
OPTIMAL 6D OBJECT POSE ESTIMATION WITH COMMODITY DEPTH SENSORS

Author:

Michael LANDAU

Advisor:

Dr. Peter BELING

A Dissertation

Presented to

THE FACULTY OF THE SCHOOL OF ENGINEERING AND APPLIED SCIENCE

UNIVERSITY OF VIRGINIA

In Partial Fulfillment

of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in the

Department of Systems and Information Engineering

August, 2016



Approval Sheet

The dissertation is submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Systems and Information Engineering

Michael J. Landau

Michael J. Landau

This dissertation has been read and approved by the Examining Committee:

Peter A. Beling

Peter A. Beling, Ph.D., Dissertation Advisor

Stephen D. Patek

Stephen D. Patek, Ph.D., Committee Chairman

Laura E. Barnes

Laura E. Barnes, Ph.D.

Scott T. Acton

Scott T. Acton, Ph.D.

Michael D. DeVore

Michael D. DeVore, Ph.D.

Accepted for the School of Engineering and Applied Science:

Craig H. Benson

Craig H. Benson, Dean

School of Engineering and Applied Science

August, 2016

Abstract

Department of Systems and Information Engineering
UNIVERSITY OF VIRGINIA

Doctor of Philosophy

Optimal 6D Object Pose Estimation with Commodity Depth Sensors

by Michael LANDAU

Accurate 6D object pose estimation, as well as other model-based shape matching objectives such as object detection, classification, and shape inspection, is prominent and necessary in a large number of domains and applications. This includes household and robotics applications to automate various tasks, where input depth measurements are aligned to a corresponding model of the object, such as a CAD that is composed of several interconnected parts, depending on the model's level of detail and complexity. Industrial metrology systems also compare the shapes of engineered components and equipment to their corresponding 3D models for quality assessment and anomaly detection; this often requires very costly and high resolution projection and detection mechanisms. Structured-light laser scanners have, however, emerged as a viable option to provide cheap and reasonably accurate depth estimates, which have recently become commodity-priced within this decade due to the maturity of the technology. These sensors are also robust for semi-transparent and reflective surfaces, dynamic scenery, ambient background light, and temperature drift. Certain structured-light light coding methods unfortunately need to implement a complicated, non-invertible transformation of a distorted light pattern to generate the pixel-by-pixel depth estimates, thereby leading to a loss of information. This can result in poorly estimated or even missing depth data, especially when the object has small 3D features or non-ideal surface properties. Moreover, the informative rigid body constraint is blurred when the nonlinear 2D depth image to 3D pseudo-measurement point cloud transformation is applied, which is the domain where most model-based shape matching methods operate.

Alternatively, estimating pose from information early in the structured-light sensing/processing chain has the potential to alleviate errors induced in subsequent nonlinear processing steps that are in fact locally scene dependent. This motivates one of the main contributions from this dissertation, which presents an asymptotically optimal maximum likelihood estimation method that operates directly on the raw output IR images. This is made possible by an extensive study on a commonly utilized commodity-priced structured-light sensor, which contributed to the proposed high-fidelity IR and depth image predictor and simulator that models the physics of the transmit and receive optics,

the unique IR pattern, and the statistical speckle and detector noise distributions. A new method is also formulated to compute the Fisher information contained in the IR images of the unique structured-light measurement data in order to establish the Cramér-Rao bound, i.e. the lower bound on error for any unbiased pose estimator. The proposed shape-based matching method is shown to outperform cutting edge point set registration methods by an order of magnitude in the respective mean square errors when applied to object pose estimation, and also to approximately attain the Cramér-Rao bound, thereby demonstrating near optimality. This method is additionally shown to produce nearly identical cost evaluation times as a function of model complexity, and to consistently converge in the neighborhood of the true pose parameter, regardless of the number of pixels on target or initialized global optimization error bound.

Acknowledgements

First I would like to thank my advisor, Dr. Peter Beling, for his helpful guidance during my time in the Department of Systems and Information Engineering. He has also provided me with numerous opportunities to gain practical experience and present to large groups within manufacturing companies; this has helped me hone my ability to engage a wide range of backgrounds and interests. I would like to thank Dr. Michael DeVore for taking extra time to carefully review and improve my publications and dissertation write-up, as well as instilling a newfound appreciation for achieving optimal estimation. I would also like to thank my other committee members, Dr. Stephen Patek, Dr. Laura Barnes, and Dr. Scott Acton. A large portion of this dissertation was inspired by their comments, suggestion, and direction before, during, and after my PhD proposal a year ago. I have also received help and support from the other members in our Adaptive Decision Systems (ADS) lab, including Benjamin Choo, Graham Crannell, and Stephen Adams.

I would like to thank my family for their love, support, and encouragement throughout my life. My step-brothers, Lance and Michael, for always being there for me. My step-mother, Diane, for always loving me and treating me as one of her own. My mom, Carolyn; she has provided me with boundless love and encouragement from the very beginning, and allowed me to be as curious and inquisitive as I wanted to be. She nurtures my ambition, and her tremendous enthusiasm for my academic, as well as my personal achievements, is something I am always grateful for. My dad, Herb; he has taught me so much from when I was very young, up to, and continuing through today. He allows me to realize my own potential, inspires my academic and engineering mindset, and his advice, support, and guidance are invaluable to me.

And finally, and most importantly, I would like to thank my fiancée, Alex. Her unwavering love, quiet patience, and encouragement have carried me through the completion of my PhD. I am also deeply grateful to her for taking care of me – not to mention planning our fast-approaching wedding – especially during these last few months and weeks of hard work and critical preparation prior to my defense. I am certain it would've taken much longer to earn my degree without Alex by my side. Our seemingly endless to-do list, long nights, and labor of love are finally coming to fruition, and I cannot wait to begin our new life together in Maryland.

CONTENTS

| | |
|---|-----------|
| Approval Sheet | i |
| Abstract | ii |
| Acknowledgements | iv |
| Contents | v |
| List of Figures | vii |
| List of Tables | ix |
| Abbreviations | x |
| Symbols | xii |
| 1 Introduction | 1 |
| 1.1 Motivation | 2 |
| 1.2 Problem Formulation | 5 |
| 1.3 Dissertation Contributions | 8 |
| 1.3.1 High Fidelity Structured-Light IR and Depth Image Simulator . . | 10 |
| 1.3.2 Information Theoretic Framework for Optimal Pose Estimation . . | 12 |
| 1.3.3 Point Set Registration Maximum Likelihood Estimation (PSR-MLE) | 12 |
| 1.3.4 Structured-Light IR Maximum Likelihood Estimation (SLIR-MLE) | 13 |
| 1.4 Dissertation Organization | 14 |
| 2 Background on Depth Sensors | 15 |
| 2.1 Depth Sensor Classes | 15 |
| 2.2 Structured-Light Depth Sensors | 18 |
| 2.3 Microsoft Kinect Sensor Mechanics | 22 |
| 3 Structured-Light Sensor Models | 28 |
| 3.1 Background | 28 |
| 3.2 Depth Image Error Models | 30 |
| 3.2.1 Measurement Model | 31 |
| 3.2.2 Formulating the Axial Error Model | 32 |
| 3.2.3 Formulating the Lateral Error Model | 32 |

| | | |
|----------|--|------------|
| 3.3 | IR Image Intensity and Noise Models | 34 |
| 3.3.1 | Measurement Model | 35 |
| 3.3.2 | Formulating the IR Intensity Model | 37 |
| 3.3.3 | Formulating the Speckle Noise Model | 38 |
| 3.3.4 | Formulating the Thermal Noise Model | 39 |
| 3.4 | Physical Sensor Models | 40 |
| 3.4.1 | Simulating IR Images | 40 |
| 3.4.2 | Simulating Depth Images | 45 |
| 3.5 | Model Validation | 48 |
| 4 | Optimal Pose Estimator Criteria | 55 |
| 4.1 | Cramér-Rao Bound of Structured-Light Sensors | 55 |
| 4.1.1 | Fisher Information Matrix for IR Image | 56 |
| 4.1.2 | Sensitivity Images | 62 |
| 4.1.3 | Cramér-Rao Bound Examples | 64 |
| 4.2 | Criteria for Informative Measurements | 70 |
| 4.3 | Criteria for an Optimal Estimator | 72 |
| 5 | Point Set Registration | 74 |
| 5.1 | Background | 74 |
| 5.2 | Data Model | 76 |
| 5.3 | Pose Hypothesis Evaluation | 76 |
| 5.3.1 | Point Prior | 77 |
| 5.3.2 | Point-Pair Assignment | 78 |
| 5.3.3 | Decision Rule and Clutter Model | 80 |
| 5.4 | Results | 81 |
| 6 | IR Image Prediction | 88 |
| 6.1 | Background | 88 |
| 6.2 | SLIR-MLE Algorithm | 89 |
| 6.3 | Results | 91 |
| 7 | Conclusion and Future Work | 99 |
| 7.1 | Summary of Dissertation | 99 |
| 7.2 | Calibrating the Sensor Model | 101 |
| 7.3 | Constructing a Real-Time System | 103 |
| | Bibliography | 105 |

LIST OF FIGURES

| | | |
|------|---|----|
| 1.1 | Examples of Model-Based Shape Matching Applications | 3 |
| 1.2 | Depth and IR Images from a Top-down View of Metallic Saucepans . . . | 4 |
| 1.3 | Teapot CAD Utilized in Dissertation for Testing Purposes | 6 |
| 1.4 | Examples of Varying Pixels on Target for IR Images | 7 |
| 1.5 | Examples of Varying Pixels on Target for Depth Images | 8 |
| 1.6 | Examples of Varying Pixels on Target for Point Clouds | 9 |
| 2.1 | Block Diagram for Depth Processing of Structured-Light Sensor | 21 |
| 2.2 | Transmitter and Receiver Position and Orientation | 23 |
| 2.3 | Real and Simulated Kinect Dot Pattern Examples | 24 |
| 2.4 | Block Diagram for Depth Processing of Kinect Sensor | 26 |
| 2.5 | All Quantized Kinect Disparity Values | 27 |
| 3.1 | IR Dot Occlusion and IR/Depth Shadow | 30 |
| 3.2 | Axial and Lateral Depth Image Error Models | 33 |
| 3.3 | Experimental Depth Image of Block Grid | 35 |
| 3.4 | IR Dot Intensity Samples and Fitted Model | 37 |
| 3.5 | IR Speckle and Thermal Noise Histograms and Fitted Models | 39 |
| 3.6 | Physical Cross-sectional Area of IR Dots are Simulated | 42 |
| 3.7 | Sensor Coordinate System to Pixel Coordinate System Transformation . . | 43 |
| 3.8 | IR Dot Splitting and Spreading Examples | 44 |
| 3.9 | Simulated Infrared Image of Block Grid CAD | 45 |
| 3.10 | Simulated Depth Image of Block Grid CAD | 48 |
| 3.11 | Flat Wall Standard Error vs. Distance from Focal Plane Center | 49 |
| 3.12 | Results for Flat and Tilted Wall Standard Errors | 50 |
| 3.13 | Experimental and Simulated Flat Wall Standard Errors and Box Plots . . | 51 |
| 3.14 | Ground Truth of Block Edge Fitted to IR Image | 52 |
| 3.15 | Normalized and Averaged Correlation Functions | 53 |
| 3.16 | Object Edge Bias and Standard Errors for Horizontal and Vertical Edges | 54 |
| 4.1 | Average Correlation Coefficients Demonstrate IR Pixel Independence . . . | 58 |
| 4.2 | Fisher Information Approximation to the FMG Distribution | 60 |
| 4.3 | Speckle, Thermal, and Total Noise Probability Density Functions | 61 |
| 4.4 | Sensitivity Image Examples for Each Pose Parameter | 63 |
| 4.5 | Teapot, Bunny, and Dragon CAD Models Used For Testing | 64 |
| 4.6 | RCRBs for Varying Number of Pixels on Target, SNRs, and IR Dot Patterns | 66 |
| 4.7 | Two IR Dot Patterns to Increase Fisher Information for IR Images | 67 |

| | | |
|-----|--|-----|
| 4.8 | Cramér-Rao Bound Examples Comparing Three Different Objects | 68 |
| 4.9 | RCRBs for Pose Regions of Three Different Objects | 69 |
| 5.1 | Examples of Model Point Clouds with Different Point Priors | 77 |
| 5.2 | PSR Accuracy from Point Clouds with Varying Measurement Points . . . | 83 |
| 5.3 | PSR Accuracy from Point Clouds with Varying Model Points | 84 |
| 5.4 | PSR-MLE Point Priors Increase Pose Estimation Accuracy | 86 |
| 5.5 | PSR Accuracy from Pseudo-Measurement Point Clouds | 87 |
| 6.1 | Operating on Raw IR Images Avoids Inhomogeneous Error Sources | 89 |
| 6.2 | SLIR-MLE is Consistent and Asymptotically Normal | 92 |
| 6.3 | SLIR-MLE Nearly Attains CRB and Outperforms PSR Methods for Teapot | 93 |
| 6.4 | SLIR-MLE is Nearly Optimal for Bunny and Dragon Data Sets | 94 |
| 6.5 | Cost Function Evaluation Time Complexity vs. CAD Model Complexity . | 95 |
| 6.6 | Negative Log-Likelihood Surfaces for Rotation and Translation Parameters | 96 |
| 6.7 | Negative Log-Likelihood Surfaces for Varying Pixels on Target | 97 |
| 6.8 | SLIR-MLE Converges in the Neighborhood of the True Pose Parameter . | 98 |
| 7.1 | Applying SLIR-MLE Concepts to Shape Inspection | 101 |

LIST OF TABLES

| | | |
|-----|---|----|
| 2.1 | Structured-Light Depth Sensors | 20 |
| 2.2 | Correlation Window Uniqueness Test | 25 |
| 3.1 | Quadratic Coefficients for Depth Image Error Models | 34 |
| 3.2 | Default Parameters of Structured-Light Sensor Simulator | 41 |

ABBREVIATIONS

| | |
|--------------|---|
| ADC | A nalog-to- D igital C onverter |
| A-ICP | A nisotropic- I terative C losest P oint |
| BRDF | B idirectional R eflectance D istribution F unction |
| CAD | C omputer- A ided D esign |
| CMOS | C omplementary M etal- O xide- S emiconductor |
| CPD | C oherent P oint D rift |
| CRB | C ramér- R ao B ound |
| DOE | D iffractive O ptical E lement |
| EO/IR | E lectro- O ptical I nfrared |
| FIM | F isher I nformation M atrix |
| FOV | F ield O f V iew |
| ΓMG | Γ amma M odified G aussian |
| ICP | I terative C losest P oint |
| IR | I nfrared R adiation |
| LED | L ight- E mitting D iode |
| MLE | M aximum L ikelihood E stimation |
| MSE | M ean S quare E rror |
| NEF | N atural E xponential F amily |
| PDF | P robability D ensity F unction |
| PSR | P oint S et R egistration |
| RRE | R elative R otation E rror |

| | |
|--------------|---|
| SAD | Sum of A bsolute D ifferences |
| SDK | Software D evelopment K it |
| SNR | Signal-to-Noise R atio |
| ToF | T ime- o f- F light |
| TTE | Total T ranslation E rror |
| UMMSE | Uniformly M inimum M ean S quare E rror |
| UMVUE | Uniformly M inimum V ariance U nbiased E stimator |

SYMBOLS

| | | |
|-------------------------------------|-------------------------------------|------------------|
| b | baseline distance | mm |
| d | horizontal IR dot disparity | pix |
| f_x & f_y | horizontal & vertical focal length | pix |
| \tilde{N} | 3D pseudo-measurement error | mm |
| \tilde{n} | detector noise | 10-bit intensity |
| $\tilde{\gamma}$ | IR speckle noise | unitless |
| $\tilde{\Gamma}$ | scaled IR speckle noise | 10-bit intensity |
| I | mean (expected) IR intensity value | 10-bit intensity |
| Z | IR intensity realization | 10-bit intensity |
| \mathbf{Z} | IR image realization | 10-bit intensity |
| D | mean (expected) depth value | mm |
| \mathcal{D} | noisy depth image | mm |
| $\mathbf{S} = [S_x, S_y, S_z]^\top$ | 3D scene surface point | mm |
| \mathcal{S} | 3D scene point cloud | mm |
| $\mathbf{M} = [M_x, M_y, M_z]^\top$ | 3D model surface point | mm |
| \mathcal{M} | 3D model point cloud | mm |
| $SO(3)$ | entire group of Euclidean rotations | deg ³ |
| V_{FOV} | volume of the FOV of the sensor | mm ³ |

| | | |
|--|-------------------------------------|-----------------------------------|
| \mathcal{I} | Fisher information | $\text{deg}^{-2}, \text{mm}^{-2}$ |
| $\mu(\boldsymbol{\theta})$ | mean IR intensity of object surface | 10-bit intensity |
| $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_6]^\top$ | true 6D pose of object | deg, mm |
| $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ | set of all possible 6D poses | deg, mm |
| $\hat{\boldsymbol{\theta}}$ | 6D pose estimate of object | deg, mm |

1

INTRODUCTION

Epiphany, heresy, novelty, ingenuity, industry, ubiquity. Science. Usually in that order.

– Anonymous

Ever since the introduction of commodity-priced depth sensors to the consumer market at the turn of this decade, researchers have found a wealth of uses that span a wide variety of fields that include manufacturing, robotics, autonomous navigation, graphics, control theory, computational biology, and home living and entertainment. In particular, *structured-light* depth sensors have been sold by the tens of millions as gaming interfaces, research tools, and components in control systems. The ubiquity of this class of sensors is due in large part to their low cost of production, and this, in turn, is possible because of a simplicity in design where the entire field of view can be sensed at a fraction of a second, and processed with minimal computational and optical expense. While the structured pattern allows for a large volume of fast depth reconstruction, this simplified design does come with limitations in terms of reduced accuracy and resolution in the depth estimates, as well as inhomogeneous measurement errors. This poses a challenge for applications that require a high degree of accurate depth measurements, especially if they involve objects with small 3D features and non-ideal surface properties. Therefore, in this dissertation, I develop a framework to get the most out of these commodity sensors by “working closer to the metal”, which includes an optimal model-based shape matching method that operates directly on the raw output infrared (IR) images.

