O}(\frac{1}{Nh'} + \frac{\sum_{n=1}^{N}\|\mathbb{E}\Delta V_n\|}{N} + h'). \tag{A.4}$$

where h' denotes the step-size and ΔV_n is an operator and it is related to bias of the stochastic gradient computations if there is any. If the iterates are obtained via full gradient computations ∇U or unbiased stochastic gradients computations (i.e. the case we consider here), then we have $\|\mathbb{E}\Delta V_n\| = 0$. Then by using a time-scaling argument similar to [226, 310], we define $h = \frac{h'}{\beta}$. This corresponds to running a standard SG-MCMC algorithm directly on the energy function $\mathscr{E}_{\mathscr{H}}(\mathbf{x}, \mathbf{v})$. The result is then obtained by replacing h' by $\frac{h}{\beta}$ in (A.4). ■

A.4 Proof of Lemma A.2.2

In order to prove Lemma A.2.2, we first need some rather elementary technical results, which we provide in appendix A.5 for clarity.

Proof. We use a similar proof technique to the one given in [226][Proposition 11]. We assume that $\pi_{\mathbf{x}}$ admits a density, denoted as $\rho(\mathbf{x}) \triangleq \frac{1}{Z_\beta}\exp(-\beta U(\mathbf{x}))$, where Z_β is the normalization constant:

$$Z_\beta \triangleq \int_{\mathscr{X}} \exp(-\beta U(\mathbf{x}))d\mathbf{x}. \tag{A.5}$$

We start by using the definition of $\bar{U}_\beta$, as follows:

$$\bar{U}_\beta = \int_{\mathscr{X}} U(\mathbf{x})\pi_{\mathbf{x}}(d\mathbf{x}) = \frac{1}{\beta}(H(\rho) - \log Z_\beta), \tag{A.6}$$

where $H(\rho)$ is the *differential entropy*, defined as follows:

$$H(\rho) \triangleq -\int_{\mathscr{X}} \rho(\mathbf{x})\log\rho(\mathbf{x})d\mathbf{x}. \tag{A.7}$$

We now aim at upper-bounding $H(\rho)$ and lower-bounding $\log Z_\beta$.

By Assumption **H**2, the distribution $\pi_{\mathbf{x}}$ has a finite second-order moment, therefore all the marginal distributions will also have bounded second order moments. By abusing the notation and denoting $\mathbf{x} \equiv \{\mathbf{q}_1, \ldots, \mathbf{q}_n, \mathbf{t}_1, \ldots, \mathbf{t}_n\}$, and by using the fact that the joint differential entropy is smaller than the sum of the differential entropies of the individual random variables, we can upper-bound $H(\rho)$ as follows:

$$H(\rho) \leq \sum_{i=1}^{n} H(\rho_{\mathbf{q}_i}) + H(\rho_{\mathbf{t}_1,\ldots,\mathbf{t}_n}), \tag{A.8}$$

where $\rho_{\mathbf{q}_i}$ denotes the marginal density of $\mathbf{q}_i$ and $\rho_{\mathbf{t}_1,\ldots,\mathbf{t}_n}$ denotes the joint marginal density of $(\mathbf{t}_1, \ldots, \mathbf{t}_n)$. Since $\rho_{\mathbf{t}_1,\ldots,\mathbf{t}_n}$ is defined on $\mathbb{R}^{3n}$, we know that $H(\rho_{\mathbf{t}_1,\ldots,\mathbf{t}_n})$ is upper-bounded by the differential entropy of a Gaussian distribution on $\mathbb{R}^{3n}$ that has the same second order moment. By denoting the covariance matrix of the Gaussian distribution with Σ, we obtain:

$$\begin{aligned}\mathscr{H}(\rho_{\mathbf{t}_1,\ldots,\mathbf{t}_n}) &\leq \frac{1}{2}\log[(2\pi e)^{3n}\det(\Sigma)] \\ &\leq \frac{1}{2}\log[(2\pi e)^{3n}\Big(\frac{\mathrm{tr}(\Sigma)}{3n}\Big)^{3n}] \\ &\leq \frac{3n}{2}\log\Big(2\pi e\frac{C}{3\beta n}\Big),\end{aligned} \tag{A.9}$$

The equations (1) and (1) follows by the relation between the arithmetic and geometric means, and Assumption **H**2.

By using a similar argument, since $\rho_{\mathbf{q}_i}$ lives on the unit sphere, its differential entropy is upper-bounded by the differential entropy of the uniform distribution on the unit sphere. Accordingly, we obtain:

$$H(\rho_{\mathbf{q}_i}) \leq \log\Big(\frac{(2\pi)^{3/2}}{\Gamma(3/2)}\Big), \tag{A.10}$$

where $\Gamma(\cdot)$ denotes the gamma function. By using (1) and (A.10) in (A.8), we obtain

$$\begin{aligned} H(\rho) \leq & n\log\Big(\frac{(2\pi)^{3/2}}{\Gamma(3/2)}\Big) + \frac{3n}{2}\log\Big(2\pi e\frac{C}{3\beta n}\Big) \\ = & \frac{3n}{2}\log\Big(\frac{2\pi}{\Gamma(3/2)^{2/3}}\Big) + \frac{3n}{2}\log\Big(2\pi e\frac{C}{3\beta n}\Big) \\ = & \frac{3n}{2}\log\Big(\frac{4\pi^2 eC}{3\Gamma(3/2)^{2/3}}\frac{1}{\beta n}\Big). \end{aligned} \tag{A.11}$$

We now lower-bound $\log Z_\beta$. By definition, we have

$$\begin{aligned} \log Z_\beta &= \log\int_{\mathscr{X}} \exp(-\beta U(\mathbf{x}))d\mathbf{x} \\ &= -\beta U^\star + \log\int_{\mathscr{X}} \exp(\beta(U^\star - U(\mathbf{x})))d\mathbf{x} \\ &\geq -\beta U^\star + \log\int_{\mathscr{X}} \exp(-\frac{\beta L\pi^2\|\mathbf{x}-\mathbf{x}^\star\|^2}{8})d\mathbf{x} \end{aligned} \tag{A.12}$$

Here, in (1) we used Assumption **H**1 and Corollary A.5.2 (presented below). By using $\mathbf{x} \equiv [\mathbf{q}_1^\top,\ldots,\mathbf{q}_n^\top,\mathbf{t}^\top]^\top$ and $\mathbf{x}^\star \equiv [(\mathbf{q}_1^\star)^\top,\ldots,(\mathbf{q}_n^\star)^\top,(\mathbf{t}^\star)^\top]^\top$, and $\mathbf{t} \equiv [\mathbf{t}_1^\top,\ldots,\mathbf{t}_n^\top]^\top$, $\mathbf{t}^\star \equiv [(\mathbf{t}_1^\star)^\top,\ldots,(\mathbf{t}_n^\star)^\top]^\top$ we obtain:

$$\begin{aligned} \log Z_\beta \geq & -\beta U^\star + \log\Big(\prod_{i=1}^n\int_{\mathbb{S}^3}\exp(-\frac{\beta L\pi^2\|\mathbf{q}_i-\mathbf{q}_i^\star\|^2}{8})d\mathbf{q}_i\Big) \\ & + \log\Big(\int_{\mathbb{R}^{3n}}\exp(-\frac{\beta L\pi^2\|\mathbf{t}-\mathbf{t}^\star\|^2}{8})d\mathbf{t}\Big) \\ = & -\beta U^\star + \log\Big(\prod_{i=1}^n\int_{\mathbb{S}^3}\exp(-\frac{\beta L\pi^2\|\mathbf{q}_i-\mathbf{q}_i^\star\|^2}{8})d\mathbf{q}_i\Big) \\ & + \frac{3n}{2}\log(\frac{4}{\beta L\pi}). \end{aligned} \tag{A.13}$$

Let us focus on the integral with respect to $\mathbf{q}_i$. By definition, we have:

$$\begin{aligned} \int_{\mathbb{S}^3}\exp(-\frac{\beta L\pi^2\|\mathbf{q}_i-\mathbf{q}_i^\star\|^2}{8})d\mathbf{q}_i &= \int_{\mathbb{S}^3}\exp\Big(-\frac{\beta L\pi^2}{8}(2-2\mathbf{q}_i^\top\mathbf{q}_i^\star)\Big)d\mathbf{q}_i \\ &= \exp\Big(-\frac{\beta L\pi^2}{4}\Big)\int_{\mathbb{S}^3}\exp\Big(\frac{\beta L\pi^2}{4}\mathbf{q}_i^\top\mathbf{q}_i^\star)\Big)d\mathbf{q}_i. \end{aligned} \tag{A.14}$$

By using the connection between the integral on the right hand side of the above equation and multivariate Watson's distribution (see Equations 2.1 and 2.2 in [250]), we obtain:

$$\int_{\mathbb{S}^3} \exp(-\frac{\beta L\pi^2 \|\mathbf{q}_i - \mathbf{q}_i^\star\|^2}{8}) d\mathbf{q}_i = \exp\Big(-\frac{\beta L\pi^2}{4}\Big) 2\pi^2 M(\frac{1}{2}, 2, \frac{\beta L\pi^2}{4}), \tag{A.15}$$

where M denotes the Kummer confluent hypergeometric function that is defined as follows:

$$M(a,b,c) \triangleq \sum_{j=0}^{\infty} \frac{a^{\bar{j}}}{b^{\bar{j}}} \frac{c^j}{j!}, \tag{A.16}$$

where $a^{\bar{0}} \triangleq 1$ and $a^{\bar{j}} \triangleq a(a+1)\dots(a+j-1)$ for $j \geq 1$. By Theorem 3 of [144], we know that

$$M(\frac{1}{2}, 2, \frac{\beta L\pi^2}{4}) \geq \frac{1}{1 + \frac{\beta L\pi^2}{16}} = \frac{16}{16 + \beta L\pi^2}. \tag{A.17}$$

By using this inequality in (A.15), we obtain:

$$\int_{\mathbb{S}^3} \exp(-\frac{\beta L\pi^2 \|\mathbf{q}_i - \mathbf{q}_i^\star\|^2}{8}) d\mathbf{q}_i \geq \exp\Big(-\frac{\beta L\pi^2}{4}\Big) \frac{32\pi^2}{16 + \beta L\pi^2}. \tag{A.18}$$

We can insert (A.18) in (1), as follows:

$$\begin{aligned} \log Z_\beta &\geq -\beta U^\star + \frac{3n}{2} \log(\frac{4}{\beta L\pi}) + \sum_{i=1}^{n} \log \int_{\mathbb{S}^3} \exp(-\frac{\beta L\pi^2 \|\mathbf{q}_i - \mathbf{q}_i^\star\|^2}{8}) d\mathbf{q}_i \\ &\geq -\beta U^\star + \frac{3n}{2} \log(\frac{4}{\beta L\pi}) - n\frac{\beta L\pi^2}{4} + n \log \frac{32\pi^2}{16 + \beta L\pi^2} \\ &\geq -\beta U^\star + \frac{3n}{2} \log(\frac{4}{\beta L\pi}) - n\frac{\beta L\pi^2}{4} \end{aligned} \tag{A.19}$$

Finally, by combining (A.6), (1), and (1), we obtain:

$$\begin{aligned} \bar{U}_\beta - U^\star &= \frac{1}{\beta}(\mathscr{H}(\rho) - \log Z_\beta) - U^\star \\ &\leq \frac{3n}{2\beta} \log\Big(\frac{4\pi^2 eC}{3\Gamma(3/2)^{2/3}} \frac{1}{\beta n}\Big) - \frac{3n}{2\beta} \log(\frac{4}{\beta L\pi}) + n\frac{L\pi^2}{4} \\ &= \frac{3n}{2\beta} \log\Big(\frac{CL\pi^3 e}{3\Gamma(3/2)^{2/3} n}\Big) + n\frac{L\pi^2}{4} \\ &\leq \frac{3n}{\beta} \log\Big(\frac{CL\pi^3 e}{3\Gamma(3/2)^{2/3} n}\Big). \end{aligned} \tag{A.20}$$

The last line follows from the hypothesis. This finalizes the proof. ■

A.5 Technical Results

In the following lemma, we generalize [207][Lemma 1.2.3] to manifolds. Similar arguments can be found in [174, 309].