1.1 Motivation

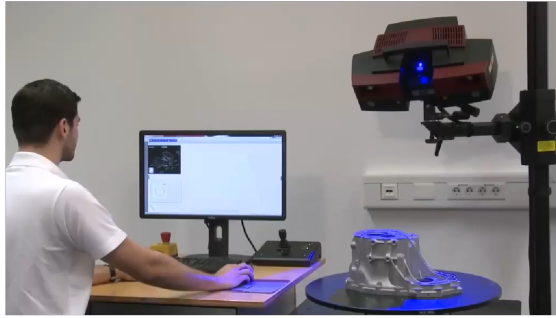
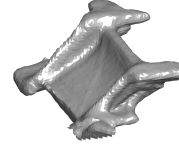
Widespread depth image and video research can mostly be attributed to Microsoft's and PrimeSense's collaborative effort to produce a low cost motion capturing device, in which the player could interact with the console without the need for a game controller. When the Kinect v1 was released in 2010, it was originally intended to compete with other motion controllers such as the Wii U Remote, the PlayStation 3 Move and Eye, and the ASUS Eee Stick. Since then, however, it has been sold by the tens of millions to third parties as a research tool, and has been used as components in various control systems for manufacturing and other industrial domains. Kinect's versatility is compounded by its compact size, making it well suited for a rapidly deployable, small footprint data collection suite. In addition to the device's extensive utility, several efforts have been conducted to reverse-engineer the sensor mechanics and image processing technology. For these reasons, the Kinect's sensor specifications are used in this dissertation for testing and algorithm evaluation as part of the broader study of optimal shape-matching within a set of commodity depth sensors. It should be noted, though, that other structured-light sensors project comparable IR patterns, and process them in much the same way as is done with the Light Coding technology. The reader may therefore use this dissertation as a guide to achieve optimal performance from many other current or future implementations, provided that the raw IR image outputs are available.

Applications for commodity depth sensors can generally be organized into a few categories, which include surface reconstruction, scene mapping and classification, human or hand gesture recognition and activity analysis, and inferencing attributes of inanimate objects (Han *et al.* [46] provide a good review). The latter category can be further broken down into object detection, classification, shape inspection, and pose estimation and tracking, which are typically aided by a known model to describe its 3D surface shape and properties, such as a computer-aided design (CAD). For reference purposes, inferencing attributes of objects with the use of a CAD is indicated by *model-based shape matching* in this dissertation. Several model-based shape matching methods exist that don't require a CAD of the objects, namely shape-based template matching with learned features from depth images [17, 18, 57, 67, 83, 119]. These methods however demand significant computational time and expense to pre-process features for each new object, and require an extensive stored library of the object in the entire discrete pose domain for any degree of accuracy – this methodology is therefore not examined in this dissertation. It should be noted that nearly all applications which involve any category of model-based shape matching require accurate 6D pose estimation to align the CAD with the measurement data set.

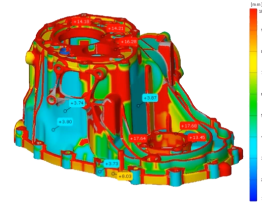
An important domain that requires accurate 6D pose estimation in which known CAD models are available is in the industrial and manufacturing engineering settings. In this domain, a common focus involves analyzing, modeling, and improving work processes.



(A)



(B)



(C)



FIGURE 1.1: Various model-based shape matching applications are displayed that include (A) automated 6D pose estimation of jet engine vanes during a blending process, (B) highly accurate structured-light sensing for industrial metrology [1], and (C) an automated kitchen via a robotic chef [81].

This is regularly motivated by the need to enforce quality, control, and productivity of a worker performing a certain task. In particular, many manufacturing processes possess the need to evaluate the performance variability and physical actions of multiple human workers, such as the blending process at Chromalloy's Orangeburg, NY plant, where jet engine vanes are repaired. Here, the brazed vanes become damaged through extended use, and are returned to the company to be coated and blended back to shape within a confined, hooded area. As seen in Figure 1.1(A), these vanes have a constant surface color and texture, a small and complex shape with intricate 3D features, and the surfaces become specular after blending. Within a group of workers performing the same blending steps, there exists a degree of variability regarding the speed and quality of work. In

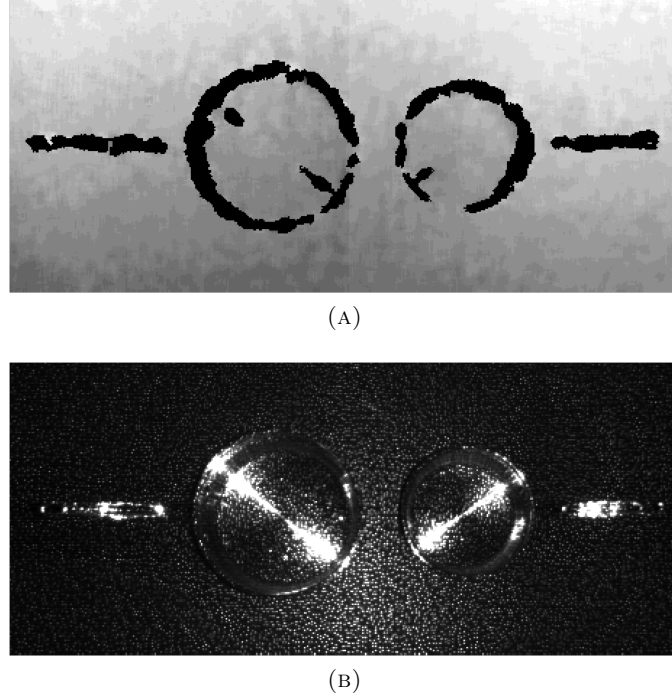


FIGURE 1.2: A top-down view of two metallic saucepans from a range of 1000 mm demonstrate (A) a loss of depth information (black pixels) for the handles and sides, and (B) a more informative IR image of dots reflected from the entire specular surfaces.

order to categorize a worker's performance, it must first be understood what is done physically to the part throughout the work process. It would therefore be beneficial to automate the data collection process, which could also potentially be provided as a tool for training and real-time quality assessment. Accordingly, this was maintained as one of the primary objectives for a research effort conducted in partnership with the Commonwealth Center for Advanced Manufacturing (CCAM), and originally motivated the research for this dissertation.

Metrology is also commonly conducted in industrial and manufacturing settings, in which the shape of components and equipment is inspected for damage or quality assessment [34, 48, 84, 116]. Depending on customer requirements, as well as national and international industry standards, a high-quality depth sensor is typically needed to measure the shape of an engineered object in order to compare with the corresponding CAD to test the accuracy, precision, reliability, and traceability of production. Moreover, the speed with which measurements can be presented, the ability to operate in manufacturing environments with variable temperatures and vibration, and total financial cost are also important factors to consider. Structured-light technology is widely used for industrial metrology, for example the ATOS Triple Scan [Figure 1.1(B)], a structured blue light 3D scanner, is able to provide a point spacing between 0.01 – 0.61 mm [1]. This \$300,000 sensor, however, requires users to operate in the transformed point cloud domain, which necessitates a very high resolution projection and detection system.

Other examples of model-based shape matching can be found in many other domains, all of which require accurate 6D pose estimation. For instance, in the computer vision domain, 3D object detection and classification has been applied to furniture and room recognition [108], household object class recognition [114], and transparent object recognition [68]. Additionally, object pose estimation has been applied to many robotics applications, including inline inspection of composite materials [113], automated unloading of containers [74], and Moley Robotics' robot chef [Figure 1.1(C)], which is set to launch in 2017 to automate gourmet cooking by mimicking the movements of master chefs. Note, Oleynik states in his patent [81] that the Kinect is a "suitable example of RGB-D sensor" for the robot chef, which could either be placed in a stationary position on a railing or on the robotic hands. Example images of two different sized saucepans recorded by a Kinect for Windows at a range of 1000 mm are shown in Figure 1.2, which demonstrate a loss of information (black pixels) for the handles and sides in the depth image, and a fully informative IR image of dots reflected from the entire specular surfaces.

1.2 Problem Formulation

Many aspects of real-world 6D object pose estimation implementations make the goal of achieving accurate results difficult. For instance, depth data sets derived from single sensor measurements provide sparse and incomplete representations of the object's full 3D shape. More specifically, self-occlusions occur, which depend on the object's position and orientation with respect to the sensor. Also, certain object properties and attributes tend to cause issues for current pose estimation methods, especially when the methods are applied to noisy and low resolution depth images. These properties include the object's size, color, texture, surface albedo, and the complexity and symmetry of the shape. Note, the CAD model of the standard Utah teapot [Figure 1.3(B)] is primarily utilized in this dissertation to simulate noisy sensor measurements in order to test the effectiveness of various pose estimation methods. This object was chosen for its complex yet symmetric shape, and because there is no ambiguity in pose over the entire set of Euclidean rotations. The Stanford bunny and dragon are also utilized later in this report in order to demonstrate pose estimator robustness, as they both possess interesting and more complex CAD models.

6D object pose estimation with the aid of CADs can be accomplished in a variety of ways, including but not limited to feature extraction, generalized surface approximations, and by minimizing a cost metric of the measured points. Feature extraction typically involves predefined local features such as edges, corners, or any other distinguishable sections on the surface of the object, which can be matched to corresponding CAD features created offline. This method can be applied to depth images alone [8, 9, 41], or in conjunction with reciprocal RGB intensity images [26, 49, 51]. Most feature identification methods are however sensitive to noise and sensor resolution, and tend to be computationally



FIGURE 1.3: An example image of the standard Utah teapot shown in (A) with dimensions $113.8 \text{ mm} \times 70.2 \text{ mm} \times 56.8 \text{ mm}$ was 3D printed by utilizing the CAD model file in (B), which is used in this dissertation for testing purposes.

expensive for complex structures. Also, the aid of RGB images is more applicable for objects with varying color and texture, though some information can be extracted from monochromatic surfaces through shading cues. Generalized geometric shapes can be fit to the surface of rigid objects, simplifying the iterative cost function reduction or analytical closed form solution to an optimal pose [11, 64]. However, methods for approximating CAD surfaces have trouble in the presence of occlusions or excessive noise, and sacrifice accuracy for decreased computational expense, making generalized surface approximations more suitable for object recognition [21, 111, 125]. Also, similar to 3D feature extraction, decomposing object shapes into convex sub-bodies is not trivial for complex CAD models. Thus, a commonly utilized approach to estimating the pose of complex and monochromatic objects is to maximize a likelihood function of the measured data in the point cloud domain.

Perhaps the most popular process for model-based shape matching is point set registration (PSR), which aligns a discretized version of the CAD model to an unordered collection of measurement points in the 3D sensor-coordinate domain [13, 19, 44, 47, 52, 66, 69, 70, 78]. This class of methods can be advantageous because their usage is sensor independent, they can be robust for statistical mismodels, and are usually intuitive to understand and straightforward to implement. PSR methods, however, suffer from a number of assumptions and measurement limitations. Firstly, they typically assume that the 3D measurement errors of point pairs are independent and corrupted by homogeneous, zero-mean Gaussian error. During alignment, the independence assumption allows for conceptual point swaps, i.e. PSR methods don't impose rigid body constraints. Point cloud methods, in general, also rely on the nonlinear 2D depth image to 3D point cloud transformation to generate the *pseudo-measurement* points. Moreover, information losses are compounded by the structured-light sensor mechanics, where a complicated, non-invertible transformation of a distorted light pattern is performed to generate the pixel-by-pixel depth estimates. These limitations often result in sub-optimal PSR estimator performance when applied to single sensor measurement sets, especially since a significant portion of the full object shape is self-occluded.

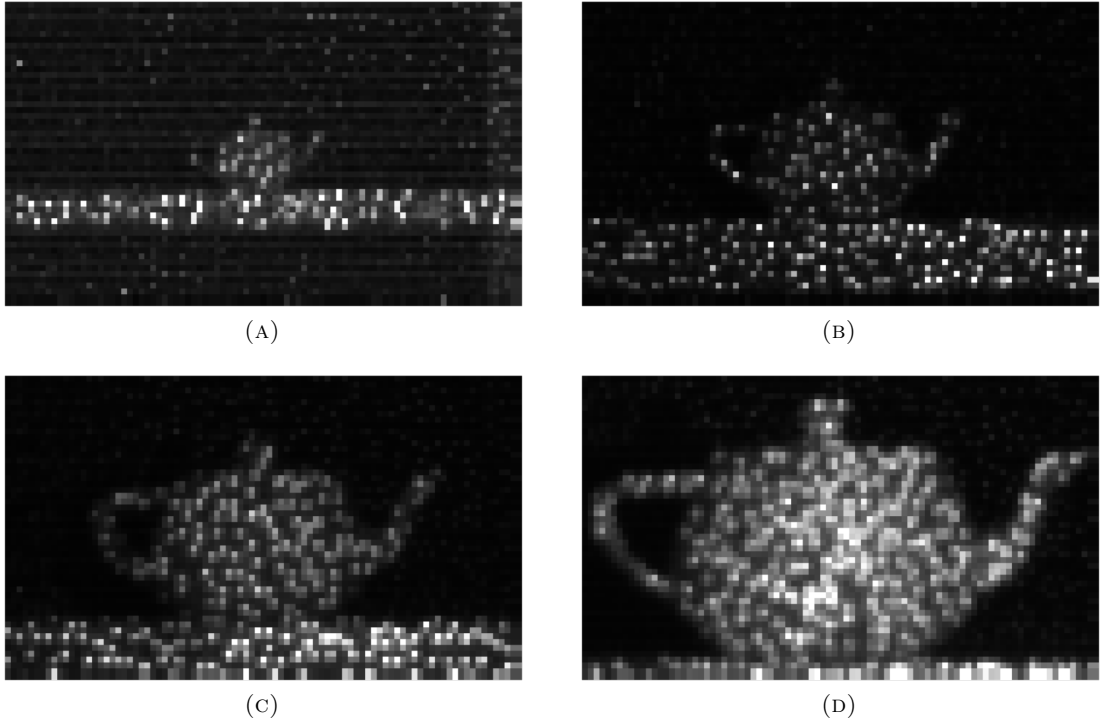


FIGURE 1.4: Examples of real teapot IR images with roughly (A) 157, (B) 563, (C) 1003, and (D) 2036 pixels on target at distances of 2650 mm, 1384 mm, 1048 mm, and 735 mm, respectively, demonstrate retention of information concerning the shape of the object. Note, the maximum display range for (A) was set to a lower IR intensity in order to highlight the full shape.

As is discussed in Chapter 2 in greater detail, structured-light sensors reconstruct the 3D surface via an intricate transformation from IR images (Figure 1.4) to depth images (Figure 1.5). In Chapter 3, it is shown that each processing step involves a degree of error that is highly dependent on the local surface of the scene. For example, correlation window mismatching may occur if part of the dot pattern is occluded, or if IR shadows exist within the image. Mismatch errors also occur when there is depth variation within the window, especially at or near the edges of objects. Also, since depth estimates are restricted to a set number of bit depth values, a degree of systematic quantization error exists, especially at further ranges. It is important to note that these steps result in a non-invertible transformation and a loss of information, i.e. it is impossible to fully recover the original IR image from its respective depth image.

In addition to inhomogeneous and correlated depth errors, sensors that perform spatial-multiplexing coding tend to smooth out features on objects with small cross-sectional areas within the IR image projection. This is depicted in Figure 1.5, when the size of a teapot with a fixed depth is scaled to generate roughly 157, 563, 1003, and 2036 pixels on target. Here, the handle and spout can only become visible in the depth image when the cross-sectional widths are greater than approximately 9 pixels. Varying the number of pixels on target will also result in pseudo-measurement point clouds that exhibit

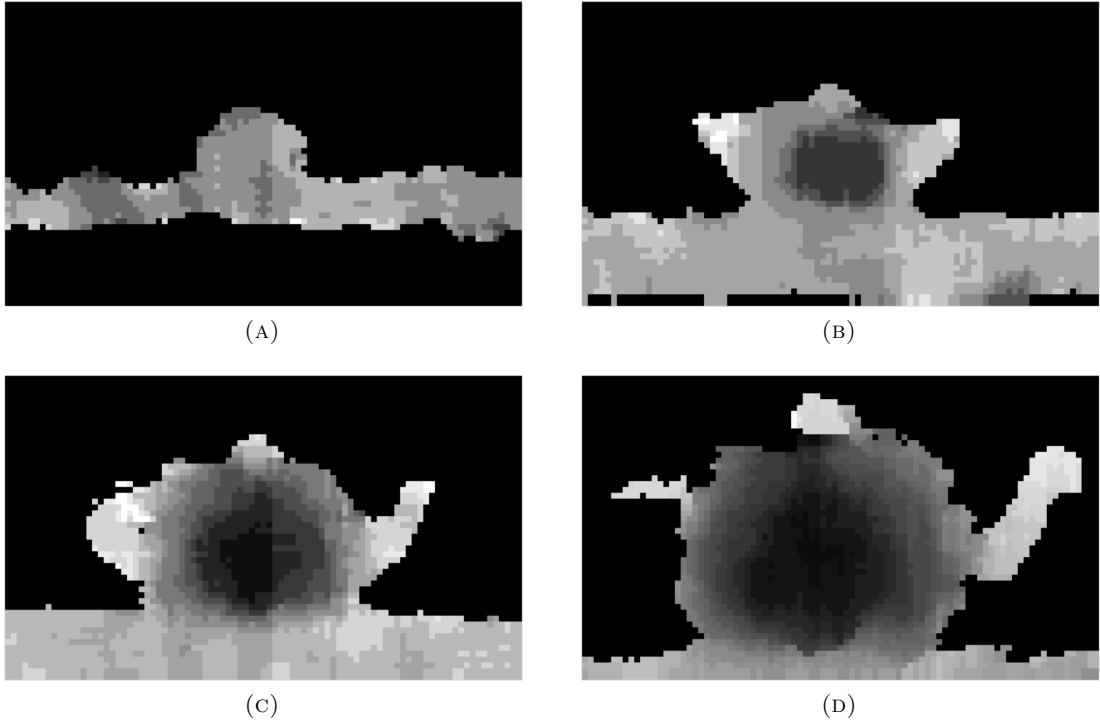


FIGURE 1.5: Examples of real teapot depth images with roughly (A) 157, (B) 563, (C) 1003, and (D) 2036 pixels on target at distances of 2650 mm, 1384 mm, 1048 mm, and 735 mm, respectively, demonstrate a loss of information after depth processing. Most notably, the handle and spout only become visible when the teapot is positioned closer to the sensor.

ambiguity when assigning a correspondence with the discretized CAD model points (Figure 1.6). This presents a challenge to PSR methods at each step of the iterative cost function reduction process. It's worth noting that 'complex' objects with high-accuracy CAD models (i.e. when they are discretized into denser point clouds) will also lead to a mismatch in correspondence.

All of these issues can be managed by a method that predicts the likelihood of the unprocessed IR image. This, in turn, motivates the 6D object pose estimation method proposed in this dissertation, which attempts to predict the raw structured-light IR images for each single sensor measurement. This is made possible with a well-calibrated sensor model that provides the intensity distribution of the dot pattern, as well as the speckle and thermal noise that affect the IR detectors.

1.3 Dissertation Contributions

In order to support the requirements of the model-based shape matching applications presented in Section 1.1, the major emphasis of this dissertation is to develop an analytically optimal pose estimator of arbitrarily shaped objects with corresponding

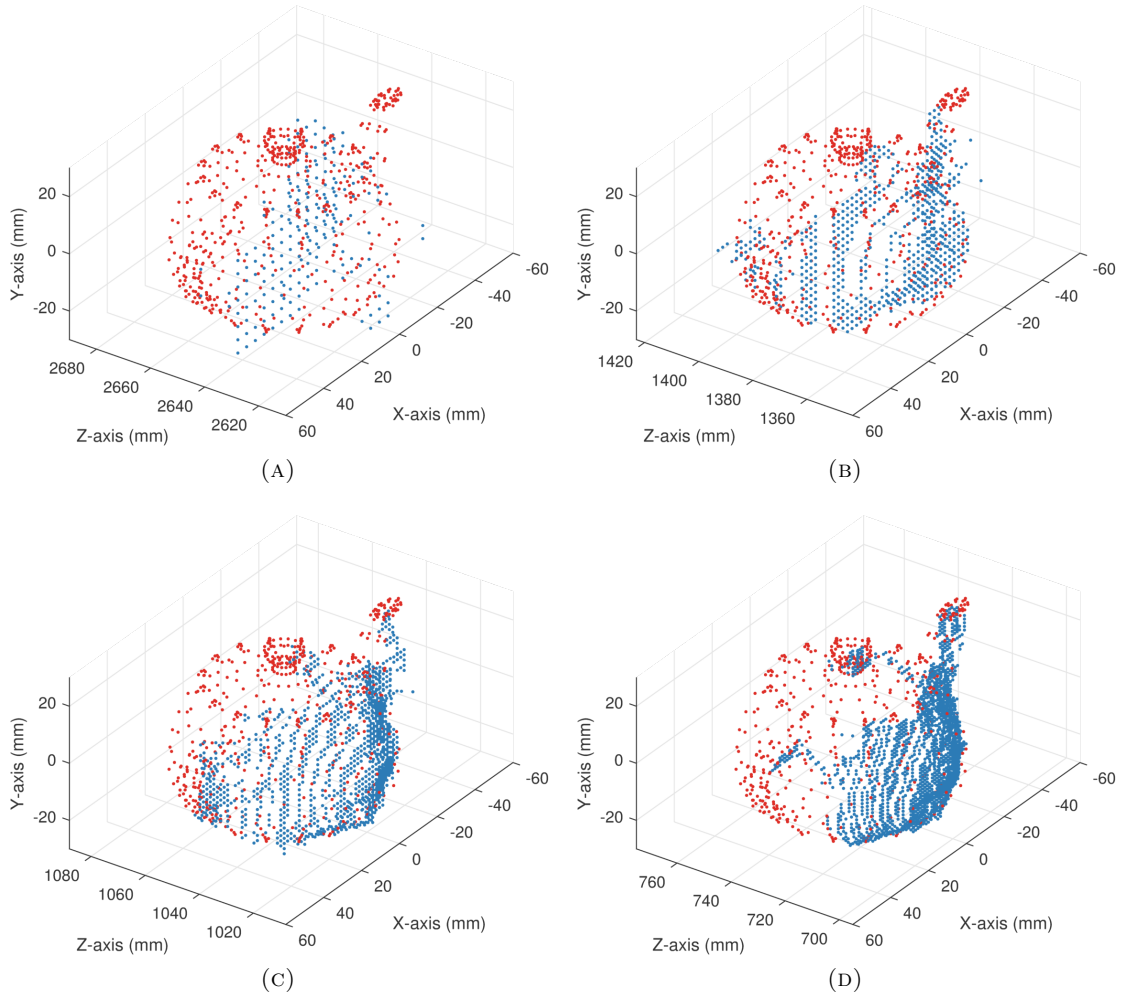


FIGURE 1.6: Examples of real teapot pseudo-measurement point clouds (blue points) with roughly (A) 157, (B) 563, (C) 1003, and (D) 2036 pixels on target at distances of 2650 mm, 1384 mm, 1048 mm, and 735 mm, respectively, demonstrate ambiguity when assigning a correspondence to the discretized CAD model points (red points).

CADs, given the noisy and low resolution data generated by commodity, structured-light sensors. As discussed in Section 1.2, since a loss of information and a reduction of potential accuracy occur by operating on the highly processed depth images and subsequently transformed pseudo-measurement point clouds, an asymptotically optimal maximum likelihood estimation method that operates directly on the raw structured-light IR images is formulated. I also address the question: What constitutes the highest achievable accuracy given the unique sensing products that structured-light sensors provide? Since a closed-formed solution to provide such a metric doesn't exist, a novel method is formulated to approximate the lower bound on 6D pose estimation error for structured-light data. This bound serves as a benchmark for optimal performance for any pose estimator, and provides the supporting evidence that the method introduced in this dissertation approximately achieves the best possible performance. It should be noted that the methods developed as part of this dissertation could also be applied to a

variety of other depth sensors that generate an initial, unprocessed image of a projected light pattern. Moreover, use of the presented method is not limited to just object pose estimation, but could also be applied to other shape matching objectives such as object detection, recognition, and change in shape/anomaly detection.

1.3.1 High Fidelity Structured-Light IR and Depth Image Simulator

In order to formulate an optimal 6D pose estimation model, a complete understanding of the mechanisms, performance characteristics, and the resulting sources of error produced by a given depth sensor need to be established. An initial, collaborative effort was conducted to construct a new set of empirical models for axial and lateral Kinect depth error across the entire focal plane. Through this study, it was noted that the observed error in depth images is due not only to unpredictable IR noise, but also to systematic quantization and mismatched correlation windows from measured sections of a distorted dot pattern. This work resulted in an open access *Sensors* journal paper, titled ‘Statistical Analysis-Based Error Models for the Microsoft Kinect Depth Sensor’ [23].

From the empirical model work, it was observed that the Kinect sensor portrays depth error characteristics that are inhomogeneous and difficult to predict. Accordingly, numerous attempts to construct geometric, empirical, and statistical error models are largely too simplistic, ad-hoc, and imprecise. Another reason for these inadequacies is due to the fact that these models ignore the intermediate noisy IR images, which are processed by way of Kinect’s embedded Light Coding technology. Since the depth processing technology implements a correlation-based method to match local sections of the sensed dot pattern to templates from a reference image, a match is not well-defined when part of the IR pattern is occluded or when segments of the pattern are horizontally offset. This leads to erratic and inhomogeneous depth estimation errors that cannot be faithfully modeled by constructed point cloud covariances that bypass IR imaging and subsequent disparity/depth processing. More specifically, depth estimation error of a measured Kinect image is largely dependent on the orientation of the object and surrounding scene, where error tends to grow larger for pixels that sense near an object edge or tilted surface.

A concurrent observation was that in order to test the effectiveness of a new system that processes output depth data streams, an appropriately large ensemble of experiments with accompanying ground truth is required. This however is not always feasible, or at least proves to be an arduous task that may require the assistance of imprecise or expensive tools to gather the imperfect ‘truthing’ data. There are extensive data sets of objects recorded by the Kinect sensor that were made available online to avoid the tedium of estimating a ground truth, such as the depth data sets provided by Berkley, NYU, Princeton, and the University of Washington¹. These data sets only cover a few specific

¹See: Berkley (kinectdata.com); NYU (cs.nyu.edu/~silberman); Princeton (vision.princeton.edu); UW (rgb-d-dataset.cs.washington.edu)

objects and scenes however, and as such, users are limited to testing their model on depth images not necessarily related to their specific goals. Furthermore, these online resources, as well as many other available data sets, do not include a model of the recorded object, and the user is left to find a CAD model of a related object close in shape and size, or to use a 3D object scanning method [79] to generate a less than perfect model [e.g. the CAD in Figure 1.1(A)].

These limitations motivate the need for a practical, reliable, and extensible Kinect IR and depth image simulator with included built-in speckle and noise models. Several general depth sensor simulators already exist, such as the GA Tech ladar simulation developed by Dixon², which allows the user to modify the sensor specifications and construct a scene with CAD models. However, the Kinect sensor implements specialized depth processing technology in accordance with a unique transmit IR dot pattern, and therefore estimates unique noisy depth images. The Microsoft Robotics Developer Studio (RDS) provides a simulated Kinect sensor that estimates depth and RGB images within the limited depth range of the real Kinect. The RDS software however focuses mainly on simulating the sensor on a mobile platform, and lacks the granularity of Kinect’s inherent depth image noise and error characteristics. More specifically, synthetic sequences are generated by simply rendering ideal images of 3D objects and adding homogeneous Gaussian noise.

Thus, this dissertation’s first main contribution is a high fidelity Kinect simulator that constructs noisy IR and depth images of user-supplied CAD models by modeling the physics of the transmit and receive optics, the unique dot pattern, and by closely following the actual methods and algorithms that the real Kinect sensor employs. The IR and depth image simulators involve newly constructed models, but also take advantage of methods and tools previously developed from other projects, and couple them with well-defined Kinect sensor mechanics and specifications. For instance, I make use of an idealized bitmap representation of the Kinect dot pattern [88], as well as an optimized ray casting method [104] to simulate line-of-sight vectors originating from Kinect’s IR laser transmitter. By modifying these tools, IR dot splitting and spreading can be simulated, as well as random intensity variation and IR noise. These idiosyncrasies distort the measured pattern and contribute to Kinect’s unique error composition, and are therefore a necessary inclusion for a fidelitous simulator. Also, since the IR image is received from a sensor offset by a known baseline distance from the transmitter to perform stereo triangulation, dot occlusions in the IR image and depth image shadowing can be simulated. This study resulted in an IEEE *Transactions on Cybernetics* journal paper, titled ‘Simulating Kinect Infrared and Depth Images’ [60]. The code was also made publicly available on Matlab Central’s File Exchange resource center with an included demo, which takes as input a transformed CAD model that defines the 3D scene, and outputs the corresponding noisy IR and depth images³.

²Dixon, J.H., “LADAR Simulator User Manual,” (2007). (jdixon@ece.gatech.edu)

³See: Matlab Central’s File Exchange resource center for ([Kinect Infrared \(IR\) and Depth Image Simulator](#))

1.3.2 Information Theoretic Framework for Optimal Pose Estimation

In the second contribution of this dissertation, an information theoretic framework is derived to define the criteria for an optimal 6D object pose estimator. The uncovered properties of the Fisher information of structured-light IR images are first exploited to formulate a method that calculates the lower bound on shape matching error, i.e. the Cramér-Rao bound (CRB). This is made possible by the extensive study of the Kinect sensor’s *modus operandi* in [60], where the simulator is utilized to compute the Fisher information contained in the IR image of an object under a given pose. The calculated CRB ultimately serves as a benchmark for optimal performance for any method that operates on either the pseudo-measurement point cloud, noisy depth image, or raw IR image. A number of criteria are then considered that could establish an estimation process as optimal. Common to all criteria, however, requires the design to evaluate accurate statistical models of fully informative measurements, and to estimate parameters with minimum bias and variance. Thus, the need to process the raw IR images of structured-light sensors is further motivated, which is followed by the theory behind formulating a best mean square error estimator.

1.3.3 Point Set Registration Maximum Likelihood Estimation (PSR-MLE)

While object pose estimation continues to be a widely researched topic, the iterative closest point (ICP) algorithm is maintained as perhaps the most popular PSR method because it is intuitive to understand and straightforward to implement, and also doesn’t require local feature extraction. This conceptually simple and fast algorithm iteratively associates measured points to model points (referred to as the source and reference point clouds) by nearest neighbor criteria through a variant of the greedy algorithm. There are many drawbacks, however, including slow convergence, a sensitivity to outliers, a reliance on a good initial guess, and assumed independent and isotropic errors. Moreover, the main issue with using this method arises from the general assumption that each measured point has a corresponding match in the model point cloud. This causes ICP and many variants to perform poorly when the measurement point cloud is much denser or more sparse when compared to the model point cloud.

Instead of requiring an exact pairing between each point set, the third contribution of this dissertation presents a method to combine all of the potential model point candidates via a soft assignment for each measurement point. This is done by adapting the Bayesian object classification model constructed by Zhou and DeVore [28, 123], and merging it with an optimization routine to perform an iterative search on the continuous pose domain until the criterion function converges to the global optimum on the likelihood distribution. Accordingly, the newly proposed method, referred to as point set registration maximum likelihood estimation (PSR-MLE), determines the probability distribution of each measurement generated from the collected surface of transformed model points

based on a Bayesian approach, while also making use of the estimated Kinect sensor error models for depth data from Section 1.3.1. PSR-MLE also begins to address the inhomogeneous error sources by predicting scene and pose dependent facet visibility, though it is still prohibitive to characterize other errors like mismatched correlation windows or depth shadows. This study resulted in an IEEE *Industrial Electronics* conference paper, titled ‘Efficacy of Statistical Model-Based Pose Estimation of Rigid Objects with Corresponding CAD Models using Commodity Depth Sensors’ [59].

1.3.4 Structured-Light IR Maximum Likelihood Estimation (SLIR-MLE)

Ideally, a faithful sensor prediction model can be incorporated into an optimal estimator. As follows, the proposed simulator can be used as part of a new class of estimators for model-based shape matching. One could, for example, generate a noise-free depth image of an object’s CAD at each hypothesized pose, and transform the depth image into a ‘predicted model point cloud’ to align with the noisy pseudo-measurement point cloud. In this case, the predicted model point cloud would manifest shadowing, occlusion, and point spacing due to pixelation from the sensor. Though depth processing of the mean IR image is not precisely the same as the mean depth image, a one-to-one correspondence between the measurement and model point clouds would more closely exist at the correct pose, thereby improving PSR efficacy. This, however, does not alleviate the independent and homogeneity assumptions made with PSR methods. Furthermore, as described in Section 1.2, certain structured-light light coding methods can lead to a loss of information, which results in a reduction of potential accuracy for model-based pose estimation methods that operate on the depth images or subsequently transformed 3D point clouds.

The fourth contribution of this dissertation thus formulates an asymptotically optimal maximum likelihood estimation method that operates directly on the raw structured-light IR images. The proposed structured-light IR image maximum likelihood estimation (SLIR-MLE) method maximizes the likelihood of the measured IR image over the pose region given the object model, sensor model, and calibrated speckle and thermal noise distributions. SLIR-MLE is shown to nearly achieve the calculated CRB for the Kinect sensor by operating on the more informative raw IR images. Furthermore, this method is shown to outperform cutting edge PSR methods by an order of magnitude in the respective mean square errors. Note, the main utility from [60] is taken advantage of by simulating noisy IR and depth images of CAD models given a ground truth pose to evaluate and compare the accuracy of estimator architectures. This study, in conjunction with the work summarized in Section 1.3.2, culminated in a journal paper submitted to IEEE *Transactions on Computational Imaging*, titled ‘Optimal Model-Based 6D Object Pose Estimation with Structured-Light Depth Sensors’ [58].

1.4 Dissertation Organization

The following chapters develop novel frameworks and methodologies, where each following chapter builds upon the research and results of earlier phases, all toward the ultimate goal of optimal 6D pose estimation performance given the noisy and low resolution commodity sensor data. Chapter 2 provides a general overview of the different types of depth sensor technology, as well as background information on the various types of commodity structured-light sensors currently on the market. Also, a thorough overview of the mechanisms, performance characteristics, and the resulting sources of Kinect IR and depth error are presented in this chapter. This detailed review then allowed for the generation of novel depth error, IR intensity, and IR noise models presented in Chapter 3. Physical sensor models are also presented in this chapter, which simulate noisy IR and depth images with errors that are qualitatively and quantitatively close to errors extracted from real measurement data. Chapter 4 motivates the need to utilize the raw structured-light IR images as the measurement source, and establishes the information theoretic framework to compute the Fisher information contained in an IR image and the resulting CRB for any unbiased pose estimator. The PSR-MLE method is then formulated in Chapter 5, which is shown to achieve a monotonic increase in performance with an increase in either points on target or CAD model accuracy. PSR-MLE, as well as other popular and cutting edge PSR methods are, however, unable to achieve the established CRB for various simulated pseudo-measurement point cloud experiments. The SLIR-MLE method is therefore formulated in Chapter 6, which is shown to perform an order of magnitude better than cutting edge PSR methods, and to nearly achieve the calculated CRB with varying amounts of pixels on target. Finally, in Chapter 7, I summarize the dissertation, discuss the steps required to calibrate and adapt the presented framework in order to support imperfect sensor installations, and propose directions to reduce computational time and cost in order to construct a real-time system.