Lemma A.5.1. *Let $\mathscr{X} \subset \mathbb{R}^n$ be a Rimannian manifold with metric $\mathrm{d}_{\mathscr{X}}$, and let $\gamma : [0,1] \mapsto \mathscr{X}$ be a constant-speed geodesic curve between two points $\mathbf{x}, \mathbf{y} \in \mathscr{X}$, such that $\gamma(0) = \mathbf{x}$ and $\gamma(1) = \mathbf{y}$. Let $f : \mathscr{X} \mapsto \mathbb{R}$ be a continuously differentiable function with Lipschitz continuous gradients. Then the following bound holds for every $\mathbf{x}, \mathbf{y} \in \mathscr{X}$:*

$$\left| f(\mathbf{y}) - f(\mathbf{x}) - \int_0^1 \langle \nabla f(\mathbf{x}), \gamma'(t) \rangle dt \right| \leq \frac{L}{2} \mathrm{d}_{\mathscr{X}}(\mathbf{x}, \mathbf{y})^2, \tag{A.21}$$

where $\langle \mathbf{x}, \mathbf{y} \rangle \triangleq \mathbf{x}^\top \mathbf{y}$ and L denotes the Lipschitz constant.

Proof. Let us define a function $\varphi : [0,1] \mapsto \mathbb{R}$, such that $\varphi(t) \triangleq f(\gamma(t))$. By definition, we have $\varphi(0) = f(\mathbf{x})$ and $\varphi(1) = f(\mathbf{y})$. By using the second fundamental theorem of calculus, we can write:

$$\varphi(1) - \varphi(0) = \int_0^1 \varphi'(t) dt, \tag{A.22}$$

where $\varphi'(t)$ denotes the derivative of $\varphi(t)$ with respect to t. By the theorem of derivation of composite functions, we have

$$\varphi'(t) = \langle \nabla f(\gamma(t)), \gamma'(t) \rangle. \tag{A.23}$$

By combining (A.22) and (A.23), we obtain the following identity for all $\mathbf{x}, \mathbf{y} \in \mathscr{X}$:

$$\begin{aligned} f(\mathbf{y}) &= f(\mathbf{x}) + \int_0^1 \langle \nabla f(\gamma(t)), \gamma'(t) \rangle dt \\ &= f(\mathbf{x}) + \int_0^1 \langle \nabla f(\mathbf{x}), \gamma'(t) \rangle dt + \int_0^1 \langle \nabla f(\gamma(t)) - f(\mathbf{x}), \gamma'(t) \rangle dt. \end{aligned} \tag{A.24}$$

Therefore, we obtain

$$\begin{aligned} \left| f(\mathbf{y}) - f(\mathbf{x}) - \int_0^1 \langle \nabla f(\mathbf{x}), \gamma'(t) \rangle dt \right| &= \left| \int_0^1 \langle \nabla f(\gamma(t)) - f(\mathbf{x}), \gamma'(t) \rangle dt \right| \\ &\leq \int_0^1 \left| \langle \nabla f(\gamma(t)) - f(\mathbf{x}), \gamma'(t) \rangle \right| dt \\ &\leq \int_0^1 \| \nabla f(\gamma(t)) - f(\mathbf{x}) \| \| \gamma'(t) \| dt \\ &\leq L \int_0^1 \mathrm{d}_{\mathscr{X}}(\gamma(t), \mathbf{x}) \| \gamma'(t) \| dt. \end{aligned} \tag{A.25}$$

We can now use the fact that the geodesic curve has a constant velocity, such that $\|\gamma'(t)\| = \mathrm{d}_{\mathscr{X}}(\mathbf{x},\mathbf{y})$ for all $t \in [0,1]$, which also implies $\mathrm{d}_{\mathscr{X}}(\gamma(t_1),\gamma(t_2)) = |t_1 - t_2|\mathrm{d}_{\mathscr{X}}(\gamma(1),\gamma(0))$. Then, using $\mathbf{x} = \gamma(0)$, $\mathbf{y} = \gamma(1)$, we obtain:

$$\begin{aligned}\left| f(\mathbf{y}) - f(\mathbf{x}) - \int_0^1 \langle \nabla f(\mathbf{x}), \gamma'(t) \rangle \, dt \right| &\leq L \int_0^1 t \mathrm{d}_{\mathscr{X}}(\mathbf{x},\mathbf{y})^2 dt \\ &= \frac{L}{2} \mathrm{d}_{\mathscr{X}}(\mathbf{x},\mathbf{y})^2.\end{aligned} \tag{A.26}$$

This concludes the proof. ■

Corollary A.5.2. *Under the assumptions of Lemma A.5.1, the following bound holds for all* $\mathbf{x} \in \mathscr{X}$

$$f(\mathbf{x}) - f^\star \leq \frac{L\pi^2}{8} \|\mathbf{x} - \mathbf{x}^\star\|^2, \tag{A.27}$$

where $\mathscr{X} \triangleq (\mathbb{S}^3)^n \times \mathbb{R}^{3n}$, $f^\star = \min_{\mathbf{x}' \in \mathscr{X}} f(\mathbf{x}')$ *and* $\mathbf{x}^\star = \arg\min_{\mathbf{x}' \in \mathscr{X}} f(\mathbf{x}')$.

Proof. By using Lemma A.5.1 and the obvious facts that $\nabla f(\mathbf{x}^\star) = 0$ and $f(\mathbf{x}) > f^\star$ for all $\mathbf{x} \in \mathscr{X}$, we have:

$$f(\mathbf{x}) - f^\star \leq \frac{L}{2} \mathrm{d}_{\mathscr{X}}(\mathbf{x},\mathbf{x}^\star)^2. \tag{A.28}$$

The inequality given in Equation A.1.1 in [75] states that the geodesic distance on the sphere is bounded by the 2-norm, more precisely, for all $\mathbf{q}, \mathbf{q}' \in \mathbb{S}^{d-1}$ we have:

$$\mathrm{d}_{\mathbb{S}^{d-1}}(\mathbf{q},\mathbf{q}') \leq \frac{\pi}{2} \|\mathbf{q} - \mathbf{q}'\|. \tag{A.29}$$

Using $\mathbf{x} \equiv [\mathbf{q}_1^\top, \ldots, \mathbf{q}_n^\top, \mathbf{t}_1^\top, \ldots, \mathbf{t}_n^\top]^\top$ and $\mathbf{x}^\star \equiv [(\mathbf{q}_1^\star)^\top, \ldots, (\mathbf{q}_n^\star)^\top, (\mathbf{t}_1^\star)^\top, \ldots, (\mathbf{t}_n^\star)^\top]^\top$ yields:

$$\begin{aligned} f(\mathbf{x}) - f^\star &\leq \frac{L}{2} \Big(\sum_{i=1}^n \mathrm{d}_{\mathbb{S}^3}(\mathbf{q}_i, \mathbf{q}_i^\star)^2 + \sum_{i=1}^n \|\mathbf{t}_i - \mathbf{t}_i^\star\|^2 \Big) \\ &\leq \frac{L}{2} \Big(\frac{\pi^2}{4} \sum_{i=1}^n \|\mathbf{q}_i - \mathbf{q}_i^\star\|^2 + \sum_{i=1}^n \|\mathbf{t}_i - \mathbf{t}_i^\star\|^2 \Big) \\ &\leq \frac{L\pi^2}{8} \Big(\sum_{i=1}^n \|\mathbf{q}_i - \mathbf{q}_i^\star\|^2 + \sum_{i=1}^n \|\mathbf{t}_i - \mathbf{t}_i^\star\|^2 \Big) \\ &= \frac{L\pi^2}{8} \|\mathbf{x} - \mathbf{x}^\star\|^2. \end{aligned} \tag{A.30}$$

This concludes the proof. ■

A.6 Gradients of Likelihood and Prior Terms

In this section we provide derivations of the gradients for data and prior terms. For completeness, we find it worthy to once again repeat our MLE formulation:

$$\arg\max_{\mathbf{Q},\mathbf{T}} \Big(\sum_{(i,j)\in E} \{\log p(\mathbf{q}_{ij}|\mathbf{Q},\mathbf{T}) + \log p(\mathbf{t}_{ij}|\mathbf{Q},\mathbf{T})\} + \sum_i \log p(\mathbf{q}_i) + \sum_i \log p(\mathbf{t}_i) \Big). \tag{A.31}$$

We begin by deriving the gradients of the **rotational components** first, and translations second. To ease implementation and increase efficiency, we drop the first column of $\mathbf{V}$ as it is given by the mode. So, with the abuse of notation: $\mathbf{V} \in \mathbb{R}^{4\times 3}$ In the setting where $\mathbf{V}$ is constant w.r.t. $\mathbf{q}$ the gradient of log Bingham distribution w.r.t. the random variable $\mathbf{q}$ reads:

$$\nabla_{\mathbf{x}} \log \mathscr{B}(\mathbf{x};\mathbf{\Lambda},\mathbf{V}) = \nabla_{\mathbf{x}} \log \frac{1}{F} + \nabla_{\mathbf{x}}(\mathbf{x}^T\mathbf{V}\mathbf{\Lambda}\mathbf{V}^T\mathbf{x}) = (\mathbf{V}\Lambda\mathbf{V}^T + \mathbf{V}\mathbf{\Lambda}^T\mathbf{V}^T)\mathbf{x} = 2\mathbf{V}\Lambda\mathbf{V}^T\mathbf{x}. \tag{A.32}$$

Normalizing constant drops as it depends on $\mathbf{\Lambda}$ only [157]. We also have cases where $\mathbf{V}$ is a function of the mode $\mathbf{q}$, $\mathbf{V} \to \mathbf{V}(\mathbf{q})$. Then:

$$\nabla_{\mathbf{q}} \log \mathscr{B}(\mathbf{x};\mathbf{\Lambda},\mathbf{V}(\mathbf{q})) = \nabla_{\mathbf{q}}(\mathbf{x}^T\mathbf{V}(\mathbf{q})\mathbf{\Lambda}\mathbf{V}^T(\mathbf{q})\mathbf{x}) = \nabla_{\mathbf{k}}(\mathbf{k}^\mathbf{T}\mathbf{\Lambda}\mathbf{k})\,\nabla_{\mathbf{q}}(\mathbf{k}) = 2\mathbf{k}^T\mathbf{\Lambda}\,\nabla_{\mathbf{q}}(\mathbf{k}), \tag{A.33}$$

where $\mathbf{k} = \mathbf{V}^T(\mathbf{q})\mathbf{x} \in \mathbb{R}^3$ is used to ease the computations. Note that in our particular application it is the case that $\mathbf{x} \leftarrow \mathbf{q}_{ij}$, i.e. the data is specified by the relative poses attached to the edges of the graph. We then speak of the gradient of $\log(p(\mathbf{q}_{ij}\,|\,\mathbf{q}_i,\mathbf{q}_j))$ with $\mathbf{V} \to \mathbf{V}(\mathbf{q}_j\bar{\mathbf{q}}_i)$ w.r.t. $\mathbf{q}_i$. We shorten $\mathbf{r} \leftarrow \mathbf{q}_j\bar{\mathbf{q}}_i$ and write $\mathbf{V}$ as a function of $\mathbf{r}$, $\mathbf{V}(\mathbf{r}) \triangleq \mathbf{V}(\mathbf{q}_j\bar{\mathbf{q}}_i)$. Then:

$$\nabla_{\mathbf{q}_i} \log \mathscr{B}(\mathbf{x};\mathbf{\Lambda},\mathbf{V}(\mathbf{r})) = \nabla_{\mathbf{r}} \log \mathscr{B}(\mathbf{x};\mathbf{\Lambda},\mathbf{V}(\mathbf{r}))\nabla_{\mathbf{q}_i}(\mathbf{r}) = 2\mathbf{k}^T\mathbf{\Lambda}\,\nabla_{\mathbf{r}}(\mathbf{k})\,\nabla_{\mathbf{q}_i}(\mathbf{r}), \tag{A.34}$$

this time with $\mathbf{k} = \mathbf{V}^T(\mathbf{r})\mathbf{x} \in \mathbb{R}^3$. Note that $\nabla_{\mathbf{r}} \log \mathscr{B}(\mathbf{x};\mathbf{\Lambda},\mathbf{V}(\mathbf{r}))$ is expanded as in Eq. A.33. Using the definition of $\mathbf{V}$ in Eq. 8.11, the terms simplify to:

$$\mathbf{k} = \begin{bmatrix} q_1x_2 - q_2x_1 + q_3x_4 - q_4x_3 \\ q_1x_3 - q_3x_1 - q_2x_4 + q_4x_2 \\ q_1x_4 + q_2x_3 - q_3x_2 - q_4x_1 \end{bmatrix} \qquad \nabla_{\mathbf{q}}(\mathbf{k}) = \begin{bmatrix} x_2 & -x_1 & x_4 & -x_3 \\ x_3 & -x_4 & -x_1 & x_2 \\ x_4 & x_3 & -x_2 & -x_1 \end{bmatrix}. \tag{A.35}$$

The last term in Eq. A.34 expands as:

$$\nabla_{\mathbf{q}_i}(\mathbf{q}_j\bar{\mathbf{q}}_i) = \begin{bmatrix} q_{j,1} & q_{j,2} & q_{j,3} & q_{j,4} \\ q_{j,2} & -q_{j,1} & q_{j,4} & -q_{j,3} \\ q_{j,3} & -q_{j,4} & -q_{j,1} & q_{j,2} \\ q_{j,4} & q_{j,3} & -q_{j,2} & -q_{j,1} \end{bmatrix}. \tag{A.36}$$

We will now derive the gradients for **translational components**. Similarly, we start by the gradient of the log likelihood w.r.t. the data. While a shorter derivation through matrix calculus is also possible, we deliberately provide a longer version, as it might be more intuitive:

$$\begin{aligned}\nabla_{\mathbf{t}} \log \mathcal{N}\left(\mathbf{t}; \boldsymbol{\mu}, \sigma^2 \boldsymbol{I}\right) &= \nabla_{\mathbf{t}} \log \frac{1}{G} + \nabla_{\mathbf{t}} \left(-\frac{1}{2}(\mathbf{t}-\boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{t}-\boldsymbol{\mu})\right) \\ &= \nabla_{\mathbf{t}} \Big(-\frac{1}{2}\big(\mathbf{t}^T \Sigma^{-1} \mathbf{t} - \mathbf{t}^T \Sigma^{-1} \boldsymbol{\mu} - \boldsymbol{\mu}^T \Sigma^{-1} \mathbf{t} + \boldsymbol{\mu}^T \Sigma^{-1} \boldsymbol{\mu}\big)\Big) \\ &= -\frac{1}{2}\big(\mathbf{t}^T (\Sigma^{-1} + \Sigma^{-T}) - (\Sigma^{-1}\boldsymbol{\mu})^T - (\boldsymbol{\mu}^T \Sigma^{-1}) + 0\big) \\ &= -\frac{1}{2}\big(2\mathbf{t}^T \Sigma^{-1} - 2\boldsymbol{\mu}^T \Sigma^{-1}\big) = (\boldsymbol{\mu}^T - \mathbf{t}^T)\Sigma^{-1}\end{aligned} \tag{A.37}$$

The normalizing constant drops similarly as it does not depend on $\mathbf{t}$.

Similar to rotational counterpart, our algorithm centers the data on the mean of the distribution, also requiring to compute the gradients w.r.t. the mean of the distribution. With a derivation similar to but simpler from Eq. A.6, it follows:

$$\nabla_{\boldsymbol{\mu}} \log \mathcal{N}\left(\mathbf{x}; \boldsymbol{\mu}, \sigma^2 \boldsymbol{I}\right) = -\frac{1}{2}\big(2\boldsymbol{\mu}^T \Sigma^{-1} - 2\mathbf{x}^T \Sigma^{-1}\big) = (\mathbf{x}^T - \boldsymbol{\mu}^T)\Sigma^{-1} \tag{A.38}$$

When we center the distribution on the data, substituting $\boldsymbol{\mu} \leftarrow \mathbf{t}_j - \mathbf{r}\mathbf{t}_i\bar{\mathbf{r}}$, where $\mathbf{r} \leftarrow \mathbf{q}_j\bar{\mathbf{q}}_i$, $\mathbf{x} \leftarrow \mathbf{t}_{ij}$, we arrive at:

$$\begin{aligned}\nabla_{\mathbf{t}_i} \log \mathcal{N}\left(\mathbf{x}; \boldsymbol{\mu}, \sigma^2 \boldsymbol{I}\right) &= \nabla_{\boldsymbol{\mu}} \log \mathcal{N}\left(\mathbf{x}; \boldsymbol{\mu}, \sigma^2 \boldsymbol{I}\right) J_{\mathbf{t}_i}(\boldsymbol{\mu}) \\ \nabla_{\mathbf{q}_i} \log \mathcal{N}\left(\mathbf{x}; \boldsymbol{\mu}, \sigma^2 \boldsymbol{I}\right) &= \nabla_{\boldsymbol{\mu}} \log \mathcal{N}\left(\mathbf{x}; \boldsymbol{\mu}, \sigma^2 \boldsymbol{I}\right) J_{\mathbf{q}_i}(\boldsymbol{\mu})\end{aligned} \tag{A.39}$$

Note that the first term of the right hand side is given in Eq. A.38. The second one can be computed from the derivative of the sandwich action on the 1-blade $\mathbf{t}_i$ and $\mathbf{q}_i$. Here, we describe $\mathbf{t}_i$ and $\mathbf{q}_i$ can be obtained in a similar fashion. With slight abuse of notation, in the following we assume that translation-quaternion is purified: $\mathbf{t}_i \leftarrow [0; \mathbf{t}_i]$.

$$J_{\mathbf{t}_i}(\boldsymbol{\mu}) = J_{\mathbf{t}_i}(\mathbf{t}_j - \mathbf{r}\mathbf{t}_i\bar{\mathbf{r}}) = -J_{\mathbf{t}_i}\Big(Q(\bar{\mathbf{r}})Q(\mathbf{t}_i)\mathbf{r}^T\Big)$$

$$= -J_{\mathbf{t}_i}\Big(\big(\mathbf{q}_{ij} \otimes Q(\bar{\mathbf{r}})\big)\,\mathrm{vec}\big(Q(\mathbf{t}_i)\big)\Big) = -J_{\mathbf{t}_i}\Big(\mathbf{K}\mathrm{vec}\big(Q(\mathbf{t}_i)\big)\Big) = -\mathbf{K}\nabla_{\mathbf{t}_i}\mathrm{vec}\big(Q(\mathbf{t}_i)\big) = -\mathbf{K}\mathbf{J}_{\mathbf{t}_i}$$

$$= \begin{bmatrix} 0 & 0 & 0 \\ -q_1^2 - q_2^2 + q_3^2 + q_4^2 & 2q_1q_4 - 2q_2q_3 & -2q_1q_3 - 2q_2q_4 \\ -2q_1q_4 - 2q_2q_3 & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2q_1q_2 - 2q_3q_4 \\ 2q_1q_3 - 2q_2q_4 & -2q_1q_2 - 2q_3q_4 & -q_1^2 + q_2^2 + q_3^2 - q_4^2 \end{bmatrix}$$

where $\mathbf{K} \in R^{4\times 16}$ refers to the Kronecker product matrix $\mathbf{K} = \mathbf{r} \otimes Q(\bar{\mathbf{q}}_{ij})$, $\mathbf{J}_{\mathbf{t}_i} = \nabla_{\mathbf{t}_i}\mathrm{vec}\big(Q(\mathbf{t}_i)\big)$ is the 16×3 Jacobian matrix and $\mathrm{vec}(\cdot)$ denotes the linearization operator. Individual components q_i belong to $\mathbf{r} = [q_1, q_2, q_3, q_4]$. The map $Q(\cdot) : \mathbb{H}_1 \rightarrow R^{4x4}$ constructs a Quaternion matrix form:

$$Q(\mathbf{q}) = \begin{bmatrix} q_1 & -q_2 & -q_3 & -q_4 \\ q_2 & q_1 & q_4 & -q_3 \\ q_3 & -q_4 & q_1 & q_2 \\ q_4 & q_3 & -q_2 & q_1 \end{bmatrix} \tag{A.40}$$

leading to a more compact notation of the quaternion product. In fact this is not very different from the definition of $\mathbf{V}(\mathbf{q})$, as one is free to pick any of the 48 distinct representations out of the matrix ring $\mathbb{M}(4,\mathbb{R})$. Note that Eq. A.6 has zeros in the initial row. This is due to the property that all the operations respect the purity of the blade. The final Jacobian matrix can be extracted from the remaining three rows corresponding to the vector part.

BIBLIOGRAPHY

[1] L. 3D, *Photogrammetry by linearis 3d*.
http://www.linearis3d.com/, 2016.

[2] K. AASTROM AND L. MORIN, *Random cross ratios*, Technical Report IMAG-RT - 92-088 ; LIFIA - 92-014, INRIA, 1992.

[3] R. ABLAMOWICZ AND G. SOBCZYK, *Lectures on Clifford (geometric) algebras and applications*, Springer Science & Business Media, 2004.

[4] S. AGARWAL, K. MIERLE, AND OTHERS, *Ceres solver*.
`http://ceres-solver.org`, 2018.
Accessed: 2018-05-15.

[5] AICON3D, *Aicon 3d systems - move inspect technology - dpa*, 2016.

[6] M. ALEXA, J. BEHR, D. COHEN-OR, S. FLEISHMAN, D. LEVIN, AND C. T. SILVA, *Computing and rendering point set surfaces*, Visualization and Computer Graphics, IEEE Transactions on, 9 (2003), pp. 3–15.

[7] M. ALEXA, S. RUSINKIEWICZ, D. NEHAB, AND P. SHILANE, *Stratified point sampling of 3d models*, in Eurographics, 2004.

[8] S. ALLAIRE, J.-J. JACQ, V. BURDIN, C. ROUX, AND C. COUTURE, *Type-constrained robust fitting of quadrics with application to the 3d morphological characterization of saddle-shaped articular surfaces*, in International Conference on Computer Vision, IEEE, 2007.

[9] P. ALLIEZ, G. UCELLI, C. GOTSMAN, AND M. ATTENE, *Recent advances in remeshing of surfaces*, in Shape Analysis and Structuring, Springer, 2008.

[10] J. ANDREWS, *User-guided inverse 3d modeling*, tech. rep., EECS Department. University of California, Berkeley, May 2013.

[11] J. ANDREWS AND C. H. SÉQUIN, *Type-constrained direct fitting of quadric surfaces*, Computer-Aided Design and Applications, (2014).

[12] R. ARORA, Y. H. HU, AND C. DYER, *Estimating correspondence between multiple cameras using joint invariants*, in 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2009, pp. 805–808.

[13] F. ARRIGONI, A. FUSIELLO, AND B. ROSSI, *Spectral motion synchronization in se (3)*, arXiv preprint arXiv:1506.08765, (2015).

[14] ——, *Camera motion from group synchronization*, in 3D Vision (3DV), 2016 Fourth International Conference on, IEEE, 2016, pp. 546–555.

[15] F. ARRIGONI, L. MAGRI, B. ROSSI, P. FRAGNETO, AND A. FUSIELLO, *Robust absolute rotation estimation via low-rank and sparse matrix decomposition*, in 3D Vision (3DV), 2014 2nd International Conference on, vol. 1, IEEE, 2014, pp. 491–498.

[16] F. ARRIGONI, E. MASET, AND A. FUSIELLO, *Synchronization in the symmetric inverse semigroup*, in International Conference on Image Analysis and Processing, Springer, 2017, pp. 70–81.