2

BACKGROUND ON DEPTH SENSORS

The first function of a conceptual model is relating the research to the existing body of literature. With the help of a conceptual model a researcher can indicate in what way he is looking at the phenomenon of his research.

– Jan Jonker, Bartjan Pennink (2010)

This chapter is organized as follows: Section 2.1 provides a summary of various range imaging techniques and devices. Section 2.2 explains structured-light pattern projection and spatial-multiplexing coding technology in more detail, and provides a list of current commodity structured-light sensors. Finally, in Section 2.3, a thorough review of the Kinect hardware and software specifications is provided.

2.1 Depth Sensor Classes

The principles behind generating range measurements have been around for centuries – beginning with Greek navigators and astronomers around 200 BCE – but have only become a viable technology for industrial and commercial sensors around the late 1990s. As Blais [16] explains in his review of range sensor development prior to January 2004, the initial stage of research curiosity for 3D-based technology was first developed and demonstrated between the early and mid 1980s. This generated a surge of possible applications in the 1990s, and with the introduction of the microcomputer and low-cost electro-optical IR (EO/IR) devices, hardware capabilities gradually caught up with algorithm development. Thus, in the late 1990s, several companies were able to construct and offer different variants of commercial 3D range sensor systems.

Range imaging and 3D reconstruction can be accomplished in a variety of ways, usually involving triangulation or time delay principles. The techniques include, but are not limited to, passive stereo vision and triangulation/photogrammetry, single-point or slit

laser scanners/sheet of light triangulation, moiré or structured-light pattern projection, time-of-flight (ToF), and interferometry. Pattern projection can be further broken down into three main types of discrete match filters that utilize direct coding, time-multiplexing coding, or spatial-multiplexing coding [91]. Each technique has its advantages and disadvantages, wherein the range camera types are more suitable for different applications depending on the types of object surfaces, environment, and the required precision or accuracy, depth of focus, etc.

Perhaps the earliest technology for practical range reconstruction was stereo vision and triangulation, where two or more rectified images from either a stereo or multiple-camera system are processed by identifying the 2D projections of corresponding points or features with lines that intersect in 3D space. Scharstein and Szeliski [93] provide a comprehensive taxonomy for stereo algorithms that existed around and before the early 2000s. Since any set of passive cameras that sense visible radiation can be implemented in a stereo vision system, it is ideal for objects with varying color or texture and outdoor environments. The main limitations with this technique are its required computational time and cost for high quality results [73], its difficulty in depth discontinuity regions, and its inability to solve the correspondence problem inside regions of homogeneous intensity or color.

Active triangulation became a practical technology for commercial devices when the availability of electrical components that convert optical information to electrical signals emerged. Synchronizing a laser projection with a lateral effect photodiode (LEP) was an early adoption for range reconstruction technology [112]. But when the charge-coupled device (CCD) array was introduced in the early 1980s as a viable optical detector, active triangulation-based sensors became more accurate and stable. Thus, single-point 3D laser scanners were developed that generate high resolution range images with a large depth of focus. This is achieved by sweeping a narrow laser beam across the object surface at different positions of the detector and measuring the displacement of the expected dot position, though at the expense of high cost and computational complexity. The slit laser scanner, on the other hand, is able to trade off a minimal decrease in depth resolution for mechanical simplicity by projecting and sweeping a laser line across the object. Because of this reduction in complexity, slit scanners became widely used in the late 1990s for relatively accurate 3D reconstruction of short-range, indoor surfaces. A disadvantage of triangulation-based 3D scanners, though, are their susceptibility to distortion from motion. This can occur from small amounts of vibration, temperature change, and of course a moving object or sensor; fast image processing is therefore necessary. The introduction of the complementary metal-oxide-semiconductor (CMOS) detector allowed for fast digital conversion and on-chip processing, though slit scanners still need to sacrifice depth of focus or resolution for accuracy.

3D scanning and triangulation technology is also applied to range imaging systems that implement an optical detector and a projected structured-light pattern, without the need for a complex mechanical scanning apparatus. This technique can typically

be made more cost-effective due to its reduction in hardware requirements and image processing complexity. An approach for pattern projection is direct coding, where the gray-level intensity or color is measured and identified in the recorded image for each pixel separately. This technique can be robust for occlusions and projective distortion (e.g. lens distortion), though it tends to perform poorly with non-ideal surfaces (e.g. reflective surfaces), projector and detector noise, and external illumination since they are highly sensitive to gray-level or color distortion. Pattern projection can also include time-multiplexing coding, such as binary patterns or Gray code sequences [40], to generate a unique code-word for each pixel, where multiple frames are required to identify the pattern in the recorded images. This technique is also robust for occlusions and projective distortion, in addition to non-ideal surfaces; they, however, suffer from distortion from motion, and are therefore not suitable for dynamic scenes. Arguably the most popular approach for pattern projection is utilizing spatial-multiplexing coding technology with a static pattern for fast, accurate, and low-cost depth processing. Spatial-multiplexing structured-light depth sensors are also generally robust for projector and detector noise and non-idealities, color and gray-level distortion, non-ideal surfaces, external illumination, and distortion from motion [77]. Hence, this class of depth sensors is the primary focus in this dissertation, and is explained in more detail in the next two sections and Chapter 3.

ToF cameras are relatively new range imaging devices when compared to the commercial implementations of the other techniques. The technology is similar to radar systems, in that a pulse of laser light is reflected off a surface and back to the detector, where depth is estimated by the measured time or phase delay (referred to as the ‘mixing process’). This technique is part of the larger class of scannerless lidar/ladar devices, where the entire scene is captured and measured by a single laser pulse. As with scanning lidar systems, these devices are ideal for larger volumes and longer ranges (e.g. aerial topography) since range accuracy is mostly constant for the entire imaged volume. Moreover, the illumination unit can be placed next to the detector (ideally, they are collocated), which can reduce systematic and inhomogeneous errors from occlusions. Various mixing processes can however result in a superimposed signal by light reflected from surfaces at different tilt angles and depths, which results in incorrect range estimates (in contrast to scanning lidars that illuminate a single point). Though ToF cameras don’t require mechanical moving parts like rotating laser beams or mirrors, and can estimate depth at higher frame rates with lower expense (when compared to scanning lasers), very sensitive, high-frequency bandwidth electronics are required to resolve time delays on the order of picoseconds. ToF sensors are beginning to be a viable, cost-effective approach for estimating the depth of a scene, however, due to recent advances in intensity-modulated continuous wave laser technology. For instance, Microsoft released their second version of the Kinect in September 2014 with the Xbox One, and a Kinect for Windows v2 in July 2015 for around \$200, which contributes to the commoditization of ToF technology. When compared to the Kinect v1, the Kinect v2 is better suited for applications that require high precision and accuracy at longer ranges (e.g. face gesture recognition of a static

person at a depth of 1.2 meters, as suggested by Sarbolandi et al. [92]). Nonetheless, ToF cameras suffer from various systematic and stochastic error sources, including ambient background light, temperature drift, and internal scattering. Time averaging can be implemented in order to reduce some of these errors (e.g. the Kinect v2 collects 10 frames), though this can result in distortion from motion or incorrect object edge measurements. For a detailed comparison and error analysis of eight current ToF sensors, refer to [36].

Though shape measurement and reconstruction is more commonly done with triangulation and time delay methods, depth can also be determined by measuring the phase shift of a reflected coherent light or wave source. For example, holographic interferometry analyzes the optical frequency phase difference of a diffracted light field scattered on the surface of an object, mostly for object deformation, stress, strain, and vibration analysis [16]. Also, the differences in phase of two or more synthetic aperture radar (SAR) images can be used in an interferometric SAR (InSAR) system to measure longer range, static surfaces, primarily for geophysical monitoring and structural engineering. SAR systems, however, require known rotational movements of the sensor or object. Additionally, inverse SAR (ISAR) systems jointly estimate the rotational motion and resulting image of moving objects. These methodologies become difficult when the rotation is erratic during the collection process however, and are therefore not suitable for applications with rapidly changing object poses.

2.2 Structured-Light Depth Sensors

In the early to mid 2000s, 3D reconstruction capabilities were still considered in their early stages of development, and needed improvements with image quality, rendering, and ease of use to compete with equivalent 2D processing counterparts. However, as was discussed in [16], the major drawback for depth sensor production was the limited availability of cheap electronic components and matured electro-optical technology. While some companies offered promise of extremely high range precision and accuracy, many failed due to an inability to market their products. Even by the late 2000s, commercial offerings were still in the \$10,000 range, including the Mesa Imaging SwissRanger 4000 (SR4000) ToF (offered for around \$9,000), and the PMD Technologies CamCube 2.0 ToF (offered for around \$12,000) [29]. There was therefore a demand for a much lower cost depth sensor within the commercial and research communities.

The Microsoft Kinect v1 was then released in November 2010, which was initially intended as a human gesture recognition controller in the gaming context. Resultantly, because it is small sized and commodity-priced, its usefulness as a 3D depth sensor was rapidly recognized within many third party applications, and in February 2012 a version for Windows was released. The Kinect v1 has since become the most popular and widely used consumer-grade depth sensor for research purposes, which is apparent in the vast number of publications regarding Kinect data, as well as in its total sales surpassing

24 million units. There are also several well-established developer communities that share resources involving the use of Kinect, such as the Robot Operating System (ROS) and the Microsoft Developer Network (MSDN). And since the sensor has a simple and inexpensive optical design [120], the cost of Kinect has been driven down so low that nearly any research group across many disciplines can have access to a reliable and fairly accurate depth sensor for a variety of applications.

This widespread growth clearly demonstrates the maturity of the underlying principle for structured-light range sensing. Accordingly, several companies have proliferated structured-light capabilities, and provide an alternate selection of commodity sensors that include: the Asus Xtion, PrimeSense Carmine, Intel’s RealSense F200 and SR300, and the Orbbec Astra and Persee (Table 2.1). With each new iteration, the sensors become more refined and capable of providing higher IR, as well as axial and lateral depth resolution. For instance, PrimeSense reported an axial and lateral resolution of 0.5 – 6 mm and 0.6 – 2.4 mm, respectively, between the Carmine’s operational range of 350 – 1400 mm [6] (compared to Kinect’s 2 – 40 mm and 1.3 – 6.7 mm axial and lateral resolution between 800 – 4000 mm). The RealSense F200, released in January 2015, provides 16-bit depth images [63] (compared to Kinect’s 11-bit depth images), and is intended to be integrated with a desktop or laptop computer for natural gesture-based interaction. Finally, the most recent structured-light sensor introduced to the market in June 2016 is the Persee, with an axial resolution of 5 mm at a depth of 2000 mm, and a much wider operational range of 400 – 8000 mm. Orbbec also boasts an on-board computer and built-in Advanced RISC Machine (ARM) processor, making the Persee the “world’s first 3D camera-computer” [7], which can automate home devices, control TVs through voice commands and gestures, and integrate within industrial settings for mobile machinery, object scanning, etc. It is also rumored that Apple will integrate 3D sensing technology into their iDevices for gesture recognition in TVs, computers, smartphones/tablets, cars, etc., which will most likely involve structured-light technology. This is supported by several recent patents [37, 76, 85] published between 2013 and 2016 by PrimeSense. Note, PrimeSense is the company that conceived the hardware design and camera sensor chip used in the Kinect v1, and was purchased by Apple in November 2013.

Spatial-multiplexing coding structured-light depth sensors are a subset of active triangulation sensors that process unique, static patterns of light that are projected onto the scene. Several discrete spatial-multiplexing techniques exist, such as De Bruijn sequences, non-formal codification, and M-arrays [91], each with their advantages and disadvantages. Note, continuous coding methods also exist such as phase shifting and frequency-multiplexing [91], though this technology hasn’t progressed nearly as much in the last few years due to their high sensitivity to noise. While most spatial-multiplexing methods can process color-coded patterns, invisible structured-light patterns such as IR, imperceptible, and filtered light sources are advantageous because quick and easy depth processing can be performed in weakly lit or homogeneously textured environments [32].



| Sensor | Year | Cost | IR Pattern | Range | IR Res | Depth Res |
|--|------|-------|------------|---------------|--------|-----------|
|  Kinect v1 | 2010 | \$100 | Dots | 400 – 4500 mm | 10-bit | 11-bit |
|  Asus Xtion Pro/LIVE | 2011 | \$170 | Dots | 800 – 3500 mm | 10-bit | 12-bit |
|  PrimeSense Carmine | 2013 | \$130 | Dots | 350 – 1400 mm | 10-bit | 11-bit |
|  Intel RealSense F200 | 2015 | \$100 | Stripes | 200 – 1200 mm | 8-bit | 16-bit |
|  Orbbec Astra | 2015 | \$150 | Dots | 600 – 6000 mm | 16-bit | 16-bit |
|  Intel RealSense SR300 | 2016 | \$130 | Stripes | 200 – 1200 mm | 10-bit | 16-bit |
|  Orbbec Persee | 2016 | \$200 | Dots | 400 – 8000 mm | 16-bit | 16-bit |

TABLE 2.1: A comparison of current structured-light sensors demonstrate competitive commodity pricing as well as the corresponding depth ranges and resolutions.

Among the different patterns for spatial-multiplexing coding, unique stripe and punctual dot patterns remain as the most widely used. Features in stripe pattern grids can be identified in several ways, though identifying individual stripes is mostly limited to either

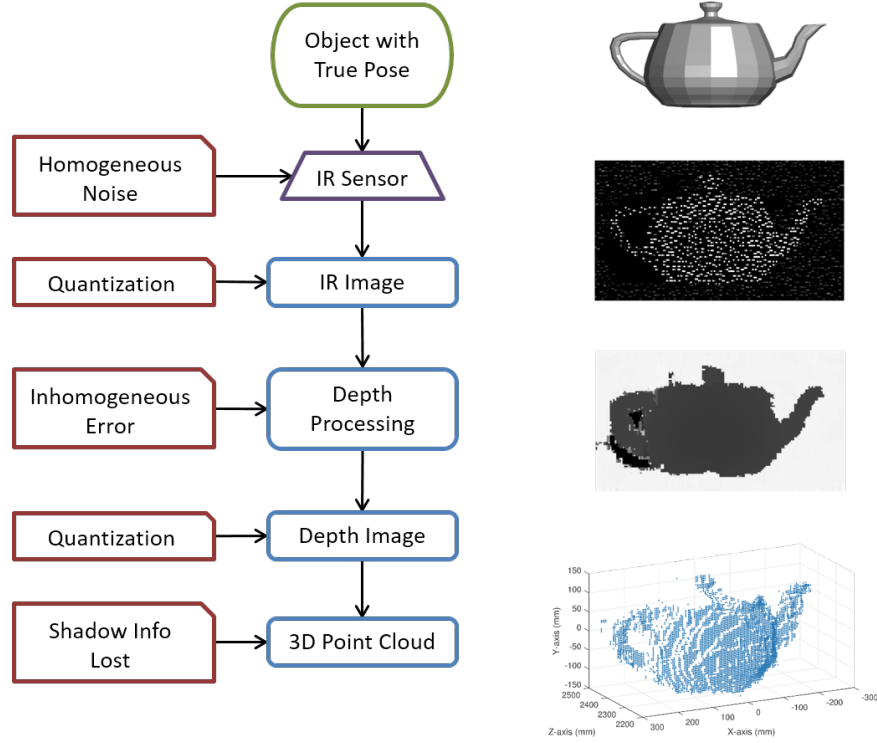


FIGURE 2.1: The block diagram of structured-light sensor mechanics summarizes the general sources of error associated with each step of depth processing and pseudo-measurement point cloud transformation.

tracing or counting them when a monotone light pattern is implemented. Note, indexing can also be accomplished with segmented stripes or repeated gray-scale stripe patterns that are generated from a sequence of three or more intensity levels [40], though these methods are limited to smooth and continuous surfaces. On the other hand, devices that implement IR dot patterns (also referred to as grid indexing) can measure the spatial displacement of a local rectangular grid or *window* of pixels by correlating it to the expected pattern from a reference IR image, which is faster and more robust for IR occlusions and shadows. 2D grid pattern techniques include mini-patterns used as code words and pseudo-random binary arrays for invisible light projections.

While structured-light pattern projection devices have many cost advantages and a robustness for certain scene properties, they do suffer from numerous error sources during depth processing, as summarized in Figure 2.1. For instance, the size of the window in M-arrays and grid indexing determines the robustness for discontinuous surfaces or noise. More precisely, if the window is small, there is less of a chance for a mismatch to the reference image when processing depth in regions of depth discontinuities. In these cases, pattern stretching exists on highly tilted surfaces, and pattern occlusions and shadows exist near the edges of objects; this is explained in more detail later in Section 3.5. Conversely, if the window is large, depth processing is more robust for random noise from speckle and thermal agitation. Note, speckle noise is the result of coherent laser light reflected off of rough surfaces (with respect to the laser wavelength) that scatter the

beam out of phase on point sources with random depths. The optical phases of the point sources result in exponentially distributed powers that get summed into each detector to produce a time-invariant, gamma distributed power [43]. Also, thermal noise is the result of residual heat inside the casing of the device, which agitates the charge carriers of the electronics components (e.g. on the CCD or CMOS), and ends up as white Gaussian distributed power at each detector [12]. Though pattern projecting sensors require less power to illuminate the scene when compared to other devices such as ToF, temperature drift does have an effect on range measurement precision and accuracy [31].

2.3 Microsoft Kinect Sensor Mechanics

This section provides a detailed review of the Microsoft Kinect sensor’s underlying mechanisms and performance characteristics based on existing literature, which in turn motivated the design for the constructed simulation models presented in Chapter 3. Before the initial release in 2010, PrimeSense first published a patent [120] for the design of a system allowing for the capability of real-time object reconstruction by depth mapping with projected dot patterns. Many additional patents [35, 38, 95–97] were then issued and disclosed to the public that detail several structured-light and projection system designs to triangulate and estimate a scene’s depth data. However, much of the sensor implementation details were still left undisclosed. In the past several years, great lengths have been taken to understand the device¹ in order to surmise exactly how depth images are generated using PrimeSense’s patented Light Coding technology.