[17] F. ARRIGONI, B. ROSSI, AND A. FUSIELLO, *Global registration of 3d point sets via lrs decomposition*, in European Conference on Computer Vision, Springer, 2016, pp. 489–504.

[18] B. ATCHESON, F. HEIDE, AND W. HEIDRICH, *Caltag: High precision fiducial markers for camera calibration*, VMV, (2010).

[19] J. AULINAS, Y. PETILLOT, J. SALVI, AND X. LLADÓ, *The slam problem: A survey*, in Proceedings of the 2008 Conference on Artificial Intelligence Research and Development: Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence, Amsterdam, The Netherlands, The Netherlands, 2008, IOS Press, pp. 363–371.

[20] N. AYACHE AND F. LUSTMAN, *Trinocular stereo vision for robotics*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 13 (1991), pp. 73–85.

[21] H. BADINO, D. HUBER, Y. PARK, AND T. KANADE, *Fast and accurate computation of surface normals from range images*, in Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE, 2011, pp. 3084–3091.

[22] D. BEALE, Y.-L. YANG, N. CAMPBELL, D. COSKER, AND P. HALL, *Fitting quadrics with a bayesian prior*, Computational Visual Media, (2016).

[23] M. BEETZ, U. KLANK, I. KRESSE, A. MALDONADO, L. MOSENLECHNER, D. PANGERCIC, T. RUHR, AND M. TENORTH, *Robotic roommates making pancakes*, in Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on, IEEE, 2011, pp. 529–536.

[24] F. BERGAMASCO, A. ALBARELLI, L. COSMO, E. RODOLA, AND A. TORSELLO, *An accurate and robust artificial marker based on cyclic codes*, IEEE Transactions on Pattern Analysis and Machine Intelligence, PP (2016), pp. 1–1.

[25] F. BERGAMASCO, A. ALBARELLI, AND A. TORSELLO, *Pi-tag: a fast image-space marker design based on projective invariants*, Machine vision and applications, 24 (2013), pp. 1295–1310.

[26] F. BERGAMASCO, L. COSMO, A. ALBARELLI, AND A. TORSELLO, *A robust multi-camera 3d ellipse fitting for contactless measurements*, in 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on, IEEE, 2012, pp. 168–175.

[27] F. BERNARD, J. THUNBERG, J. GONCALVES, AND C. THEOBALT, *Synchronisation of partial multi-matchings via non-negative factorisations*, CoRR, abs/1803.06320 (2018).

[28] P. J. BESL AND N. D. MCKAY, *Method for registration of 3-d shapes*, in Robotics-DL tentative, International Society for Optics and Photonics, 1992, pp. 586–606.

[29] C. BINGHAM, *An antipodally symmetric distribution on the sphere*, The Annals of Statistics, (1974), pp. 1201–1225.

[30] T. BIRDAL, *Task oriented 3d sampling via genetic algorithms*, in 2018 26th Signal Processing and Communications Applications Conference (SIU), IEEE, 2018, pp. 1–4.

[31] T. BIRDAL, E. BALA, T. EREN, AND S. ILIC, *Online inspection of 3d parts via a locally overlapping camera network*, in WACV 2016: IEEE Winter Conference on Applications of Computer Vision, IEEE, 2016.

[32] T. BIRDAL, B. BUSAM, N. NAVAB, S. ILIC, AND P. STURM, *A minimalist approach to type-agnostic detection of quadrics in point clouds*, in Computer Vision and Pattern Recognition (CVPR), IEEE, 2018.

[33] T. BIRDAL, U. ŞIMŞEKLI, M. ONUR EKEN, AND S. ILIC, *Bayesian Pose Graph Optimization via Bingham Distributions and Tempered Geodesic MCMC*, ArXiv e-prints, (2018).

[34] T. BIRDAL, I. DOBRYDEN, AND S. ILIC, *X-tag: A fiducial tag for flexible and accurate bundle adjustment*, in IEEE Conference on 3D Vision, 2016.

[35] T. BIRDAL AND S. ILIC, *Point pair features based object detection and pose estimation revisited*, in 3D Vision, IEEE, 2015, pp. 527–535.

[36] ——, *Cad priors for accurate and flexible instance reconstruction*, in The IEEE International Conference on Computer Vision (ICCV), Oct 2017.

[37] ——, *A point sampling algorithm for 3d matching of irregular geometries*, in International Conference on Intelligent Robots and Systems (IROS 2017), IEEE, 2017.

[38] G. BIRKHOFF, *Tres observaciones sobre el algebra lineal*, Univ. Nac. Tucumán Rev. Ser. A, 5 (1946), pp. 147–151.

[39] M. M. BLANE, Z. LEI, H. ÇIVI, AND D. B. COOPER, *The 3l algorithm for fitting implicit polynomial curves and surfaces to data*, IEEE Transactions on Pattern Analysis and Machine Intelligence, (2000).

[40] F. BONARRIGO AND A. SIGNORONI, *An enhanced'optimization-on-a-manifold'framework for global registration of 3d range data*, in 3D

Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on, IEEE, 2011, pp. 350–357.

[41] D. BORRMANN, J. ELSEBERG, K. LINGEMANN, AND A. NÜCHTER, *The 3d hough transform for plane detection in point clouds: A review and a new accumulator design*, 3D Research, 2 (2011).

[42] S. BOUAZIZ, A. TAGLIASACCHI, AND M. PAULY, *Sparse iterative closest point*, Computer Graphics Forum (Symposium on Geometry Processing), 32 (2013), pp. 1–11.

[43] J.-Y. BOUGUET, *Camera calibration toolbox for m atlab*, (2004).

[44] E. BRACHMANN, A. KRULL, F. MICHEL, S. GUMHOLD, J. SHOTTON, AND C. ROTHER, *Learning 6d object pose estimation using 3d object coordinates*, in Computer Vision–ECCV 2014, Springer, 2014, pp. 536–551.

[45] G. BRADSKI, *The opencv library*, Dr. Dobb's Journal of Software Tools, (2000).

[46] R. BRÉGIER, F. DEVERNAY, L. LEYRIT, AND J. L. CROWLEY, *Defining the pose of any 3d rigid object and an associated distance*, International Journal of Computer Vision, 126 (2018), pp. 571–596.

[47] J. BRIALES AND J. GONZALEZ-JIMENEZ, *Fast global optimality verification in 3d slam*, in Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on, IEEE, 2016, pp. 4630–4636.

[48] ——, *Initialization of 3d pose graph optimization using lagrangian duality*, in Robotics and Automation (ICRA), 2017 IEEE International Conference on, IEEE, 2017, pp. 5134–5139.

[49] D. C. BROWN, *Close-range camera calibration*, PHOTOGRAMMETRIC ENGINEERING, 37 (1971), pp. 855–866.

[50] B. BUSAM, T. BIRDAL, AND N. NAVAB, *Camera pose filtering with local regression geodesics on the riemannian manifold of dual quaternions*, in 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Oct 2017, pp. 2436–2445.

[51] B. BUSAM, M. ESPOSITO, S. CHE'ROSE, N. NAVAB, AND B. FRISCH, *A stereo vision approach for cooperative robotic movement therapy*, in Proceedings of the IEEE International Conference on Computer Vision Workshops, ICCVW, 2015, pp. 127–135.

[52] B. BUSAM, M. ESPOSITO, B. FRISCH, AND N. NAVAB, *Quaternionic upsampling: Hyperspherical techniques for 6 dof pose tracking*, in 2016 Fourth International Conference on 3D Vision (3DV), Oct 2016, pp. 629–638.

[53] S. R. BUSS AND J. P. FILLMORE, *Spherical averages and applications to spherical splines and interpolation*, ACM Transactions on Graphics (TOG), 20 (2001), pp. 95–126.

[54] S. BYRNE AND M. GIROLAMI, *Geodesic monte carlo on embedded manifolds*, Scandinavian Journal of Statistics, 40 (2013), pp. 825–845.

[55] F. CALAKLI AND G. TAUBIN, *Ssd: Smooth signed distance surface reconstruction*, in Computer Graphics Forum, vol. 30, Wiley Online Library, 2011, pp. 1993–2002.

[56] D. R. CANELHAS, T. STOYANOV, AND A. J. LILIENTHAL, *Sdf tracker: A parallel algorithm for on-line pose estimation and scene reconstruction from depth images*, in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2013, pp. 3671–3676.

[57] L. CARLONE AND G. C. CALAFIORE, *Convex Relaxations for Pose Graph Optimization with Outliers*, ArXiv e-prints, (2018).

[58] L. CARLONE, R. TRON, K. DANIILIDIS, AND F. DELLAERT, *Initialization techniques for 3d slam: a survey on rotation estimation and its use in pose graph optimization*, in Robotics and Automation (ICRA), 2015 IEEE International Conference on, IEEE, 2015, pp. 4597–4604.

[59] A. CHATTERJEE AND V. M. GOVINDU, *Efficient and robust large-scale rotation averaging*, in Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE, 2013, pp. 521–528.

[60] ——, *Robust relative rotation averaging*, IEEE transactions on pattern analysis and machine intelligence, 40 (2018), pp. 958–972.

[61] C. CHEN, D. CARLSON, Z. GAN, C. LI, AND L. CARIN, *Bridging the gap between stochastic gradient MCMC and stochastic optimization*, in AISTATS, 2016.

[62] C. CHEN, N. DING, AND L. CARIN, *On the convergence of stochastic gradient MCMC algorithms with high-order integrators*, in Advances in Neural Information Processing Systems, 2015, pp. 2269–2277.

[63] H. CHEN AND B. BHANU, *3d free-form object recognition in range images using local surface patches*, Pattern Recognition Letters, 28 (2007), pp. 1252–1262.

[64] H. H. CHEN, *A screw motion approach to uniqueness analysis of head-eye geometry*, in Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Jun 1991, pp. 145–151.

[65] D. CHETVERIKOV, D. SVIRKO, D. STEPANOV, AND P. KRSEK, *The trimmed iterative closest point algorithm*, in Pattern Recognition, 2002. Proceedings. 16th International Conference on, vol. 3, IEEE, 2002, pp. 545–548.

[66] C. CHOI AND H. I. CHRISTENSEN, *3d pose estimation of daily objects using an rgb-d camera*, in Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, IEEE, 2012, pp. 3342–3349.

[67] C. CHOI, Y. TAGUCHI, O. TUZEL, M.-Y. LIU, AND S. RAMALINGAM, *Voting-based pose estimation for robotic assembly using a 3d sensor*, in Robotics and Automation (ICRA), 2012 IEEE International Conference on, IEEE, 2012, pp. 1724–1731.

[68] S. CHOI, Q.-Y. ZHOU, S. MILLER, AND V. KOLTUN, *A large dataset of object scans*, arXiv:1602.02481, (2016).

[69] M. CORSINI, P. CIGNONI, AND R. SCOPIGNO, *Efficient and flexible sampling with blue noise properties of triangular meshes*, Visualization and Computer Graphics, IEEE Transactions on, 18 (2012), pp. 914–924.

[70] C. CORTES AND V. VAPNIK, *Support-vector networks*, Machine learning, 20 (1995), pp. 273–297.

[71] G. CROSS AND A. ZISSERMAN, *Quadric reconstruction from dual-space geometry*, in International Conference on Computer Vision, 1998.

[72] B. CURLESS AND M. LEVOY, *A volumetric method for building complex models from range images*, in Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, ACM, 1996, pp. 303–312.

[73] S. DAFTRY, M. MAURER, A. WENDEL, AND H. BISCHOF, *Flexible and usercentric camera calibration using planar fiducial markers*, in in British Machine Vision Conference (BMVC, Citeseer, 2013.

[74] A. DAI, M. NIESSNER, M. ZOLLHÖFER, S. IZADI, AND C. THEOBALT, *Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface re-integration*, arXiv preprint arXiv:1604.01093, (2016).

[75] F. DAI AND Y. XU, *Approximation theory and harmonic analysis on spheres and balls*, Springer, 2013.