The Kinect system makes use of two devices to construct depth images: a class 1M IR laser and an Aptina MT9M001 CMOS sensor [2, 3], which are coplanar and aligned to have parallel optical axes, and are offset by a baseline distance of 75 mm [56]. The laser projector and monochrome IR image sensor both work in stereo as a matricial active triangulation system where a pseudo-random IR dot pattern is transmitted and received, respectively. According to [98] and stated as an assumption in [38], the origin of the 3D sensor coordinate system is centered exactly over the position of the receiver’s image capture assembly with coordinates (0,0,0) mm, which places the center of the optical axes of transmit illumination at coordinates (−75,0,0) mm. In this regard, the terms *receiver coordinate system* and *sensor coordinate system* are used interchangeably. The Kinect for Windows v1 SDK [4] reports the angular field of view (FOV) of the sensor as 58.5° and 45.6° along the horizontal and vertical axes, respectively. The SDK also reports the default operational depth range to be between 800 mm and 4000 mm, though the hardware can be switched to Near Mode, which provides a range of 500 mm to 3000 mm. For a visual depiction of the transmitter and receiver position, orientation, and FOV, refer to Figure 2.2.

¹See: blog sites hackaday.com and futurepicture.org for multiple examples of Kinect ‘hacking’

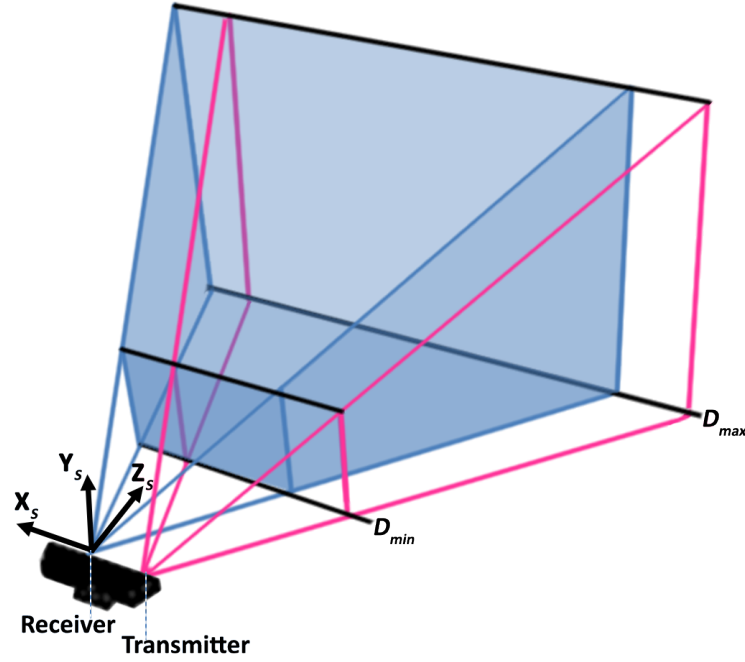


FIGURE 2.2: A pattern projection sensor transmitter and receiver position and orientation are displayed, as well as the minimum D_{min} and maximum D_{max} operational depths. The sensor coordinate system is collocated with the receiver coordinate system, and the X-axis \mathbf{X}_s and Z-axis \mathbf{Z}_s of the sensor coordinate system are respectively collinear and parallel with the corresponding transmitter coordinate system axes.

The laser projection system, referred to as the *transmitter*, contains the illumination unit that produces a coherent light source, and a diffuser unit that generates and propagates a pseudo-random, uncorrelated dot pattern onto a region in space. The illumination unit consists of an IR light-emitting diode (LED) which generates a beam in a relatively narrow band of wavelengths [96]. The diffuser unit contains two diffractive optical elements (DOEs) arranged in series, which are comprised of transparencies that act as active optical surfaces to diffract the input IR beam [35, 97]. A positive image of the dot pattern on the first DOE generates the uncorrelated distribution of bright and dark spots so that the auto-correlation of local patterns on the transverse plane is minimal. The second DOE then tiles the diffracted beam into a 3×3 grid of output beams with a specified fan-out angle to at least partially cover the region of space within the FOV of the imaging unit. According to [35], since the receiver's FOV is within a certain constraint, the second DOE is constructed to provide nearly perfect tiling with minimal overlap and/or gaps of the transmitted dot pattern. The diffraction pattern in the center tile suffers slight barrel distortion however, and the outer tiles suffer significant pincushion distortion [97], which are likely due to the manufactured miniaturized projection system. It should be noted that while the dot pattern is common to all Kinect sensors, small discrepancies in the factory installed diffuser unit result in differences between the projected patterns of each sensor when placed at the same position and orientation [Figure 2.3(A) – 2.3(C)]. The yellow stars in Figure 2.3 represent bright dots on the projected surface, which result

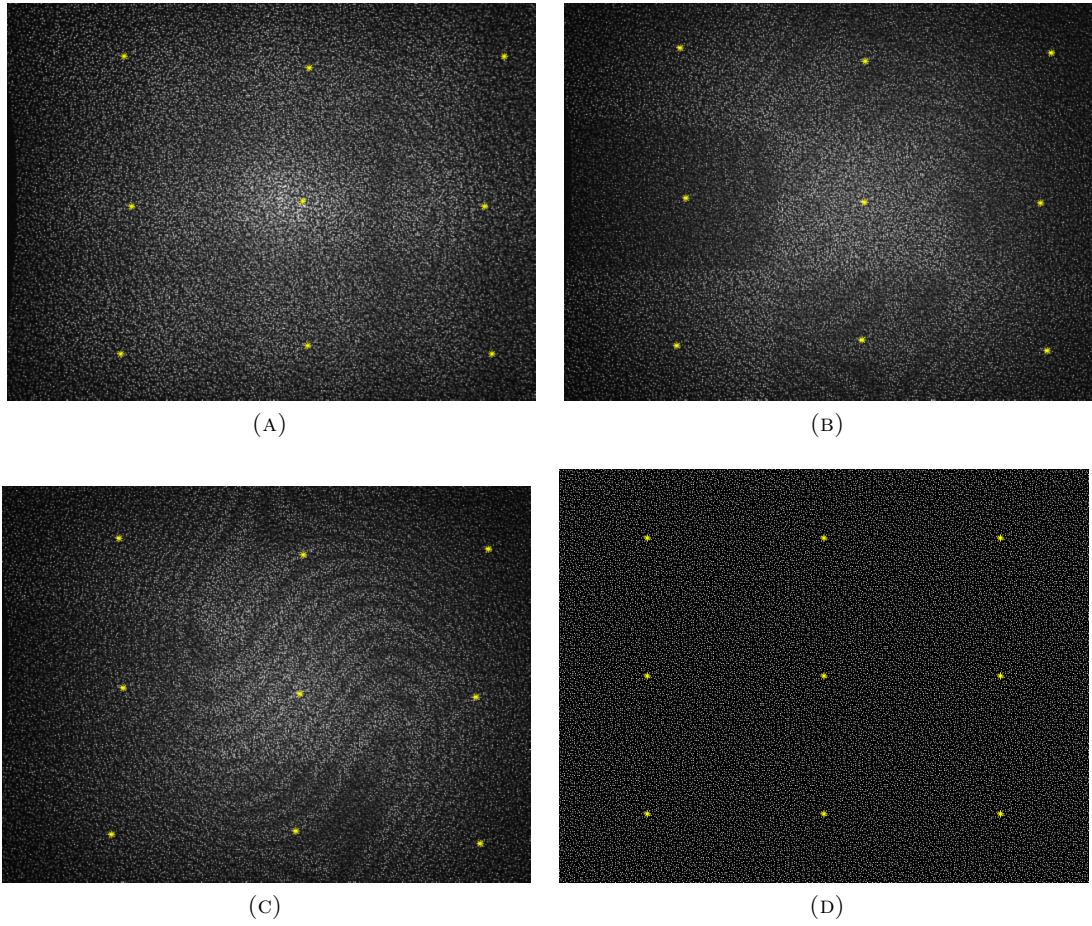


FIGURE 2.3: The dot patterns of a Kinect for Windows in (A) and two Kinects for Xbox in (B) and (C) are projected on a flat wall from a distance of 1000 mm. Note that the projection of each pattern is similar, and related by a 3D rotation depending on the orientation of the Kinect diffuser installation. The installation variability can clearly be observed from differences in the bright dot locations (yellow stars), which differ by an average distance of 10 pixels. Also displayed in (D) is the idealized binary replication of the Kinect dot pattern [88], which was used in this dissertation to simulate IR images.

from an excess of IR light that is passed through and propagated by the system; this is a side effect of the zero-order beam problem [95].

The laser imaging system, referred to as the *receiver*, contains the CMOS active-pixel digital sensor with $1,280 \text{ columns} \times 1,024 \text{ rows}$ of active pixels and a 10-bit analog-to-digital converter (ADC) on-chip, which records images at 30 fps [5]. A bandpass filter positioned near the lens on the receiver assembly allows light only in the IR spectrum to be detected, while filtering out most of the ambient light that would otherwise reduce the contrast of the image [35]. Accordingly, each photoconductor in the sensing array receives different sections of the transmitted IR dot pattern, where photons are interpreted as a current or voltage drop [103], and quantized into integer intensity values between 1 and $2^{10} = 1,024$ for each pixel on the imaging platform. Though the size of the dots increases with an increase in depth, the ratio remains constant, and for ideal surfaces the average

| | 5 Columns | 7 Columns | 9 Columns | 11 Columns |
|----------------|-----------|-----------|-----------|------------|
| 5 Rows | 43130 | 6152 | 838 | 91 |
| 7 Rows | 5900 | 352 | 13 | 0 |
| 9 Rows | 810 | 18 | 0 | 0 |
| 11 Rows | 133 | 0 | 0 | 0 |

TABLE 2.2: A uniqueness test for correlation windows with varying rows and columns demonstrates that Kinect implements a 9×9 pixel window.

dot size is designed to remain at least two pixels wide [120]. It is worth noting here that prior to depth mapping, Kinect uses 2×2 binning to downsample the IR image to 640×512 pixels, where a transmit dot then fills a single pixel on an ideal surface. Also, since the propagated beam has a relatively large depth of focus, high contrast IR images are maintained over the operational range of depths [95]. A predetermined threshold presumably filters and converts the IR intensity images to binary images to allow for faster processing.

Since the pseudo-random dot pattern is not time varying and does not vary along the Z-axis, Reichinger [88] was able to construct a single image of an idealized binary replication by capturing the IR pattern with a camera. The term ‘idealized’ is used to imply a theoretically perfect factory installation of the diffuser unit that generates and propagates the dot pattern, and as such does not portray a 3D rotation when projected on a flat wall. Moreover, it is assumed that the tiled pattern forms a perfect 3×3 grid of an initially generated 211×165 sub-pattern of bright and dark spots, where nonlinear distortions are removed. Also, for simplicity, the shape and size of each dot fills a single cell (pixel) of the constructed grid. The final image [Figure 2.3(D)] has a resolution of 633×495 pixels, where $3,861/34,815 = 11.09\%$ of the pixels contain a dot. This is consistent with the low duty cycle requirement enforced in [35], where the fraction of bright spots should not exceed $\frac{1}{10}^{th}$ the area of the total projected pattern in order to have good performance in the depth mapping process.

All the defining properties of Kinect’s spatially fixed dot pattern allow for reduced optical and computational complexity for object reconstruction [120]. Therefore, with the advent of PrimeSense’s Light Coding object reconstruction technology, all 3D information of a scene is processed in real-time on the company’s PS1080 SoC (System-on-Chip), another device integrated within the Kinect casing [2]. Object reconstruction is performed by first estimating the horizontal shift or *disparity* of a local window of the dot pattern on the received image, and then computing the depth D from disparity d using the standard stereo-triangulation formula [103]

$$D = f_x \frac{b}{d} \quad (2.1)$$

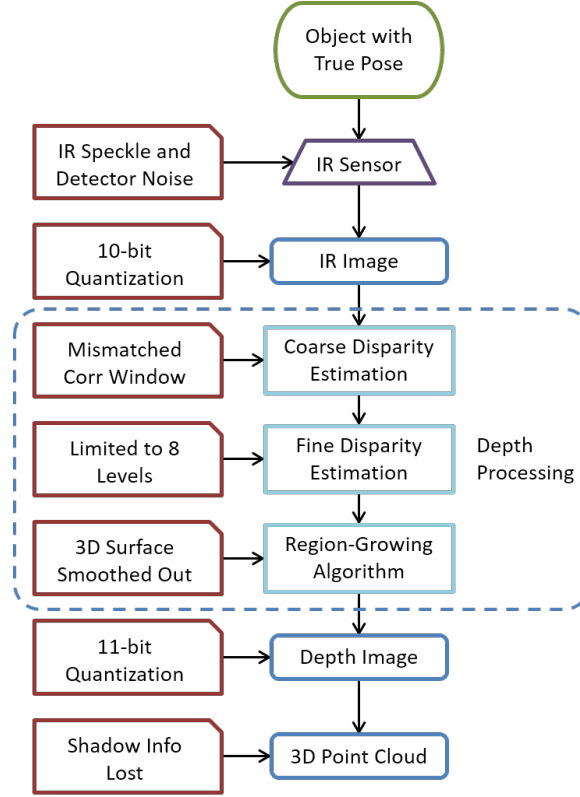


FIGURE 2.4: The block diagram of Kinect sensor mechanics summarizes the steps taken to process depth images of an object, and details the sources of error associated with each step.

for each of the 640×480 downsampled pixels. Here, b represents the baseline distance between the transmitter and receiver, and f_x represents the horizontal focal length of the receiver, measured in pixels. The size of the correlation window has been determined to be 9×9 pixels based on observation and analysis of various IR image samples [56]. This theory can be further supported by constructing correlation windows of various sizes, centering them on each pixel within the sub-pattern of the idealized dot pattern, and comparing them to windows centered on the remaining pixels within the same row (Table 2.2). A 9×9 window size is the smallest in order to achieve no pattern overlap and preserve a unique code.

Disparity estimation is accomplished in two steps, as depicted in the block diagram in Figure 2.4. The first step determines the best match between a spatial-multiplexing window centered on a pixel of the measured IR image and a window of a reference IR image. Since the transmitter and receiver are rectified to have corresponding horizontal epipolar lines, the search can be limited to the same row of pixels. A series of reference images, which represents the pattern projected on planar surfaces parallel to the optical axis at various depths, is pre-captured and stored on the memory utility of the system. In order to find the best match between the measured and reference images, the cross-correlation values of each window pair is maximized [38]. This step is inherently robust for

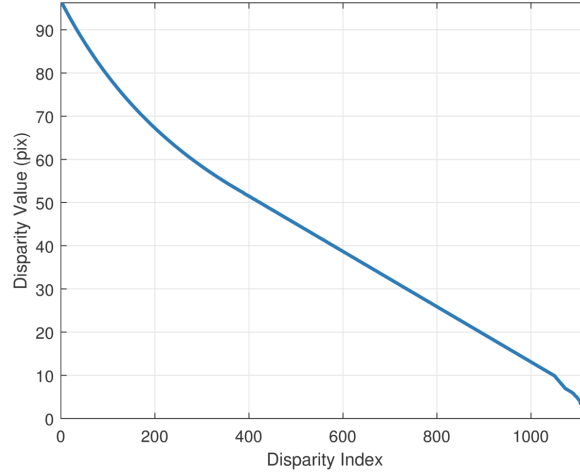


FIGURE 2.5: The disparity values were computed by using (2.1) with all quantized Kinect for Windows depth values.

a degree of lens and perspective distortion, though the disparity estimate is limited to an integer pixel shift corresponding to the correlation peak. Furthermore, cross-correlation works best on flat surfaces that are parallel to the focal plane, but suffers error when the dot pattern is significantly distorted for windows that span significant depth variation. Though it is not explicitly stated, [120] suggests that a prediction-based region growing algorithm that interpolates depth values from neighboring pixels provides a good trade-off between complexity and performance. Thus, this method is also most likely performed to disambiguate potential matches and improve the quality of depth estimates on pixels with correlations below a threshold.

The second step in disparity estimation performs a sub-pixel refinement to determine how much the IR dots of the window are splitting between adjacent receive pixels, which further improves the estimated disparity accuracy. Based on the observation of Kinect output disparity values done by [56] and [72], the Kinect decidedly estimates disparity to a resolution of $\frac{1}{8}^{th}$ of a pixel. Concordantly, by plotting the estimated disparity from the array of all quantized Kinect depth values, the data shows a linear trend with a slope of roughly $-\frac{1}{8}$ pixels (Figure 2.5). This is consistent with the sub-pixel refinement interpolation factor, which means Kinect provides quantized 11-bit depth values contingent on all estimable sub-pixel disparity values.

3

STRUCTURED-LIGHT SENSOR MODELS

There are many specific techniques that modelers use, which enable us to discover aspects of reality that may not be obvious to everyone...

– William Silvert (2001)

This chapter is organized as follows: Section 3.1 presents an overview of several attempts to generate empirical models that estimate axial and lateral error from Kinect depth images. In Section 3.2, a new set of axial and lateral depth error models is presented that accounts for lens distortion across the focal plane. A set of empirical models for the intensity and random speckle of the IR dots, as well as the thermal noise in the detectors is then constructed in Section 3.3. Together, the ensemble of information is used to construct the proposed simulator in Section 3.4, which models disparity estimation to construct depth images from simulated IR images. It should be noted that the simulator focuses on first principles, and therefore does not model post-processing filters to improve depth estimates. Example results of the simulator are then presented in Section 3.5, which are compared to experimental data and existing error models to validate the accuracy.

3.1 Background

Several reports have been published that model the 3D spatial errors exhibited by point clouds extracted from Kinect depth images. In these reports, various geometric, empirical, and statistical models are developed for *axial* and *lateral* error. In this context, the axial component refers to the distance along the depth or Z-axis and the lateral component refers to distances orthogonal to depth along the X and Y-axes of the focal plane. A thorough compilation of independent analyses that characterize error and noise types in Kinect depth images can be found in [71]. Here, Mallick *et al.* provide a uniform

nomenclature for different error types and models, which are used for reference in this chapter.

An early investigation of the Kinect for Xbox composed by Menna *et al.* [72] reports the sensor's lateral and depth error models based on basic geometric constraints of a typical triangulation system. More specifically, the model assumes a pinhole camera system specified only by the intrinsic parameters of pixel size and focal length of the receiver, and the baseline distance. Their proposed standard error models account for predicted quadratic and linear errors that exist in the axial and lateral components over the range space of Kinect. However, such geometric models have since been deemed too simplistic, and more sophisticated models were proposed that account for other key influential factors.

In contrast, Nguyen *et al.* [80] developed empirical models for axial and lateral error, and explored the effects of flat tilted surfaces on depth estimation. In doing so, they added a hyperbolic term to their quadratic axial and linear lateral models by fitting parameters to experimental data. This hyperbolic term accounts for an observed increase in measured variance with an increase in surface tilt angle and decrease in surface depth. Nguyen *et al.* imprecisely relate their models to the sensor's point spread function (PSF), which was characterized as the distribution of absolute errors from the distance of the observed edges to fitted lines. Instead, Mutto *et al.* [77] correctly explain that IR dots projected on excessively slanted surfaces are spread over more than two pre-downsampled pixels, where shifted dots characterized by various disparities within the correlation window can lead to a poor match from the correspondence algorithm. It should be noted that due to the nature of empirical models, the estimates of model coefficients are strongly dependent on the measured data sets, and errors from any hidden variables not accounted for will be mixed into the estimated parameter values. In the case of [80], factors such as surface material properties and ambient thermal conditions are implicitly kept constant since they are not modeled. Also, since depth data was collected on surfaces near the center of the optical axis, error due to lens distortion was not considered.

It has been demonstrated by several reports that post-processing filters can reduce certain causes of error to improve the accuracy of depth image estimates. For instance, Tasic *et al.* [105] and Yang *et al.* [117] provide methods to inpaint non-measured depth values due to certain causes (e.g. specular and absorptive surfaces) by respectively utilizing Kinect IR and RGB images. Yu *et al.* [118] also provide a depth map repair method to fill in missing values due to IR image shadows, which are caused by IR dots obstructed from an object separated at a distance from its background. Shadowing and occlusion are depicted in Figure 3.1, where the region of a 3D scene (upper yellow section) imaged by the receiver (blue line-of-sight vector) cannot be reached by IR dots emanating from the transmitter (magenta line-of-sight vector). There may be a benefit from predicting the size and shape of shadows, though, since they produce estimable systematic errors.

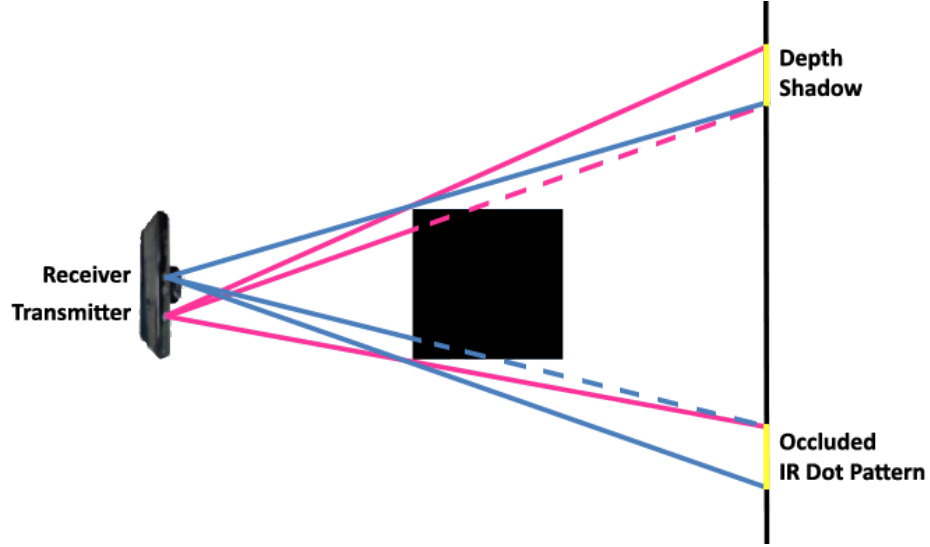


FIGURE 3.1: A schematic is provided to explain the cause of IR dot occlusions and depth shadows. This mechanism also motivates the design for the simulated IR dot visibility detection, discussed later in Section 3.4.1.

Unfortunately, all reported error models are directly formed from depth images, and overlook the intermediate IR images that are used to generate them. As many researchers point out ([77],[71], etc.), various sources of error (e.g. disparity quantization and missing depth due to shadows) are systemic, many of which are predictable given a CAD model of the scene. Additionally, it may be possible to formulate disparity mismatch error statistics as a function of the non-uniform distribution of IR dots and depth variation within correlation windows. There is also little focus placed on analyzing the effect of IR intensity variation due to dot splitting, spreading, and random speckle and IR detector noise. These effects contribute to unpredictable components of random depth error, and are therefore an important factor to consider for adequate error model construction.

3.2 Depth Image Error Models

This section presents a new set of empirical models for axial and lateral depth error to be utilized with point set registration (PSR) methods, which were presented in a journal article authored by Choo *et al.* at the University of Virginia [23]. These models were formulated in recognition of the nonlinear pincushion distortion that contributes to an increase in depth error with an increase in radial distance from the optical center, as discussed in Section 2.3. Thus, the observed error in depth images is due not only to unpredictable IR noise, but also to systematic quantization and mismatched correlation windows from measured sections of a distorted dot pattern. In order to construct the error models, two depth data sets were constructed, each consisting of 100 frames at 17 separate surface depths between 800 mm and 4000 mm, separated by 200 mm. The first data set corresponds to flat walls in order to obtain axial errors and the second corresponds to

a checkerboard block grid in order to obtain axial errors. As is assumed in most other depth error models ([72],[80], etc.), the error from each pseudo-measurement transformed from a depth image is represented as a unique multivariate Gaussian distribution. Thus, standard errors in each dimension (σ_x , σ_y , and σ_z) are obtained by first computing the differences between the measured and expected values, and then by computing the standard deviation for each pixel separately across the entire focal plane from the 100 frame data sets. Quadratic polynomials are subsequently fit to the 3D hypersurfaces of standard errors, where the main effects and interaction terms are described by

$$\sigma(i, j, z) = \beta_1 + \beta_2 i + \beta_3 j + \beta_4 z + \beta_5 i j + \beta_6 i z + \beta_7 j z + \beta_8 i^2 + \beta_9 j^2 + \beta_{10} z^2. \quad (3.1)$$

Note, integer i, j and real valued depth z represent a triplet in the receiver coordinate system, where i, j denote the pixel row and column distance from the center pixel, respectively.

3.2.1 Measurement Model

In the analysis of the depth image data sets, let the set of points comprising the pseudo-measurements transformed from the pixel to sensor coordinate system be denoted by \mathcal{S} . Let $\mathbf{S} \in \mathcal{S}$ be a point from the measured scene in 3D coordinates, where $\mathbf{S} = [S_x, S_y, S_z]^\top$ represents the coordinates of the measurement with respect to the sensor coordinate system. Next, let the discretized model point cloud (in this case, the ground truth surface) be denoted by \mathcal{M} . Let $\mathbf{M} \in \mathcal{M}$ be a point on the surface of the model in 3D coordinates, where $\mathbf{M} = [M_x, M_y, M_z]^\top$ represents the coordinates of a model point center with respect to the sensor coordinate system. If \mathbf{M} is treated as the point of origin from measured point \mathbf{S} , the measurement is modeled as

$$\mathbf{S} = \mathbf{M} + \tilde{N}, \quad (3.2)$$

where the ground truth point is corrupted by a unique measurement error distribution \tilde{N} . Since it is assumed that each pseudo-measurement can be represented as a unique multivariate Gaussian distribution, the probability density function (PDF) is described by

$$f_{\tilde{N}}(s; m) = \frac{1}{(2\pi)^{\frac{3}{2}} \left| \Sigma_{\mathbf{S}} \right|^{\frac{1}{2}}} e^{-\frac{1}{2}(s-m)^\top \Sigma_{\mathbf{S}}^{-1}(s-m)}. \quad (3.3)$$

Note, the standard error in each dimension is assumed as independent; therefore each pseudo-measurement covariance is modeled as a 3×3 diagonal matrix

$$\Sigma_{\mathbf{S}} = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix}, \quad (3.4)$$

with quadratic models as described by (3.1).

3.2.2 Formulating the Axial Error Model

In order to construct the axial depth error model, 17 distinct 100 frame data sets of a matte-painted (i.e. diffuse) planar wall parallel to the optical axis were examined, which were recorded by a Kinect for Windows sensor positioned at various distances. The ground truth model point clouds were computed by performing a least-squares solution to the best-fit planes over the 100 frame depth data sets. Thus, the standard deviation of errors is calculated from the histogram of depth differences, which is referred to as the *standard depth error* metric. The resulting coefficients of the quadratic model for the hypersurface of standard depth errors are shown in Table 3.1, which resulted in a 0.785 R^2 value. Note, these values are a result of a recalibration to a new collection of depth data sets, different from those presented in [23]. In Figure 3.2(C), the hypersurface fits are plotted on the left for the sensor positioned in front of the wall at a depth of 1600 mm, 2400 mm, and 3200 mm. On the right, the standard depth errors are scatter plotted as a function of radial distance from the center of the focal plane, along with the Menna [72], Nguyen [80], and recalibrated Choo models. The Menna and Nguyen models clearly misrepresent the standard depth errors by assuming a constant, linear model for depth estimates across the focal plane.

3.2.3 Formulating the Lateral Error Model

In the analysis of X and Y-axis lateral depth errors, a checkered block grid pattern was recorded at 17 depths for a sequence of 100 frames. Ground truth models were obtained for each depth data set by aligning a grid with known dimensions to the edge data extracted from the transformed pseudo-measurement point cloud surfaces. Thus, the distances between the measured vertical and horizontal block edges and ground truth models construct the histogram of X and Y-axis lateral depth errors, where the standard deviation of the edge location error is referred to as the *standard edge error* metric. The resulting recalibrated coefficients of the quadratic models for the X and Y-axis hypersurfaces of standard edge errors are shown in Table 3.1. In Figures 3.2(A) and 3.2(B), the hypersurface fits are again plotted on the left for the sensor positioned in front of the wall at a depth of 1600 mm, 2400 mm, and 3200 mm. On the right, the

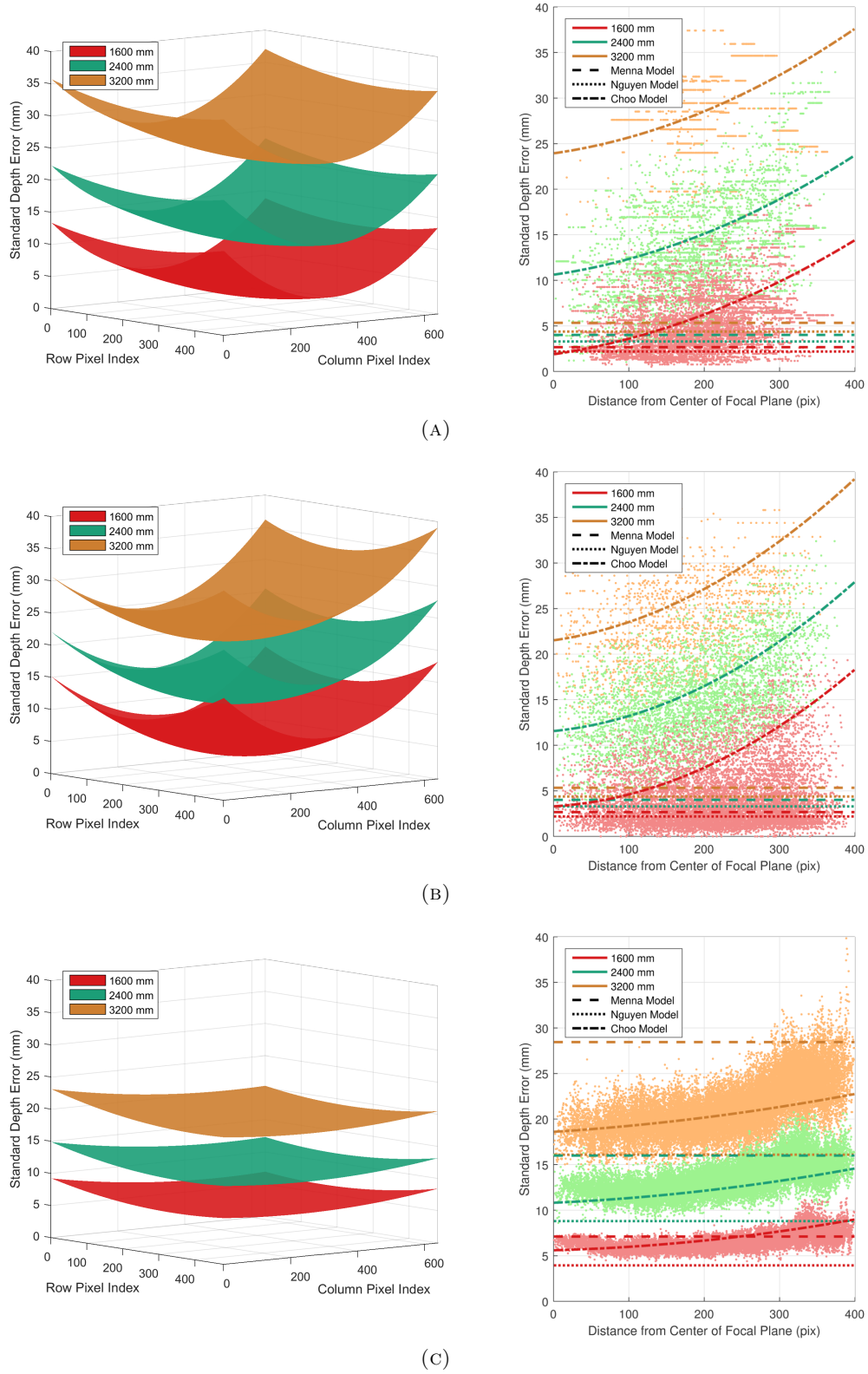


FIGURE 3.2: The 3D standard error hypersurface fits for the recalibrated Choo models [23] are plotted alongside the 2D standard error fits as a function of radial distance for the Menna [72], Nguyen [80], and recalibrated Choo models at different depths in the (A) lateral X-axis, (B) lateral Y-axis, and (C) axial Z-axis dimensions.

| | <i>X</i> | <i>Y</i> | <i>Z</i> |
|--------------|----------|----------|----------|
| β_1 | 9.36 | 6.42 | 5.63 |
| β_2 | -1.11e-2 | -3.52e-2 | -1.18e-2 |
| β_3 | -5.71e-2 | -6.03e-2 | -9.52e-3 |
| β_4 | -3.18e-3 | 3.36e-3 | -9.65e-4 |
| β_5 | -8.24e-7 | 3.04e-6 | 1.16e-5 |
| β_6 | -2.48e-6 | 1.50e-6 | -1.72e-6 |
| β_7 | 8.60e-7 | 4.09e-6 | -5.05e-7 |
| β_8 | 3.03e-5 | 7.17e-5 | 2.13e-5 |
| β_9 | 8.81e-5 | 8.71e-5 | 1.05e-5 |
| β_{10} | 3.58e-6 | 1.33e-6 | 2.01e-6 |

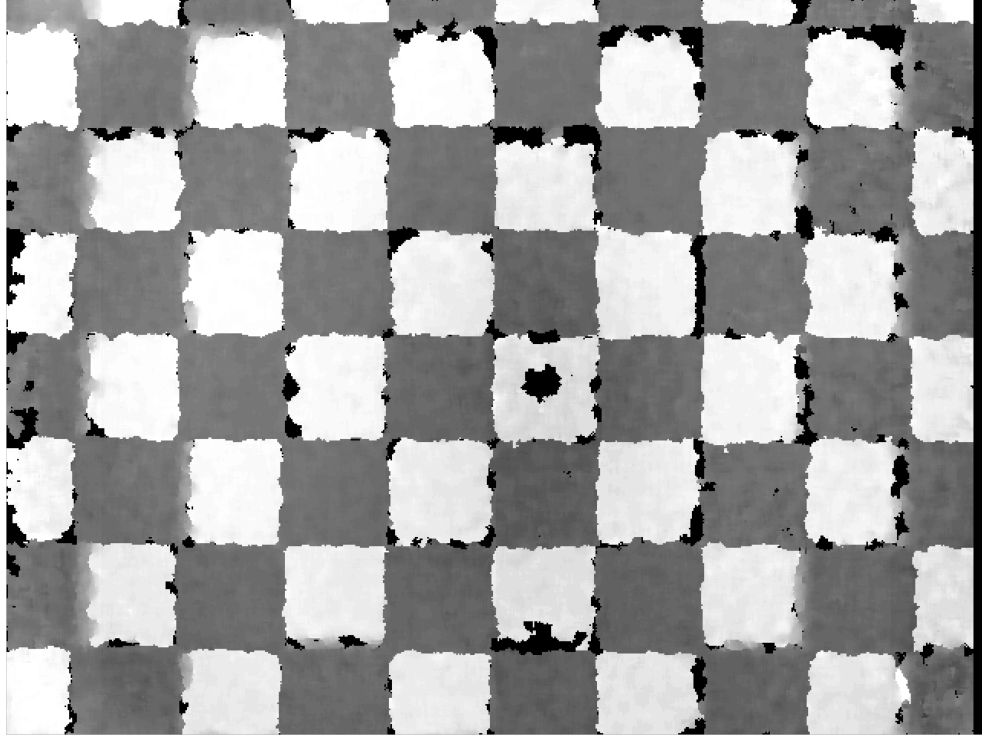
TABLE 3.1: The recalibrated coefficients of the main effects and interaction terms for the quadratic polynomial fits of the 3D hypersurfaces are displayed for the X, Y, and Z-axes.

standard depth errors are scatter plotted as a function of radial distance, where the Menna and Nguyen models significantly underestimate the standard edge errors.