[76] M. DANELLJAN, G. MENEGHETTI, F. S. KHAN, AND M. FELSBERG, *A probabilistic framework for color-based point set registration*, in CVPR, vol. 1, 2016, p. 3.

[77] K. DANIILIDIS, *Hand-eye calibration using dual quaternions*, The International Journal of Robotics Research, 18 (1999), pp. 286–298.

[78] R. P. DE FIGUEIREDO, P. MORENO, AND A. BERNARDINO, *Fast 3d object recognition of rotationally symmetric objects*, in Pattern Recognition and Image Analysis, Springer, 2013, pp. 125–132.

[79] M. DEFFERRARD, X. BRESSON, AND P. VANDERGHEYNST, *Convolutional neural networks on graphs with fast localized spectral filtering*, in Advances in Neural Information Processing Systems, 2016, pp. 3844–3852.

[80] H. DENG, T. BIRDAL, AND S. ILIC, *Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors*, in Proceedings of the European Conference on Computer Vision (ECCV), 2018.

[81] ——, *Ppfnet: Global context aware local features for robust 3d point matching*, in Computer Vision and Pattern Recognition (CVPR), IEEE, 2018.

[82] P. DIACONIS, S. HOLMES, M. SHAHSHAHANI, ET AL., *Sampling from a manifold*, in Advances in Modern Statistical Theory and Applications: A Festschrift in honor of Morris L. Eaton, Institute of Mathematical Statistics, 2013, pp. 102–125.

[83] A. DOUIK AND B. HASSIBI, *Manifold Optimization Over the Set of Doubly Stochastic Matrices: A Second-Order Geometry*, ArXiv e-prints, (2018).

[84] B. DROST AND S. ILIC, *3d object detection and localization using multimodal point pair features*, in 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on, IEEE, 2012, pp. 9–16.

[85] ——, *Local hough transform for 3d primitive detection*, in 3D Vision (3DV), International Conference on, IEEE, 2015.

[86] B. DROST, M. ULRICH, P. BERGMANN, P. HARTINGER, AND C. STEGER, *Introducing mvtec itodd - a dataset for 3d object recognition in industry*, in The IEEE International Conference on Computer Vision (ICCV), Oct 2017.

[87] B. DROST, M. ULRICH, N. NAVAB, AND S. ILIC, *Model globally, match locally: Efficient and robust 3d object recognition*, in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 998–1005.

[88] B. H. DROST, *Point Cloud Computing for Rigid and Deformable 3D Object Recognition*, PhD thesis, Technische Universität München, 2016.

[89] B. ECKART, K. KIM, A. TROCCOLI, A. KELLY, AND J. KAUTZ, *Accelerated generative models for 3d point cloud data*, in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2016, pp. 5497–5505.

[90] A. ERIKSSON, C. OLSSON, K. FREDRIK, AND T.-J. CHIN, *Rotation averaging and strong duality*, in The IEEE Conference on Comptuter Vision and Pattern Recognition (CVPR), June 2018.

[91] G. D. EVANGELIDIS AND R. HORAUD, *Joint registration of multiple point sets*, CoRR, abs/1609.01466 (2016).

[92] S. FANTONI, U. CASTELLANI, AND A. FUSIELLO, *Accurate and automatic alignment of range surfaces*, in 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission, IEEE, 2012, pp. 73–80.

[93] O. D. FAUGERAS, *Three-Dimensional Computer Vision.*, The MIT Press, fourth printing ed., 2001.

[94] P. F. FELZENSZWALB AND D. P. HUTTENLOCHER, *Efficient graph-based image segmentation*, International Journal of Computer Vision, 59 (2004), pp. 167–181.

[95] M. FIALA, *Artag, a fiducial marker system using digital techniques*, in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 2, IEEE, 2005, pp. 590–596.

[96] M. A. FISCHLER AND R. C. BOLLES, *Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography*, Commun. ACM, (1981).

[97] A. W. FITZGIBBON, *Robust registration of 2d and 3d point sets*, Image and Vision Computing, 21 (2003), pp. 1145–1153.

[98] J.-M. FRAHM AND M. POLLEFEYS, *Ransac for (quasi-) degenerate data (qdegsac)*, in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 1, IEEE, 2006, pp. 453–460.

[99] J. FREDRIKSSON AND C. OLSSON, *Simultaneous multiple rotation averaging using lagrangian duality*, in Asian Conference on Computer Vision, Springer, 2012, pp. 245–258.

[100] J. GALLIER, *Notes on differential geometry and lie groups*, University of Pennsylvannia, (2012).

[101] S. GARRIDO-JURADO, R. MUÑOZ-SALINAS, F. J. MADRID-CUEVAS, AND M. J. MARÍN-JIMÉNEZ, *Automatic generation and detection of highly reliable fiducial markers under occlusion*, Pattern Recognition, 47 (2014), pp. 2280–2292.

[102] P. GAY, C. RUBINO, V. BANSAL, AND A. DEL BUE, *Probabilistic structure from motion with objects (psfmo)*, in The IEEE International Conference on Computer Vision (ICCV), Oct 2017.

[103] N. GELFAND, L. IKEMOTO, S. RUSINKIEWICZ, AND M. LEVOY, *Geometrically stable sampling for the icp algorithm*, in 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on, IEEE, 2003, pp. 260–267.

[104] S. B. GELFAND AND S. K. MITTER, *Recursive stochastic algorithms for global optimization in R^d*, SIAM Journal on Control and Optimization, 29 (1991), pp. 999–1018.

[105] K. GEORGIEV, M. AL-HAMI, AND R. LAKAEMPER, *Real-time 3d scene description using spheres, cones and cylinders*, arXiv preprint arXiv:1603.03856, (2016).

[106] D. GIRARDEAU-MONTAUT, *Cloudcompare - 3d point cloud and mesh processing software*.
`http://www.danielgm.net/cc/`, 2016.

[107] J. GLOVER, G. BRADSKI, AND R. B. RUSU, *Monte carlo pose estimation with quaternion kernels and the bingham distribution*, in Robotics: science and systems, vol. 7, 2012, p. 97.

[108] J. GLOVER AND L. P. KAELBLING, *Tracking the spin on a ping pong ball with the quaternion bingham filter*, in 2014 IEEE International Conference on Robotics and Automation (ICRA), May 2014, pp. 4133–4140.

[109] GOM, *Tritop: Gom*.
https://www.gom.com/metrology-systems/tritop.html, 2016.

[110] P. F. GOTARDO, K. L. BOYER, O. R. P. BELLON, AND L. SILVA, *Robust extraction of planar and quadric surfaces from range images*, in International Conference on Pattern Recognition (ICPR), IEEE, 2004.

[111] V. M. GOVINDU, *Combining two-view constraints for motion estimation*, in Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, vol. 2, IEEE, 2001, pp. II–II.

[112] ——, *Lie-algebraic averaging for globally consistent motion estimation*, in Computer Vision and Pattern Recognition (CVPR), vol. 1, IEEE, 2004, pp. I–I.

[113] ——, *Robustness in motion averaging*, in Asian Conference on Computer Vision, Springer, 2006, pp. 457–466.

[114] V. M. GOVINDU AND A. POOJA, *On averaging multiview relations for 3d scan registration*, IEEE Transactions on Image Processing, 23 (2014), pp. 1289–1302.

[115] F. S. GRASSIA, *Practical parameterization of rotations using the exponential map*, Journal of graphics tools, 3 (1998), pp. 29–48.

[116] F. GUNEY AND A. GEIGER, *Displets: Resolving stereo ambiguities using object knowledge*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4165–4175.

[117] Y. GUO, M. BENNAMOUN, F. SOHEL, M. LU, AND J. WAN, *3d object recognition in cluttered scenes with local surface features: A survey*, IEEE Transactions on Pattern Analysis & Machine Intelligence, (2014), pp. 2270–2287.

[118] R. HADSELL, S. CHOPRA, AND Y. LECUN, *Dimensionality reduction by learning an invariant mapping*, in Computer vision and pattern recognition, 2006 IEEE computer society conference on, vol. 2, IEEE, 2006, pp. 1735–1742.

[119] W. R. HAMILTON, *Xi. on quaternions; or on a new system of imaginaries in algebra*, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 33 (1848), pp. 58–60.

[120] R. HARTLEY, K. AFTAB, AND J. TRUMPF, *L1 rotation averaging using the weiszfeld algorithm*, in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 3041–3048.

[121] R. HARTLEY, J. TRUMPF, Y. DAI, AND H. LI, *Rotation averaging*, International journal of computer vision, 103 (2013), pp. 267–305.

[122] R. I. HARTLEY, *In defense of the eight-point algorithm*, IEEE Transactions on pattern analysis and machine intelligence, (1997).

[123] R. I. HARTLEY AND A. ZISSERMAN, *Multiple View Geometry in Computer Vision*, Cambridge University Press, ISBN: 0521540518, second ed., 2004.

[124] P. HÉBERT, É. SAINT-PIERRE, AND D. TUBIC, *Auto-referenced system and apparatus for three-dimensional scanning*, Mar. 22 2011.
US Patent 7,912,673.

[125] J. HEIKKILA, *Geometric camera calibration using circular control points*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22 (2000), pp. 1066–1077.

[126] L. HENG, B. LI, AND M. POLLEFEYS, *Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry*, in Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, IEEE, 2013, pp. 1793–1800.

[127] G. HERMANN, *Geometric error correction in coordinate measurement*, in Acta Polytechnica Hungarica, vol. 4, 2007, pp. 47–62.

[128] S. HINTERSTOISSER, V. LEPETIT, S. ILIC, S. HOLZER, G. BRADSKI, K. KONOLIGE, AND N. NAVAB, *Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes*, in Computer Vision–ACCV 2012, Springer, 2013, pp. 548–562.

[129] S. HINTERSTOISSER, V. LEPETIT, N. RAJKUMAR, AND K. KONOLIGE, *Going further with point pair features*, in European Conference on Computer Vision, Springer, 2016, pp. 834–848.

[130] T. HODAN, F. MICHEL, E. BRACHMANN, W. KEHL, A. G. BUCH, D. KRAFT, B. DROST, J. VIDAL, S. IHRKE, X. ZABULIS, ET AL., *Bop: Benchmark for 6d object pose estimation*, arXiv preprint arXiv:1808.08319, (2018).

[131] D. HOLZ, M. NIEUWENHUISEN, D. DROESCHEL, J. STÜCKLER, A. BERNER, J. LI, R. KLEIN, AND S. BEHNKE, *Active recognition and manipulation for mobile robot bin picking*, in Gearing Up and Accelerating Cross-fertilization between Academic and Industrial Robotics Research in Europe:, Springer, 2014, pp. 133–153.

[132] H. HOPPE, T. DEROSE, T. DUCHAMP, J. MCDONALD, AND W. STUETZLE, *Surface reconstruction from unorganized points*, vol. 26.2, ACM, 1992.

[133] B. K. HORN, *Closed-form solution of absolute orientation using unit quaternions*, JOSA A, 4 (1987), pp. 629–642.

[134] T. HUANG, *Computer vision: Evolution and promise*, (1996).

[135] D. F. HUBER AND M. HEBERT, *Fully automatic registration of multiple 3d data sets*, Image and Vision Computing, 21 (2003), pp. 637–650.

[136] D. HUYNH, *The cross ratio: A revisit to its probability density function*, in Proceedings of the British Machine Conference, pages, 2000, pp. 27–1.

[137] C. HWANG, *Laplace's method revisited: weak convergence of probability measures*, The Annals of Probability, (1980), pp. 1177–1182.

[138] Y. IOANNOU, B. TAATI, R. HARRAP, AND M. GREENSPAN, *Difference of normals as a multi-scale operator in unorganized point clouds*, in 3DIMPVT, 2012, International Conference on, IEEE, 2012.

[139] O. H. JAFARI, S. K. MUSTIKOVELA, K. PERTSCH, E. BRACHMANN, AND C. ROTHER, *ipose: Instance-aware 6d pose estimation of partly occluded objects*, arXiv preprint arXiv:1712.01924, (2017).

[140] A. E. JOHNSON AND M. HEBERT, *Using spin images for efficient object recognition in cluttered 3d scenes*, IEEE Transactions on pattern analysis and machine intelligence, 21 (1999), pp. 433–449.

[141] K. JYRKINEN, M. OLLIKAINEN, V. KYRKI, J. P. VARIS, AND H. KÄLVIÄINEN, *Optical 3d measurement in the quality assurance of formed sheet metal parts*, in

Proceedings of International Conference on Pattern Recognition, vol. 120, 2003, p. 12.

[142] F. KAHLESZ, C. LILGE, AND R. KLEIN, *Easy-to-use calibration of multiple-camera setups*, in Workshop on Camera Calibration Methods for Computer Vision Systems (CCMVS2007), Mar. 2007.

[143] K.-I. KANATANI, *Further improving geometric fitting*, in International Conference on 3-D Digital Imaging and Modeling, IEEE, 2005.

[144] D. KARP AND S. SITNIK, *Inequalities and monotonicity of ratios for generalized hypergeometric function*, Journal of Approximation Theory, 161 (2009), pp. 337–352.

[145] H. KATO AND M. BILLINGHURST, *Marker tracking and hmd calibration for a video-based augmented reality conferencing system*, in Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on, IEEE, 1999, pp. 85–94.

[146] L. KAVAN, S. COLLINS, C. O'SULLIVAN, AND J. ZARA, *Dual quaternions for rigid transformation blending*, Trinity College Dublin, Tech. Rep. TCD-CS-2006-46, (2006).

[147] W. KEHL, F. MANHARDT, F. TOMBARI, S. ILIC, AND N. NAVAB, *Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again*, in Proceedings of the International Conference on Computer Vision (ICCV 2017), Venice, Italy, 2017, pp. 22–29.

[148] W. KEHL, N. NAVAB, AND S. ILIC, *Coloured signed distance fields for full 3d object reconstruction*, in British Machine Vision Conference, 2014.

[149] B. KENWRIGTH, *Inverse kinematics with dual-quaternions, exponential-maps, and joint limits*, International Journal on Advances in Intelligent Systems, 6 (2013).

[150] M. KHOURY, Q.-Y. ZHOU, AND V. KOLTUN, *Learning compact geometric features*, in The IEEE International Conference on Computer Vision (ICCV), Oct 2017.

[151] L. KIFORENKO, B. DROST, F. TOMBARI, N. KRÜGER, AND A. G. BUCH, *A performance evaluation of point pair features*, Computer Vision and Image Understanding, (2017).

[152] E. KIM AND G. MEDIONI, *3d object recognition in range images using visibility context*, in Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, IEEE, 2011, pp. 3800–3807.

[153] A. KNAPITSCH, J. PARK, Q.-Y. ZHOU, AND V. KOLTUN, *Tanks and temples: Benchmarking large-scale scene reconstruction*, ACM Transactions on Graphics (TOG), 36 (2017), p. 78.

[154] S. KRISHNAN, P. Y. LEE, J. B. MOORE, S. VENKATASUBRAMANIAN, ET AL., *Global registration of multiple 3d point sets via optimization-on-a-manifold.*, in Symposium on Geometry Processing, 2005, pp. 187–196.

[155] A. KRULL, E. BRACHMANN, S. NOWOZIN, F. MICHEL, J. SHOTTON, AND C. ROTHER, *Poseagent: Budget-constrained 6d object pose estimation via reinforcement learning*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, 2017.

[156] Z. KUKELOVA, J. HELLER, AND A. FITZGIBBON, *Efficient intersection of three quadrics and applications in computer vision*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[157] A. KUME AND A. T. WOOD, *On the derivatives of the normalising constant of the bingham distribution*, Statistics & probability letters, 77 (2007), pp. 832–837.

[158] R. KÜMMERLE, G. GRISETTI, H. STRASDAT, K. KONOLIGE, AND W. BURGARD, *g 2 o: A general framework for graph optimization*, in Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE, 2011, pp. 3607–3613.

[159] G. KURZ, I. GILITSCHENSKI, S. JULIER, AND U. D. HANEBECK, *Recursive estimation of orientation based on the bingham distribution*, in Information Fusion (FUSION), 2013 16th International Conference on, IEEE, 2013, pp. 1487–1494.

[160] A. LAGAE AND P. DUTRÉ, *A comparison of methods for generating poisson disk distributions*, in Computer Graphics Forum, 2008.

[161] J. LAM AND M. GREENSPAN, *3d object recognition by surface registration of interest segments*, in 3D Vision-3DV 2013, 2013 International Conference on, IEEE, 2013, pp. 199–206.

[162] B. LEIMKUHLER AND C. MATTHEWS, *Molecular Dynamics: With Deterministic and Stochastic Numerical Methods*, vol. 39, Springer, 2015.

[163] V. LEPETIT, P. FUA, ET AL., *Monocular model-based 3d tracking of rigid objects: A survey*, Foundations and Trends® in Computer Graphics and Vision, 1 (2005), pp. 1–89.

[164] K. LEVENBERG, *A method for the solution of certain non-linear problems in least squares*, Quarterly of applied mathematics, 2 (1944), pp. 164–168.

[165] B. LÉVY AND Y. LIU, *L p centroidal voronoi tessellation and its applications*, in ACM Transactions on Graphics (TOG), 2010.

[166] Y. LI, X. WU, Y. CHRYSATHOU, A. SHARF, D. COHEN-OR, AND N. J. MITRA, *Globfit: Consistently fitting primitives by discovering global relations*, in ACM Transactions on Graphics (TOG), 2011.

[167] X. LIAN, Y. HUANG, Y. LI, AND J. LIU, *Asynchronous parallel stochastic gradient for nonconvex optimization*, in Advances in Neural Information Processing Systems, 2015, pp. 2737–2745.

[168] J. LIEBELT AND C. SCHMID, *Multi-view object class detection with a 3d geometric model*, in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 1688–1695.

[169] C. LIU, J. ZHU, AND Y. SONG, *Stochastic gradient geodesic mcmc methods*, in Advances in Neural Information Processing Systems, 2016, pp. 3009–3017.

[170] J.-S. LIU AND J.-H. CHUANG, *A geometry-based error estimation for cross-ratios*, Pattern recognition, 35 (2002), pp. 155–167.

[171] M.-Y. LIU, O. TUZEL, A. VEERARAGHAVAN, Y. TAGUCHI, T. K. MARKS, AND R. CHELLAPPA, *Fast object localization and pose estimation in heavy clutter for robotic bin picking*, The International Journal of Robotics Research, 31 (2012), pp. 951–973.

[172] Y. LIU, *Automatic 3d free form shape matching using the graduated assignment algorithm*, Pattern Recognition, 38 (2005), pp. 1615–1631.

[173] Y. LIU, *Robust segmentation of raw point clouds into consistent surfaces*, Science China Technological Sciences, (2016).

[174] Y. LIU, F. SHANG, J. CHENG, H. CHENG, AND L. JIAO, *Accelerated first-order methods for geodesically convex optimization on Riemannian manifolds*, in Advances in Neural Information Processing Systems, 2017, pp. 4875–4884.

[175] E. LÓPEZ-RUBIO, K. THURNHOFER-HEMSI, Ó. D. DE CÓZAR-MACÍAS, E. B. BLÁZQUEZ-PARRA, J. MUÑOZ-PÉREZ, AND I. L. DE GUEVARA-LÓPEZ, *Robust fitting of ellipsoids by separating interior and exterior points during optimization*, Journal of Mathematical Imaging and Vision, (2016).

[176] W. E. LORENSEN AND H. E. CLINE, *Marching cubes: A high resolution 3d surface construction algorithm*, in ACM siggraph computer graphics, vol. 21, ACM, 1987, pp. 163–169.

[177] R. LOSER, T. LUHMANN, AND L. H. PMU, *The programmable optical 3d measuring system pom-applications and performance*, International Archives of Photogrammetry and Remote Sensing, 29 (1993), pp. 533–533.

[178] K.-L. LOW, *Linear least-squares optimization for point-to-plane icp surface registration*, tech. rep., University of North Carolina, 2004.

[179] T. LUHMANN, *Close range photogrammetry for industrial applications*, ISPRS Journal of Photogrammetry and Remote Sensing, 65 (2010), pp. 558–569.

[180] T. LUHMANN, F. BETHMANN, B. HERD, AND J. OHM, *Comparison and verification of optical 3-d surface measurement systems*, The international archives of the photogrammetry, remote sensing and spatial information sciences, 37 (2008), pp. 51–56.

[181] Y. A. MA, T. CHEN, AND E. FOX, *A complete recipe for stochastic gradient MCMC*, in Advances in Neural Information Processing Systems, 2015, pp. 2899–2907.

[182] T. MAEKAWA, Y. YANAGISAWA, Y. KISHINO, K. ISHIGURO, K. KAMEI, Y. SAKURAI, AND T. OKADOME, *Object-based activity recognition with heterogeneous sensors on wrist*, in International Conference on Pervasive Computing, Springer, 2010, pp. 246–264.

[183] A. MAKHAL, F. THOMAS, AND A. P. GRACIA, *Grasping unknown objects in clutter by superquadric representation*, arXiv preprint arXiv:1710.02121, (2017).

[184] S. MALASSIOTIS AND M. STRINTZIS, *Stereo vision system for precision dimensional inspection of 3d holes*, Machine Vision and Applications, 15 (2003), pp. 101–113.

[185] F. L. MARKLEY, Y. CHENG, J. L. CRASSIDIS, AND Y. OSHMAN, *Averaging quaternions*, Journal of Guidance, Control, and Dynamics, 30 (2007), pp. 1193–1197.

[186] E. MASET, F. ARRIGONI, AND A. FUSIELLO, *Practical and efficient multi-view matching*, in 2017 IEEE International Conference on Computer Vision (ICCV), Oct 2017, pp. 4578–4586.

[187] MATHWORKS, *Documentation of vision.kalmanfilter class.*, 2016.

[188] J. MCCORMAC, R. CLARK, M. BLOESCH, A. J. DAVISON, AND S. LEUTENEGGER, *Fusion++: Volumetric object-level slam*, arXiv preprint arXiv:1808.08378, (2018).

[189] P. MEER, *Robust techniques for computer vision*, in Emerging topics in computer vision, 2004, pp. 107–190.

[190] P. MEER, R. LENZ, AND S. RAMAKRISHNA, *Efficient invariant representations*, International Journal of Computer Vision, 26 (1998), pp. 137–152.

[191] P. MEER, S. RAMAKRISHNA, AND R. LENZ, *Correspondence of coplanar features through p2-invariant representations*, in Joint European-US Workshop on Applications of Invariance in Computer Vision, Springer, 1993, pp. 473–492.

[192] A. S. MIAN, M. BENNAMOUN, AND R. OWENS, *Three-dimensional model-based object recognition and segmentation in cluttered scenes*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 28 (2006), pp. 1584–1601.

[193] A. S. MIAN, M. BENNAMOUN, AND R. OWENS, *Three-dimensional model-based object recognition and segmentation in cluttered scenes*, IEEE Trans. Pattern Anal. Mach. Intell., 28 (2006).

[194] G. MILLER, *Efficient algorithms for local and global accessibility shading*, in Proceedings of the 21st annual conference on Computer graphics and interactive techniques, ACM, 1994, pp. 319–326.

[195] J. R. MILLER, *Analysis of quadric-surface-based solid models*, IEEE Computer Graphics and Applications, 8 (1988), pp. 28–42.

[196] A. F. MÖBIUS, *Der barycentrische calcul*, JA Barth, 1827.

[197] M. MOHAMAD, D. RAPPAPORT, AND M. GREENSPAN, *Generalized 4-points congruent sets for 3d registration*, in 3D Vision (3DV), 2014 2nd International Conference on, vol. 1, Dec 2014, pp. 83–90.

[198] T. MÖLLER AND B. TRUMBORE, *Fast, minimum storage ray/triangle intersection*, in ACM SIGGRAPH Courses, ACM, 2005, p. 7.