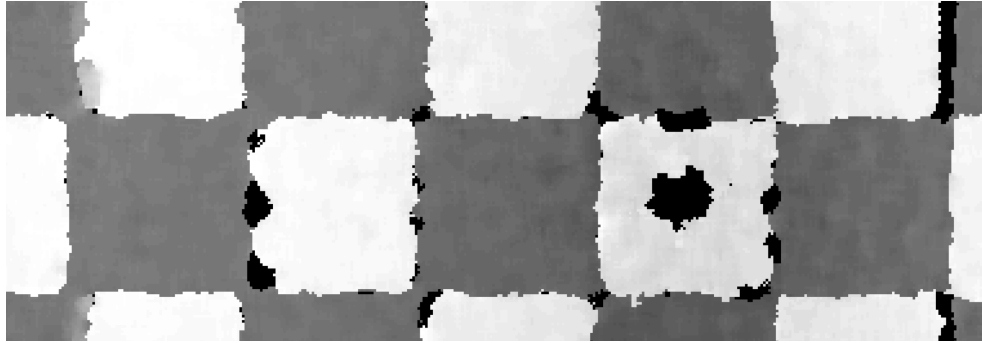
It should be noted that the blocks utilized for lateral error model construction have a specular surface property, and cause the IR light to mostly reflect in a single outgoing direction. This occasionally results in poorly estimated or even missing depth data as seen near the edges of some blocks in Figure 3.3, which could contribute to possible measurement error mismodeling. Also note that these models do not account for changing environmental and surface properties, and assume that varying distances between the object and background surfaces do not have an effect on the standard error of estimated object edges. Lastly, these models along with the Menna and Nguyen models cannot provide bias errors since they do not have available ground truth, and are relegated to using the edge fits to the measured images.

3.3 IR Image Intensity and Noise Models

In this section, a combination of physical and empirical models for the intensity of the received IR dots are presented, which includes PDFs for multiplicative speckle and additive IR detector noise. Since Kinect’s sub-pixel refinement step relies on the intensities of dots splitting neighboring pixels to determine the best match from a similarity algorithm, random IR sources directly affect disparity estimation. These effects are studied later in this chapter, where the depth data presented in Section 3.2.2 are examined (i.e. when



(A)



(B)

FIGURE 3.3: An example of an experimental depth image of the block grid recorded by a Kinect for Windows from a depth of 800 mm is displayed, which was utilized to construct the lateral depth image error models.

dot pattern distortion due to spreading is minimal). In order to construct the intensity models, the corresponding IR image streams of the flat, diffuse surfaces are utilized. Since the ADC resolution is 10-bit on-chip [5], these receiver images consist of 1024 levels of quantized integers representing the detected IR intensities.

3.3.1 Measurement Model

In the analysis of the IR image data sets, the detector outputs Z are modeled as the expected dot intensity I corrupted by random speckle \tilde{n}_I and random detector noise \tilde{n}

$$Z = I + \tilde{n}_I + \tilde{n}, \quad (3.5)$$

where \tilde{n}_I is proportional to I . The random speckle is assumed to vary for the different surface depths, but remain constant during the 30 Hz time samples. Therefore, detectors containing dots are modeled by

$$Z_{i,j,k,t} = \gamma_{i,j,k} \cdot I_{i,j,k} + \tilde{n}_{i,j,k,t}, \quad (3.6)$$

where i, j is the pixel row and column ID, k is the depth, and t is the time sample. Here, $\gamma_{i,j,k}$ is the unitless, random intensity multiplicative term providing the random speckle, and $\tilde{n}_{i,j,k,t}$ represents random detector noise that does vary with time. Note, it is assumed that the speckle and detector noise variables are independent $\tilde{n} \perp \gamma$. Also note, as will be shown later in Chapter 6, it is useful to assume unity for the mean speckle noise and naught for the mean thermal noise in (3.6) for simplicity. Thus, $\mu_\gamma = 1$ and $\sigma_\gamma^2 = \frac{1}{k}$ when speckle noise is fitted to a gamma distribution with shape parameter k , and $\mu_n = 0$ when thermal noise is fitted to a Gaussian distribution with calibrated σ_n , which are valid assumptions, as will be shown in Sections 3.3.3 and 3.3.4, respectively.

It is also useful to consider the γI term in (3.6) as a single, scaled speckle noise variable $\tilde{\Gamma}$ with scale parameter $\frac{I}{k}$, mean $\mu_\Gamma = I$, and variance $\sigma_\Gamma^2 = \frac{I^2}{k}$

$$Z_{i,j,k,t} = \tilde{\Gamma}_{i,j,k} + \tilde{n}_{i,j,k,t}. \quad (3.7)$$

The PDF for a pixel that contains IR dot energy is then the convolution of thermal $f_{\tilde{n}}$ and scaled speckle $f_{\tilde{\Gamma}}$ PDFs,

$$f_{\text{FMG}}(z; k, I, \sigma_n) = f_{\tilde{\Gamma}}(z; I) * f_{\tilde{n}}(z), \quad (3.8)$$

which has a first and second central moment given by

$$\begin{aligned} \mu_{\text{FMG}} &= \mu_\Gamma + \mu_n = I, \\ \sigma_{\text{FMG}}^2 &= \sigma_\Gamma^2 + \sigma_n^2 = \frac{I^2}{k} + \sigma_n^2. \end{aligned} \quad (3.9)$$

Note, the expression in (3.8) gives rise to the gamma modified Gaussian (FMG), an extension of the better known exponentially modified Gaussian (EMG) distribution. In the case of IR intensity distributions, the FMG is a convolution of an informative gamma, and a non-informative Gaussian, i.e. it doesn't depend on I .

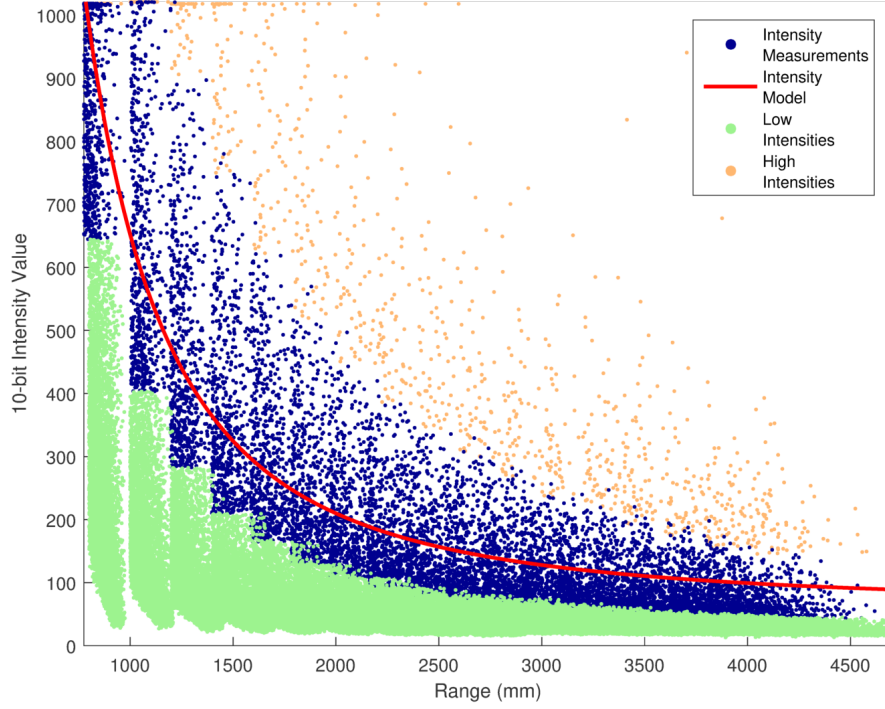


FIGURE 3.4: The filtered intensity samples generated from unsaturated IR dots (blue dots) were used to fit the intensity model (red line), which follows an inverse square model for the distance between the sensor and the surface point.

3.3.2 Formulating the IR Intensity Model

The IR intensity samples $Z_{i,j,k,t}$ from the aggregate collection of all 17 data sets are scatter plotted in Figure 3.4 as a function of range, i.e. the distance between the sensor and surface point. For each given depth experiment, range varies by pixel angular values, therefore the bands in the figure represent intensities from different depth sets. Following Reichinger’s binary dot pattern model (discussed in Section 2.3), the lower 88.9% of IR image intensities (green samples) are filtered out, separately for each depth. This is under the assumption that filtered-out pixels contained either ambient intensity only or fractional energy from dots that straddle adjacent pixels, so that there are enough remaining intensities (blue samples) from dot energy that nearly intersect single pixels. The top 1% of the highest intensities (yellow samples) were also filtered out, again separately for each depth, to avoid overfitting saturated values resulting from very bright IR dots.

As Zelevsky *et al.* [120] explain, the brightness of a detected dot depends on the range r between the illuminator and surface point, where the IR intensity I falloff rate follows the inverse square law (i.e. $I \propto 1/r^2$) when the beam is scattered in all directions on a diffuse surface with an angular extent larger than the beamwidth [33]. Many additional factors influence the detected intensity; however Choe *et al.* [22] argue that the Kinect intensity model can be simplified to the albedo of the surface, global brightness, surface normal,

and transmitter lighting direction by following the Lambertian bidirectional reflectance distribution function (BRDF) model. This is because the wavelength of the laser diode is significantly shorter than the roughness of most surfaces that are on the order of μm , which is generally true for most laser beams with wavelengths on the order of nm. Choe *et al.* also include an additional offset value I_a in their model, which accounts for a constant ambient intensity. This model was adopted by assuming constant surface reflectance and global brightness from the environment, and finding the least-squares fit of IR intensities to range, surface normal \mathbf{n} , and lighting direction \mathbf{l} . Thus,

$$I = I_a + \frac{\alpha}{r^2} (\mathbf{n} \bullet \mathbf{l}), \quad (3.10)$$

where $I_a = 62.3$ units of intensity and $\alpha = 5.90 \times 10^8$ unit-mm² from the fitted model, which resulted in a 0.814 R^2 value. This high coefficient of determination implies that the variance in over 80% of the intensity data can be explained by (3.10). The remaining variance in data can be attributed to random speckle γ_I and detector noise γ , which are assumed zero mean and independent for the fit. As Chow *et al.* [24] point out, indoor lighting conditions and object surface color have little effect on the detected IR intensity. Therefore, the constant ambient intensity can be attributed to background thermal radiation in the experimental setup. Note, in order to provide highly accurate range estimates of the transmitted dots, the least-squares solutions to the best-fit planes constructed in Section 3.2.2 were utilized to compute the distances from the sensor to the refined plane estimates for each pixel, separately.

3.3.3 Formulating the Speckle Noise Model

In order to retrieve relatively noise-free samples of the random speckle variable γ , time-averaged IR images were first determined. The speckle samples were subsequently generated by normalizing the average IR values by the corresponding predicted intensities, which were estimated using the fitted model in (3.10). Figure 3.5(A) depicts a histogram of the resulting random speckle deviates that remain after the pixels that generally contain either no dots or saturated intensities were filtered out, i.e. the lower 88.9% and upper 1% of pixels.

Laser power detected by a pixel is generally modeled as the sum of several exponentially distributed power (Rayleigh voltage) random variables [43], which results as a gamma distribution. A doubly truncated gamma was therefore fit to the deviates to recover the gamma distribution given by

$$f_{\gamma}(x; k, b) = \frac{1}{\Gamma(k)b^k} x^{k-1} e^{-\frac{x}{b}}, \quad (3.11)$$

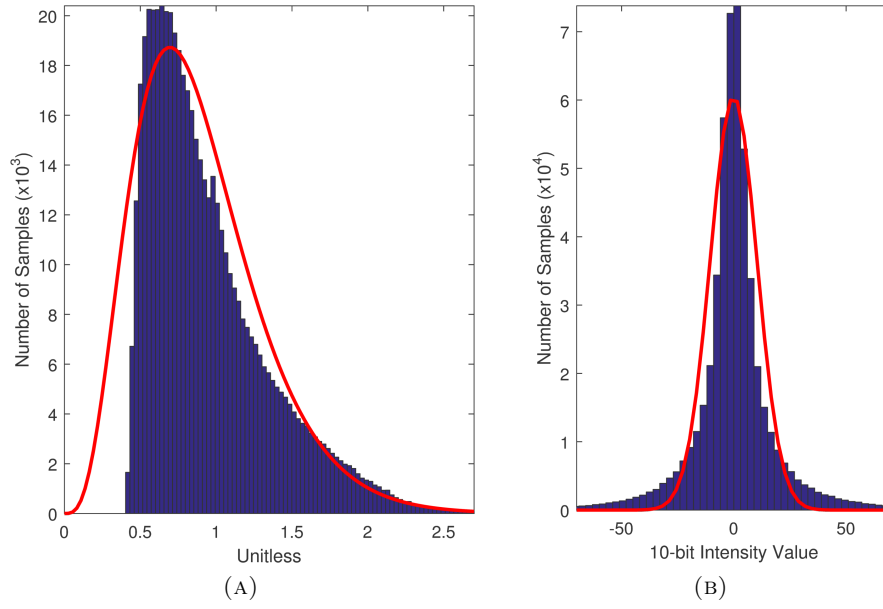


FIGURE 3.5: The multiplicative speckle distribution shown in (A) is unitless, and can be represented as a gamma distribution $\Gamma(4.54, 0.196)$. The additive detector noise distribution shown in (B) can be represented as a normal distribution $\mathcal{N}(-0.126, 10.4^2)$, and has units of 10-bit intensity.

where the shape k and scale b parameters were estimated to be 4.54 and 0.196, which produces a mean and standard deviation of $kb = 0.892$ and $\sqrt{kb} = 0.418$. The recovered gamma distribution is overlaid with the red line fit in Figure 3.5(A). Note that these fitted parameters support the previous assumption that the mean value for \tilde{n}_I is nearly zero. The scaled speckle noise $\tilde{\Gamma}$ can then be modeled by a gamma distribution, $\Gamma\left(k, \frac{I}{k}\right)$,

$$f_{\tilde{\Gamma}}(x; k, I) = \frac{1}{\Gamma(k)} \left(\frac{k}{I}\right)^k x^{k-1} e^{-\frac{kx}{I}}, \quad (3.12)$$

when the expected value of the speckle distribution γ is assumed to be unity, which is nearly true for the calibrated shape and scale parameters in (3.11).

3.3.4 Formulating the Thermal Noise Model

Lastly, random IR detector noise deviates were generated by subtracting the average IR image separately from each depth data stream. A histogram of the resulting additive deviates \tilde{n} is shown in Figure 3.5(B), which follows a Gaussian distribution

$$f_{\tilde{n}}(x; \mu_n, \sigma_n) = \frac{1}{\sigma_n \sqrt{2\pi}} e^{-\frac{(x-\mu_n)^2}{2\sigma_n^2}}, \quad (3.13)$$

where the fitted mean μ_n and standard deviation σ_n are -0.126 and 10.4 intensity units, respectively. The magnitude of the mean is less than one intensity unit and the standard deviation is expectedly small, which accounts for residual thermal noise present in the detector [12]. Note, shot noise due to quantum fluctuations is relatively low and less dominant than thermal noise, especially for pixels with higher intensity values, and is therefore absorbed in the composite Gaussian noise model [90]. Since the mean μ_n is near zero, the thermal noise can be assumed to have a distribution described by $\mathcal{N}(0, \sigma_n^2)$,

$$f_{\tilde{n}}(x; \sigma_n) = \frac{1}{\sigma_n \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma_n^2}}. \quad (3.14)$$

3.4 Physical Sensor Models

With the technical background and IR modeling presented in Sections 2.3 and 3.3, Kinect IR and depth images of any scene can be simulated by closely emulating the processes that construct the respective real images. In the following sections, the models for the proposed simulator are outlined, which was coded and tested in Matlab. The sensor specifications are also tunable input arguments, though the default parameters (Table 3.2) were adapted from Kinect literature. It should be understood that this simulation generates single image data of 3D scenes, which closely resembles the invention presented in [120].

3.4.1 Simulating IR Images

The IR image simulator incorporates many key characteristics of the real IR transmitter/receiver system, such as the laser projection system specified by [2, 3] and the CMOS sensor specified in the Aptina data sheet [5]. As mentioned earlier, the Kinect dot pattern uncovered by Reichinger [88] is also incorporated and is key to the simulator implementation. This implies that the model represents an idealized Kinect system with a perfect factory construction. Therefore, there is no rotation in the projected IR dots, the epipolar lines are rectified, and there are no gaps nor overlap in the tiled pattern. For now, the effects of lens distortion deviating the rectilinear projection of dots are excluded, though this could be accounted for in a future iteration of the simulator software. Note, since Kinect's downsampled raw IR images are 640×480 pixel grids, the simulation pads/crops from the 633×495 field of view that the idealized dot pattern spans in order to preserve the final depth image resolution. Here, padding is achieved by adding a repeated portion of the simulated pattern to the leftmost and rightmost columns of the original dot pattern.

In order to emulate the diffuser unit, line-of-sight vectors or *rays* are constructed from each cell (pixel) of the pattern that contains a dot [i.e. the white pixels in Figure 2.3(D)]. These rays in effect represent the IR laser system that transmits the pattern onto a

| | Kinect Specifications | | |
|--------------------|------------------------------|-----------------|--|
| IR/Depth Img Res | col: 640 pix | row: 480 pix | |
| Field of View | horz: 58.5° | vert: 45.6° | |
| Focal Length | horz: 571.4 pix | vert: 570.9 pix | |
| Operational Depths | min: 800 mm | max: 4000 mm | |
| | IR Image Simulator | | |
| Dot Pattern Res | col: 633 pix | row: 495 pix | |
| IR Dot Sub-ray Res | col: 17 sub-pix | row: 7 sub-pix | |
| | Depth Image Simulator | | |
| Correlation Window | col: 9 pix | row: 9 pix | |
| Ref Img Depths | min: 793.6 mm | max: 4285.5 mm | |
| Offset Disparity | min: 54 pix | max: 10 pix | |

TABLE 3.2: The default parameters for the Kinect specifications, IR image simulator, and depth image simulator are utilized to simulate IR and depth images.

given scene. Since the transmitted dots have a physical cross-sectional area, each ray is divided into a grid of sub-rays with c_{sub} columns and r_{sub} rows as depicted in Figure 3.6. Note, it is necessary to construct at least 8 columns of sub-rays to accommodate Kinect's sub-pixel disparity accuracy of $\frac{1}{8}^{th}$ of a pixel. For this report, the sub-ray grid was set to $c_{sub} = 17$ columns and $r_{sub} = 7$ rows. Though these tuning parameters can be altered at the discretion of the user, it was determined from rigorous experiments that a lower sub-ray column resolution adversely affects disparity refinement accuracy, whereas a higher row resolution is unnecessary and does not affect accuracy.