[199] A. MORAWIEC AND D. FIELD, *Rodrigues parameterization for orientation and misorientation distributions*, Philosophical Magazine A, 73 (1996), pp. 1113–1130.

[200] T. MÖRWALD, A. RICHTSFELD, J. PRANKL, M. ZILLICH, AND M. VINCZE, *Geometric data abstraction using b-splines for range image segmentation*, in Robotics and Automation, International Conference on, IEEE, 2013.

[201] N. MOSTOFI, F. SAMADZADEGAN, S. ROOHY, AND M. NOZARI, *Using vision metrology system for quality control in automotive industries*, ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 1 (2012), pp. 33–37.

[202] M. MUJA AND D. G. LOWE, *Scalable nearest neighbor algorithms for high dimensional data*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 36 (2014).

[203] R. M. MURRAY, *A mathematical introduction to robotic manipulation*, CRC press, 1994.

[204] K. MUSETH, J. LAIT, J. JOHANSON, J. BUDSBERG, R. HENDERSON, M. ALDEN, P. CUCKA, D. HILL, AND A. PEARCE, *Openvdb: an open-source data structure and toolkit for high-resolution volumes*, in ACM, 2013.

[205] L. NAIMARK AND E. FOXLIN, *Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker*, in Mixed and Augmented Reality, 2002. ISMAR 2002. Proceedings. International Symposium on, 2002, pp. 27–36.

[206] T. NAKAI, K. KISE, AND M. IWAMURA, *Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval*, in International Workshop on Document Analysis Systems, Springer, 2006, pp. 541–552.

[207] Y. NESTEROV, *Introductory lectures on convex optimization: A basic course*, vol. 87, Springer Science & Business Media, 2013.

[208] R. A. NEWCOMBE, S. IZADI, O. HILLIGES, D. MOLYNEAUX, D. KIM, A. J. DAVISON, P. KOHI, J. SHOTTON, S. HODGES, AND A. FITZGIBBON, *Kinectfusion: Real-time dense surface mapping and tracking*, in Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on, IEEE, 2011, pp. 127–136.

[209] M. NIESSNER, M. ZOLLHÖFER, S. IZADI, AND M. STAMMINGER, *Real-time 3d reconstruction at scale using voxel hashing*, ACM Transactions on Graphics (TOG), (2013).

[210] M. NIEUWENHUISEN, D. DROESCHEL, D. HOLZ, J. STUCKLER, A. BERNER, J. LI, R. KLEIN, AND S. BEHNKE, *Mobile bin picking with an anthropomorphic*

service robot, in Robotics and Automation (ICRA), 2013 IEEE International Conference on, IEEE, 2013, pp. 2327–2334.

[211] H. NOH, A. ARAUJO, J. SIM, T. WEYAND, AND B. HAN, *Large-scale image retrieval with attentive deep local features*, in The IEEE International Conference on Computer Vision (ICCV), Oct 2017.

[212] J. NOVATNACK AND K. NISHINO, *Scale-dependent/invariant local 3d shape descriptors for fully automatic registration of multiple sets of range images*, in Computer Vision–ECCV 2008, Springer, 2008, pp. 440–453.

[213] E. OLSON, *Apriltag: A robust and flexible visual fiducial system*, in Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE, 2011, pp. 3400–3407.

[214] C. PAPAZOV AND D. BURSCHKA, *An efficient ransac for 3d object recognition in noisy and occluded scenes*, in ACCV, 2010.

[215] H.-M. PARK AND K.-J. YOON, *Multi-layer random walks synchronization for multi-attributed multiple graph matching*, arXiv preprint arXiv:1712.02575, (2017).

[216] A. T. PAS AND R. PLATT, *Localizing grasp affordances in 3-d points clouds using taubin quadric fitting*, arXiv preprint arXiv:1311.3192, (2013).

[217] S. PETITJEAN, *A survey of methods for recovering quadrics in triangle meshes*, ACM Computing Surveys (CSUR), (2002).

[218] M.-T. PHAM, O. J. WOODFORD, F. PERBET, A. MAKI, B. STENGER, AND R. CIPOLLA, *A new distance for scale-invariant 3D shape recognition and registration*, in ICCV, 2011.

[219] S. PHILLIPS, B. BORCHARDT, A. ABACKERLI, C. SHAKARJI, D. SAWYER, P. MURRAY, B. RASNICK, K. SUMMERHAYS, J. BALDWIN, R. HENKE, ET AL., *The validation of cmm task specific measurement uncertainty software*, in Proc. of the ASPE 2003 summer topical meeting "Coordinate Measuring Machines, 2003.

[220] S. M. PRAKHYA, B. LIU, AND W. LIN, *Detecting keypoint sets on 3d point clouds via histogram of normal orientations*, Pattern Recognition Letters, 83 (2016).

[221] C. R. QI, H. SU, K. MO, AND L. J. GUIBAS, *Pointnet: Deep learning on point sets for 3d classification and segmentation*, Proc. Computer Vision and Pattern Recognition (CVPR), IEEE, 1 (2017), p. 4.

[222] K. QIN, *Representing quadric surfaces using nurbs surfaces*, Journal of Computer Science and Technology, 12 (1997).

[223] K. QIN, W. WANG, AND Z. TANG, *Representing spheres and ellipsoids using periodic nurbs surfaces with fewer control vertices*, in Pacific Graphics. Pacific Conference, IEEE, 1998.

[224] R. QIU, Q.-Y. ZHOU, AND U. NEUMANN, *Pipe-run extraction and reconstruction from point clouds*, in ECCV, Springer, 2014.

[225] M. RAD AND V. LEPETIT, *Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth*, in International Conference on Computer Vision, vol. 1, 2017, p. 5.

[226] M. RAGINSKY, A. RAKHLIN, AND M. TELGARSKY, *Non-convex learning via stochastic gradient Langevin dynamics: a nonasymptotic analysis*, in Proceedings of the 2017 Conference on Learning Theory, vol. 65, 2017, pp. 1674–1703.

[227] G. RIEGLER, O. ULUSOY, AND A. GEIGER, *Octnet: Learning deep 3d representations at high resolutions*, in IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), July 2017.

[228] E. RODOLÀ, A. ALBARELLI, D. CREMERS, AND A. TORSELLO, *A simple and effective relevance-based point sampling for 3d shapes*, Pattern Recognition Letters, 59 (2015), pp. 41–47.

[229] M. RODRIGUES, *Invariants for Pattern Recognition and Classification*, Series in machine perception and artificial intelligence, World Scientific Publishing Company Pte Limited, 2000.

[230] D. M. ROSEN, L. CARLONE, A. S. BANDEIRA, AND J. J. LEONARD, *Se-sync: A certifiably correct algorithm for synchronization over the special euclidean group*, arXiv preprint arXiv:1612.07386, (2016).

[231] S. RUSINKIEWICZ AND M. LEVOY, *Efficient variants of the icp algorithm*, in 3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on, IEEE, 2001, pp. 145–152.

[232] R. B. RUSU, N. BLODOW, AND M. BEETZ, *Fast point feature histograms (fpfh) for 3d registration*, in Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, IEEE, 2009, pp. 3212–3217.

[233] R. B. RUSU AND S. COUSINS, *3D is here: Point Cloud Library (PCL)*, in IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, May 9-13 2011.

[234] R. F. SALAS-MORENO, R. A. NEWCOMBE, H. STRASDAT, P. H. KELLY, AND A. J. DAVISON, *Slam++: Simultaneous localisation and mapping at the level of objects*, in Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE, 2013, pp. 1352–1359.

[235] S. SALTI, F. TOMBARI, AND L. DI STEFANO, *Shot: Unique signatures of histograms for surface and texture description*, Computer Vision and Image Understanding, 125 (2014), pp. 251–264.

[236] S. SARTORI AND G. ZHANG, *Geometric error measurement and compensation of machines*, in CIRP Annals-Manufacturing Technology, vol. 44, 1995, pp. 599 – 609.

[237] S. SAVARESE AND L. FEI-FEI, *Multi-view object categorization and pose estimation*, in Computer Vision, Springer, 2010, pp. 205–231.

[238] M. SCHIAVINATO AND A. TORSELLO, *Synchronization over the birkhoff polytope for multi-graph matching*, in International Workshop on Graph-Based Representations in Pattern Recognition, Springer, 2017, pp. 266–275.

[239] J. SCHICK AND W. BOESEMANN, *Method and system for three-dimensional spatial position detection of surface points*, Jan. 16 2001.
US Patent 6,175,647.

[240] R. SCHNABEL, R. WAHL, AND R. KLEIN, *Efficient ransac for point-cloud shape detection*, in Computer graphics forum, 2007.

[241] C.-T. SCHNEIDER AND K. SINNREICH, *Optical 3-d measurement systems for quality control in industry*, International Archives of Photogrammetry and Remote Sensing, 29 (1993), pp. 56–56.

[242] J. L. SCHÖNBERGER AND J.-M. FRAHM, *Structure-from-motion revisited*, in Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[243] J. L. SCHÖNBERGER, E. ZHENG, M. POLLEFEYS, AND J.-M. FRAHM, *Pixelwise view selection for unstructured multi-view stereo*, in European Conference on Computer Vision (ECCV), 2016.

[244] P. SCOVANNER, S. ALI, AND M. SHAH, *A 3-dimensional sift descriptor and its application to action recognition*, in Proceedings of the 15th international conference on Multimedia, ACM, 2007, pp. 357–360.

[245] J. SELIG, *Exponential and cayley maps for dual quaternions*, Advances in applied Clifford algebras, 20 (2010), pp. 923–936.

[246] J. SERAFIN, E. OLSON, AND G. GRISETTI, *Fast and robust 3d feature extraction from sparse point clouds*, in Intelligent Robots and Systems, International Conference on, IEEE, 2016.

[247] S. A. A. SHAH, M. BENNAMOUN, F. BOUSSAID, AND A. A. EL-SALLAM, *A novel local surface description for automatic 3d object recognition in low resolution cluttered scenes*, in Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on, IEEE, 2013, pp. 638–643.

[248] J. SHOTTON, B. GLOCKER, C. ZACH, S. IZADI, A. CRIMINISI, AND A. FITZGIBBON, *Scene coordinate regression forests for camera relocalization in rgb-d images*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2930–2937.

[249] M. SLAVCHEVA, W. KEHL, N. NAVAB, AND S. ILIC, *SDF-2-SDF: Highly Accurate 3D Object Reconstruction*, in European Conference on Computer Vision (ECCV), 2016.

[250] S. SRA AND D. KARP, *The multivariate Watson distribution: Maximum-likelihood estimation and other aspects*, Journal of Multivariate Analysis, 114 (2013), pp. 256–269.

[251] N. E. STEENROD, *The topology of fibre bundles*, vol. 14, Princeton University Press, 1951.

[252] C. STEGER, *An unbiased detector of curvilinear structures*, IEEE Trans. Pattern Anal. Mach. Intell., 20 (1998), pp. 113–125.

[253] F. STEINBRUECKER, J. STURM, AND D. CREMERS, *Volumetric 3d mapping in real-time on a cpu*, Hongkong, China, 2014.

[254] C. STRECHA, W. VON HANSEN, L. VAN GOOL, P. FUA, AND U. THOENNESSEN, *On benchmarking camera calibration and multi-view stereo for high resolution imagery*, in Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, Ieee, 2008, pp. 1–8.

[255] H. SU, S. MAJI, E. KALOGERAKIS, AND E. LEARNED-MILLER, *Multi-view convolutional neural networks for 3d shape recognition*, in Proceedings of the IEEE international conference on computer vision, 2015, pp. 945–953.

[256] A. SVEIER, A. L. KLEPPE, L. TINGELSTAD, AND O. EGELAND, *Object detection in point clouds using conformal geometric algebra*, Advances in Applied Clifford Algebras, (2017), pp. 1–16.

[257] K. TAKAYAMA, A. JACOBSON, L. KAVAN, AND O. SORKINE-HORNUNG, *A simple method for correcting facet orientations in polygon meshes based on ray casting*, Journal of Computer Graphics Techniques, 3 (2014).

[258] A. TAMRAKAR AND B. KIMIA, *No grouping left behind: From edges to curve fragments*, in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, Oct 2007, pp. 1–8.

[259] Y. TANG AND J. FENG, *Hierarchical Multiview Rigid Registration*, Computer Graphics Forum, (2015).

[260] T. TASDIZEN, *Robust and repeatable fitting of implicit polynomial curves to point data sets and to intensity images*, PhD thesis, Brown University, 2001.

[261] T. TASDIZEN, J.-P. TAREL, AND D. B. COOPER, *Algebraic curves that work better*, in Computer Vision and Pattern Recognition (CVPR), 1999.

[262] K. TATENO, F. TOMBARI, AND N. NAVAB, *When 2.5 d is not enough: Simultaneous reconstruction, segmentation and recognition on dense slam*, in Robotics and Automation (ICRA), 2016 IEEE International Conference on, IEEE, 2016, pp. 2295–2302.

[263] G. TAUBIN, *Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 13 (1991), pp. 1115–1138.

[264] ——, *Smooth signed distance surface reconstruction and applications*, in Iberoamerican Congress on Pattern Recognition, Springer, 2012, pp. 38–45.

[265] A. TEJANI, D. TANG, R. KOUSKOURIDAS, AND T.-K. KIM, *Latent-class hough forests for 3d object detection and pose estimation*, in Computer Vision – ECCV 2014, vol. 8694 of Lecture Notes in Computer Science, Springer International Publishing, 2014, pp. 462–477.

[266] R. TOLDO, A. BEINAT, AND F. CROSILLA, *Global registration of multiple point clouds embedding the generalized procrustes analysis into an icp framework*, in 3DPVT 2010 Conference, 2010.

[267] F. TOMBARI, S. SALTI, AND L. DI STEFANO, *Unique signatures of histograms for local surface description*, in European conference on computer vision, Springer, 2010, pp. 356–369.

[268] ——, *Performance evaluation of 3d keypoint detectors*, International Journal of Computer Vision, 102 (2013), pp. 198–220.

[269] A. TORSELLO, E. RODOLA, AND A. ALBARELLI, *Multiview registration via graph diffusion of dual quaternions*, in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 2441–2448.

[270] ——, *Sampling relevant points for surface registration*, in 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on, IEEE, 2011, pp. 290–295.

[271] T.-T. TRAN, V.-T. CAO, AND D. LAURENDEAU, *Extraction of reliable primitives from unorganized point clouds*, 3D Research, (2015).

[272] B. TRIGGS, P. F. MCLAUCHLAN, R. I. HARTLEY, AND A. W. FITZGIBBON, *Bundle adjustment—a modern synthesis*, in International workshop on vision algorithms, Springer, 1999, pp. 298–372.

[273] R. TRON AND K. DANIILIDIS, *Statistical pose averaging with non-isotropic and incomplete relative measurements*, in European Conference on Computer Vision, Springer, 2014, pp. 804–819.

[274] R. TRON, X. ZHOU, AND K. DANIILIDIS, *A survey on rotation optimization in structure from motion*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016, pp. 77–85.

[275] R. Y. TSAI, *A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses*, Robotics and Automation, IEEE Journal of, 3 (1987), pp. 323–344.

[276] V. TUOMINEN, *Cost modeling of inspection strategies in automotive quality control*, Engineering Management Research, 1 (2012), p. p33.

[277] O. TUZEL, M.-Y. LIU, Y. TAGUCHI, AND A. RAGHUNATHAN, *Learning to rank 3d features*, in Computer Vision–ECCV 2014, Springer, 2014, pp. 520–535.

[278] B. TZEN, T. LIANG, AND M. RAGINSKY, *Local optimality and generalization guarantees for the langevin algorithm via empirical metastability*, in Conference on Learning Theory, 2018.

[279] H. UCHIYAMA AND H. SAITO, *Random dot markers*, in 2011 IEEE Virtual Reality Conference, IEEE, 2011, pp. 35–38.

[280] V. UFFENKAMP, *State of the art of high precision industrial photogrammetry*, in Third international workshop on accelerator alignment., 1993.

[281] M. ULRICH, C. WIEDEMANN, AND C. STEGER, *Combining scale-space and similarity-based aspect graphs for fast 3d object recognition*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 34 (2012), pp. 1902–1914.

[282] S. UTO, T. TSUJI, K. HARADA, R. KURAZUME, AND T. HASEGAWA, *Grasp planning using quadric surface approximation for parallel grippers*, in International Conference on Robotics and Biomimetics (ROBIO), 2013.

[283] N. VAN GESTEL, *Determining Measurement Uncertainties of Feature Measurements on CMMs (Bepalen van meetonzekerheden bij het meten van vormelementen met CMMs)*, KU Leuven, 2011.

[284] N. VASKEVICIUS, K. PATHAK, R. PASCANU, AND A. BIRK, *Extraction of quadrics from noisy point-clouds using a sensor noise model*, in Robotics and Automation, International Conference on, IEEE, 2010.

[285] E. VINCENT AND R. LAGANIERE, *Matching with epipolar gradient features and edge transfer*, in Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on, vol. 1, IEEE, 2003, pp. I–277.

[286] M. VISSER, S. STRAMIGIOLI, AND C. HEEMSKERK, *Cayley-hamilton for roboticists*, in Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, IEEE, 2006, pp. 4187–4192.

[287] J. VORSATZ, C. RÖSSL, L. P. KOBBELT, AND H.-P. SEIDEL, *Feature sensitive remeshing*, in Computer Graphics Forum, 2001.

[288] E. W. WEISSTEIN, *Quadratic surface*.
MathWorld - A Wolfram Web Resource, 2017.

[289] G. WELCH AND G. BISHOP, *An introduction to the kalman filter*, tech. rep., University of North Carolina at Chapel Hill, Department of Computer Science, 1995.

[290] K. WILSON, D. BINDEL, AND N. SNAVELY, *When is rotations averaging hard?*, in European Conference on Computer Vision, Springer, 2016, pp. 255–270.

[291] K. WILSON AND N. SNAVELY, *Robust global translations with 1dsfm*, in Proceedings of the European Conference on Computer Vision (ECCV), 2014.

[292] H. J. WOLFSON AND I. RIGOUTSOS, *Geometric hashing: An overview*, IEEE computational science and engineering, 4 (1997), pp. 10–21.

[293] S. WOOP, L. FENG, I. WALD, AND C. BENTHIN, *Embree ray tracing kernels for cpus and the xeon phi architecture*, in ACM SIGGRAPH Talks, ACM, 2013.

[294] C. WU ET AL., *Visualsfm: A visual structure from motion system*, (2011).

[295] Y. XIANG, T. SCHMIDT, V. NARAYANAN, AND D. FOX, *Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes*, arXiv preprint arXiv:1711.00199, (2017).

[296] J. XIAO, A. OWENS, AND A. TORRALBA, *Sun3d: A database of big spaces reconstructed using sfm and object labels*, in Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1625–1632.

[297] D.-M. YAN, J.-W. GUO, B. WANG, X.-P. ZHANG, AND P. WONKA, *A survey of blue-noise sampling and its applications*, Journal of Computer Science and Technology, (2015).

[298] D.-M. YAN, B. LÉVY, Y. LIU, F. SUN, AND W. WANG, *Isotropic remeshing with fast and exact computation of restricted voronoi diagram*, in Computer graphics forum, vol. 28, 2009.

[299] D.-M. YAN, Y. LIU, AND W. WANG, *Quadric surface extraction by variational shape approximation*, in International Conference on Geometric Modeling and Processing, Springer, 2006, pp. 73–86.

[300] D.-M. YAN, W. WANG, Y. LIU, AND Z. YANG, *Variational mesh segmentation via quadric surface fitting*, Computer-Aided Design, (2012).

[301] P. YAN, S. M. KHAN, AND M. SHAH, *3d model based object class detection in an arbitrary view*, in 2007 IEEE 11th International Conference on Computer Vision, IEEE, 2007, pp. 1–6.

[302] J. YANG, H. LI, AND Y. JIA, *Go-icp: Solving 3d registration efficiently and globally optimally*, in Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1457–1464.

[303] Y. YANG, C. FENG, Y. SHEN, AND D. TIAN, *Foldingnet: Point cloud auto-encoder via deep grid deformation*, in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[304] K. M. YI, E. TRULLS, V. LEPETIT, AND P. FUA, *Lift: Learned invariant feature transform*, in European Conference on Computer Vision, Springer, 2016.

[305] S. YINGZE BAO, M. CHANDRAKER, Y. LIN, AND S. SAVARESE, *Dense object reconstruction with semantic priors*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 1264–1271.

[306] S. YOU AND D. ZHANG, *Think locally, fit globally: Robust and fast 3d shape matching via adaptive algebraic fitting*, Neurocomputing, (2017).

[307] A. ZAHARESCU, E. BOYER, K. VARANASI, AND R. HORAUD, *Surface feature detection and description with applications to mesh matching*, in Computer Vision and Pattern Recognition, CVPR IEEE, IEEE, 2009, pp. 373–380.

[308] A. ZENG, S. SONG, M. NIESSNER, M. FISHER, J. XIAO, AND T. FUNKHOUSER, *3dmatch: Learning local geometric descriptors from rgb-d reconstructions*, in CVPR, 2017.

[309] H. ZHANG AND S. SRA, *First-order methods for geodesically convex optimization*, in Conference on Learning Theory, 2016, pp. 1617–1638.

[310] Y. ZHANG, P. LIANG, AND M. CHARIKAR, *A hitting time analysis of stochastic gradient langevin dynamics*, in Conference on Learning Theory, 2017.

[311] Z. ZHANG, *A flexible new technique for camera calibration*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22 (2000), pp. 1330–1334.

[312] H. ZHAO, D. YUAN, H. ZHU, AND J. YIN, *3-d point cloud normal estimation based on fitting algebraic spheres*, in Image Processing (ICIP), 2016 IEEE International Conference on, IEEE, 2016, pp. 2589–2592.

[313] Y. ZHONG, *Intrinsic shape signatures: A shape descriptor for 3d object recognition*, in Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 689–696.

[314] Q.-Y. ZHOU, J. PARK, AND V. KOLTUN, *Fast global registration*, in European Conference on Computer Vision, Springer International Publishing, 2016.

[315] J. ZHU, D. MENG, Z. LI, S. DU, AND Z. YUAN, *Robust registration of partially overlapping point sets via genetic algorithm with growth operator*, Image Processing, IET, 8 (2014), pp. 582–590.

[316] J. ZHU, L. ZHU, Z. JIANG, X. BAI, Z. LI, AND L. WANG, *Local to global registration of multi-view range scans using spanning tree*, Computers & Electrical Engineering, (2016).

[317] J. ZHU, L. ZHU, Z. LI, C. LI, AND J. CUI, *Automatic multi-view registration of unordered range scans without feature extraction*, Neurocomputing, 171 (2016), pp. 1444–1453.

[318] M. Z. ZIA, M. STARK, B. SCHIELE, AND K. SCHINDLER, *Detailed 3d representations for object recognition and modeling*, IEEE transactions on pattern analysis and machine intelligence, 35 (2013), pp. 2608–2623.

[319] T. ZINSSER, J. SCHMIDT, AND H. NIEMANN, *A refined icp algorithm for robust 3-d correspondence estimation*, in Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on, vol. 2, IEEE, 2003, pp. II–695.

[320] O. ÖZYEŞIL, A. SINGER, AND R. BASRI, *Stable camera motion estimation using convex programming*, SIAM Journal on Imaging Sciences, 8 (2015).

This thesis proposes a new pipeline and a set of tools for reconstructing 3D scenes and objects from point clouds in scenarios where prior CAD models are available. Our pipeline involves multiple building blocks and we explore each block in detail proposing novel solutions in order to enable a more efficient, robust and seamless reconstruction experience. The geometric methods developed are applicable to many computer vision problems such as SLAM, SfM, and to applications such as industrial quality inspection and augmented reality.