Next, let $\mathbf{p}_T = [u_T, v_T]^\top$ represent the coordinate of a sub-ray measured in pixels, and referenced to the transmitter coordinate system. The end point of each sub-ray $\mathbf{X}_T = [x_T, y_T, z_T]^\top$ is then constructed to intersect a flat wall at a maximum range D_{max} . These are computed by utilizing the lens equation and the horizontal f_x and vertical f_y focal lengths of the sensor

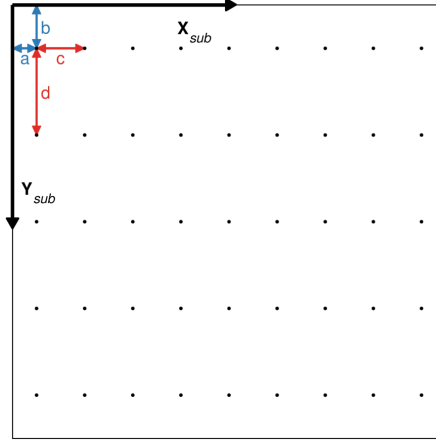


FIGURE 3.6: The physical cross-sectional area of a transmitted IR dot is simulated by dividing the dot into a grid of sub-rays. Here, $a = 1/2c_{sub}$, $b = 1/2r_{sub}$, $c = 1/c_{sub}$, $d = 1/r_{sub}$, and the origin of a simulated dot's coordinate system is positioned at the top left corner of the transmit pixel.

$$\mathbf{X}_T = D_{max} \cdot \left[\frac{u_T}{f_x}, \frac{v_T}{f_y}, 1 \right]^\top + [-b, 0, 0]^\top. \quad (3.15)$$

Kinect's horizontal and vertical focal lengths are determined to be $f_x = 571.4$ and $f_y = 570.9$ pixels, respectively, using the known image size and FOVs. Since it is assumed that the sensor and receiver coordinate systems are collocated, the X-axis coordinates of each transmitted sub-ray are shifted by the baseline distance b . A ray casting method [104] is then used to determine the 3D location where each sub-ray intersects the inputted CAD models. For each sub-ray that does in fact intersect the CADs before reaching D_{max} , the algorithm returns an output fractional distance δ_T less than 1, whereas a δ_T of 1 represents a non-intersecting sub-ray.

The locations of CAD model intersections are used to determine which parts of the simulated IR dots would be visible to the receiver. Here, receive sub-rays are constructed that emanate from the origin of the receiver coordinate system $(0, 0, 0)$ and end at the CAD model intersections, i.e. $\mathbf{X}_R = \delta_T \cdot \mathbf{X}_T$. The ray casting algorithm is redeployed and if the intersection with the CAD models is closer than expected, the returned fractional distance δ_R is less than 1, indicating occlusion. For a visual depiction of this, refer to the lower transmitter ray (solid magenta line) and the intersecting receiver ray (dashed blue line) in Figure 3.1. Only transmitted sub-rays that both intersect the CAD models and are not occluded contribute to the IR image. For each intersecting and unoccluded sub-dot, the 3D end point is finally projected into the receiver *pixel coordinate system* (Figure 3.7) via the lens equation to provide \mathbf{p}_R . It should be noted that the transmit and receive sub-rays require shifting to provide row and column indices with respect to the origin of the corresponding grids, which are both ordered from top to bottom and

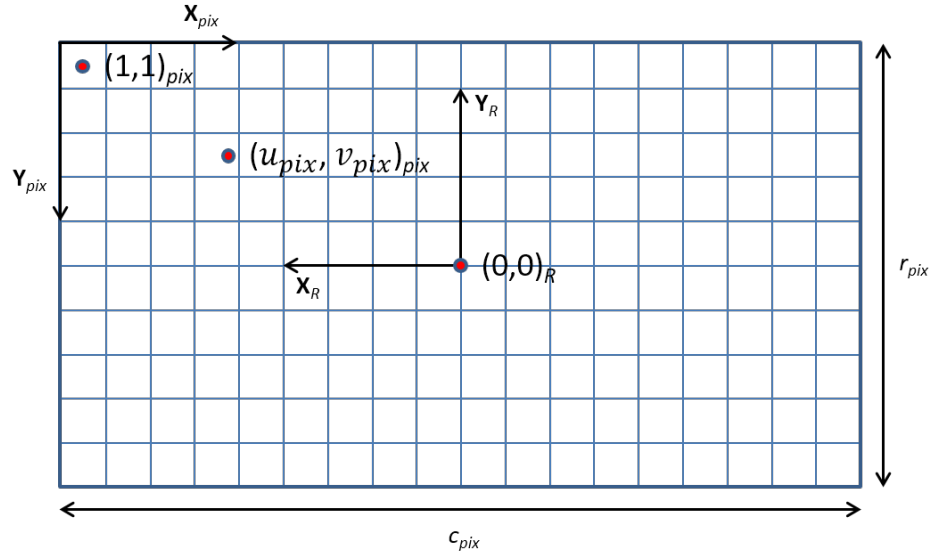


FIGURE 3.7: The sensor/receiver and pixel coordinate systems are displayed, where $(0,0)_R$ represents the origin of the receiver coordinate system and $(1,1)_{pix}$ represents the origin of the pixel coordinate system.

left to right. Also, it is assumed that transmit dot sizes are equal to the receive pixel spacing, which is roughly true according to [120].

The Kinect sensor is assumed to follow the linear property of optical imaging systems, where each dot is imaged simultaneously and integrated independently to create the simulated IR image. As follows, each dot's sub-ray is summed directly into the final grid, thereby circumventing the accessory 2×2 binning step. Again, each row of the transmitted dots lines up with its coinciding row in the received image; therefore the sub-rays within an IR dot can only spread horizontally, and cannot spread vertically. Similarly, the depth of dots intersecting a flat surface orthogonal to the Z-axis determines the disparity shift, and thus the amount of horizontal pixel splitting. To illustrate this point, let an IR dot with a sub-ray grid size of 17×7 intersect a flat, orthogonal wall. If the first 5 columns of sub-rays fall within one pixel, and the remaining 12 columns of sub-rays fall within the neighboring pixel, the two pixels would receive $\frac{5}{17} = 29.4\%$ and $\frac{12}{17} = 70.6\%$ of the full IR dot intensity [rightmost window of Figure 3.8(A)]. This design permits faithful predictions of sub-pixel refinement performance, which is employed in Section 3.4.2. In the case of a tilted surface, however, dots can be spread over more than two pixels, such as the rightmost window in Figure 3.8(B). Here, a single dot is spread to three pixels when a surface is rotated -80° about the Y-axis. Unlike the estimable flat wall projections, these imperfections and distortions in the measured dot pattern lead to systematically larger depth errors.

The analysis and generation of the empirical models presented in Section 3.3 are then used to simulate the IR intensity, speckle, and thermal noise that affect the receive IR detectors. More specifically, a modified version of (3.6) and (3.10) is used to simulate

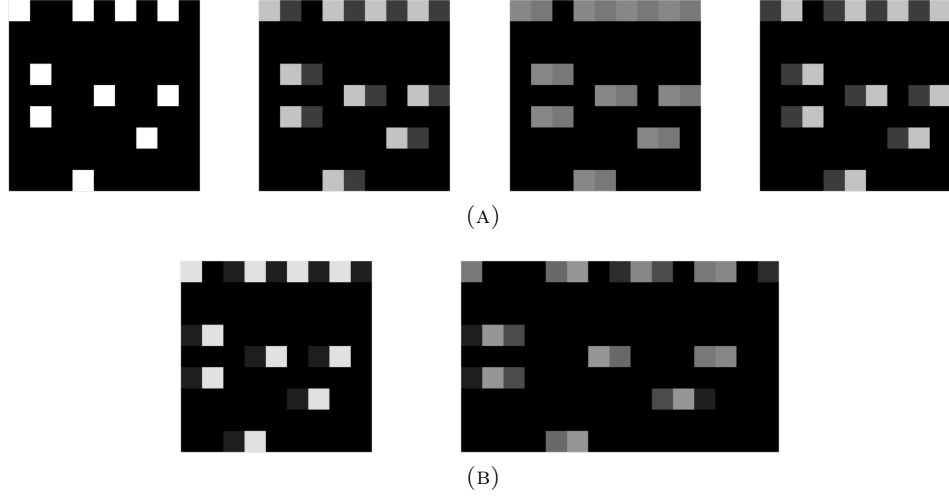


FIGURE 3.8: The leftmost window in (A) depicts IR dots intersecting only one pixel from a flat wall projection, whereas the other windows depict IR dots splitting two pixels. The rightmost window of (B) shows an example of the dot pattern spread over multiple pixels from an 80° tilted wall projection.

10-bit intensity values for each pixel Z that contains the sum of sub-rays (i, j) representing a dot's partial energy falling within it

$$Z = \gamma \frac{\alpha}{c_{sub} \cdot r_{sub}} \sum_{i,j} \frac{\mathbb{1}_{i,j} (\mathbf{n} \bullet \mathbf{l})_{i,j}}{r_{i,j}^2} + \tilde{n}. \quad (3.16)$$

Thus, the indicator function $\mathbb{1}_{i,j}$ takes on a value of 1 when sub-ray (i, j) is visible within the pixel, and a value of 0 when it is occluded. Random speckle γ is drawn from the gamma distribution in (3.11) and applied to each simulated dot, separately, whereas random detector noise \tilde{n} is drawn from the Gaussian distribution in (3.14) and applied to each pixel, separately. Note that the constant ambient offset I_a from (3.10) is left out since it depends on the experimental setup. Also, $(\mathbf{n} \bullet \mathbf{l})_{i,j}$ and $r_{i,j}^2$ depend on \mathbf{X}_R of the sub-ray and the unit normal of the intersected CAD facet. Finally, since sensors quantize the received IR images, intensity values Z are rounded to the nearest integer value.

The resulting IR image of a simulated block grid, similar to the experimental setup implemented in Section 3.2.3 to estimate edge error, is displayed in Figure 3.9. Note the existence of shadows along the left edge of some blocks that obstruct sections of the background wall. The known pattern of ‘wavy’ edges is also apparent due to the projection of the non-uniform distribution of dots on the surface of the scene. From this study, this is believed to be the direct cause of the standard lateral error observed in many reports (including [80] and [23]), which is further explored in Section 3.5.

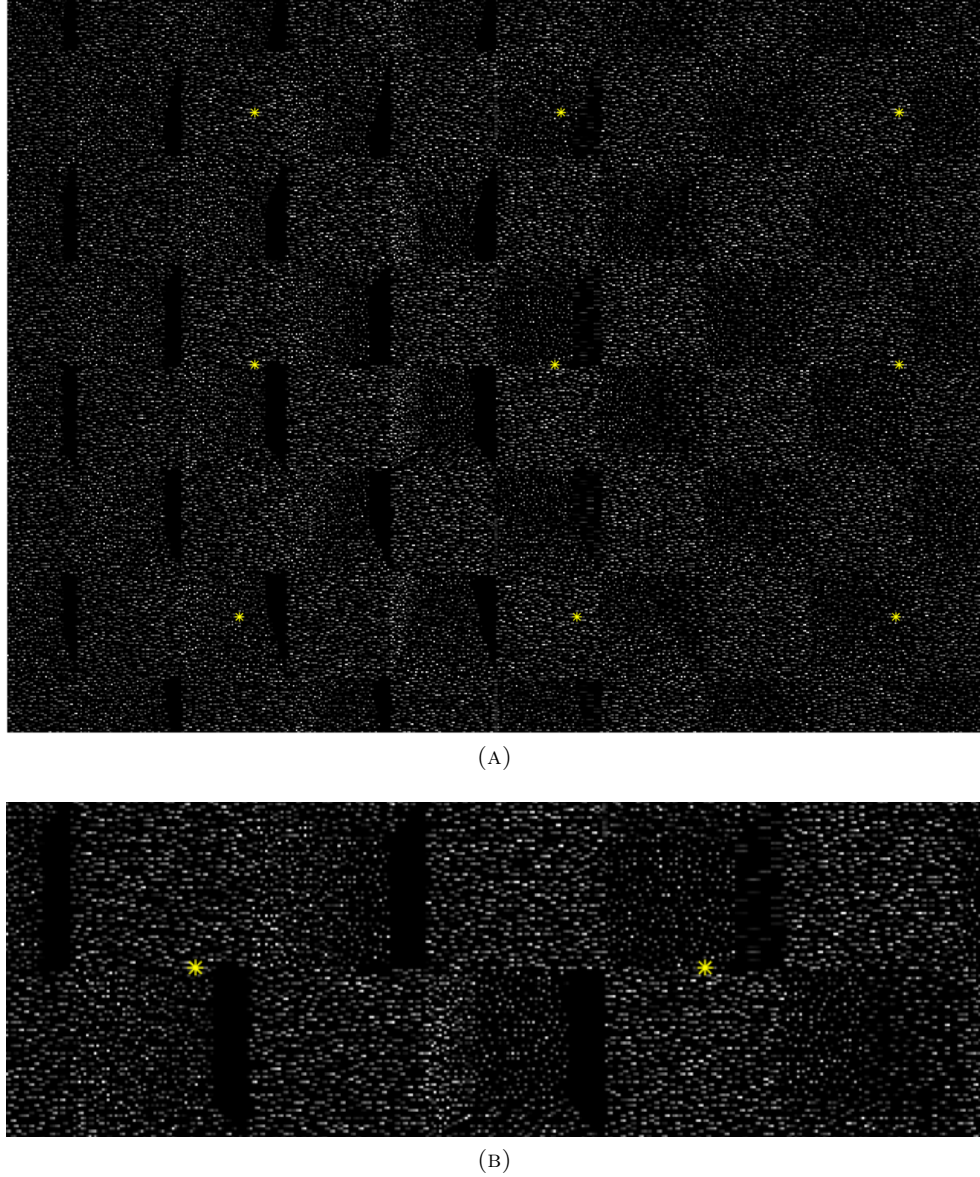


FIGURE 3.9: The simulated IR image of a block grid emulates distinctive shadowing and ‘wavy’ edges observed in real structured-light IR images. Note that IR dots projected on the surfaces of the blocks are brighter due to the inverse square law included in (3.16), and the bright dots (yellow stars) projected on the surfaces of the blocks and background represent different disparity shifts.

3.4.2 Simulating Depth Images

In this section, the methods for processing simulated IR images to generate depth images are provided. As mentioned previously, it is believed that Kinect employs a thresholding scheme that converts noisy IR images to binary images prior to the first disparity estimation step. Since the specifics of Kinect’s filter are unknown, for practical purposes the filter implemented on the device’s SoC is assumed to work sufficiently well, and contributes little to no error in the initial disparity estimates. Therefore, the model

utilizes post-filtered IR images for the correlation-based algorithm, where knowledge of pixels with unoccluded dots is given. It should be noted that while correlation-based disparity estimation utilizes filtered IR images, the sub-pixel refinement step processes the pre-filtered, noisy IR images in both the real and simulated Kinect systems.

As established in Section 2.3, a local, pixel-based correlation algorithm is used to compare windows of the measured binary image to a series of reference binary images. The shift that provides the best match is determined by the reference window that maximizes the cross-covariance C amongst the set of N windows tested. Given the default operational range of depths falling between 800 and 4000 mm, the minimum and maximum allowed disparities are computed to be 53.57 and 10.71 pixels via (2.1). Therefore the maximum disparity shift from any reference image is determined as $N = 45$ pixels.

Since the epipolar lines are coincident, the search for the best match between measured and reference image windows centered at (u_Z, v_Z) and (u_{ref}, v_{ref}) is constrained to be within the same row of pixels. This lateral shift translates to a change in disparity Δd_0 , where $u_Z = u_{ref} + \Delta d_0$ and $v_Z = v_{ref}$. As follows, the initial disparity is estimated by

$$\Delta d_0 = \arg \max_{n \in N} C_n, \quad (3.17)$$

where Δd_0 is restricted to an integer value of pixels, and the cross-covariance score C_n of a reference window centered at $(u_{ref,n}, v_{ref,n})$ is computed via

$$C_n = \sum_{(u,v) \in W_Z} [W_Z - \bar{W}_Z] \circ [W_{ref,n} - \bar{W}_{ref,n}]. \quad (3.18)$$

Here, W_Z and $W_{ref,n}$ represent measured and reference binary image windows, and \bar{W} denotes the average over the window. Due to the strong evidence supported by [56] and Table 2.2, the correlation window is set to a default size of 9×9 pixels.

After the initial integer disparity is computed, the estimate is refined to provide higher depth image accuracy. Sub-pixel refinement estimates an inter-window fractional disparity Δd_s by determining where the IR dots split pixels within the window of an assumed constant depth, i.e. an orthogonal and flat surface segment. Unfortunately, the refinement method is not provided in the PrimeSense documentation, so it is unclear what matching metric is employed. Though there are many ways to measure the matching cost, the Sum of Absolute Differences (SAD) is generally considered the most common pixel-based method due to its computational simplicity [103]. Therefore, the SAD cost between the measured and predicted image is utilized in the simulated model. And due to the analysis supported by [56], [72], and Figure 2.5, the default interpolation factor is set to 8 sub-pixel levels.

In contrast to the coarse disparity estimation model, which employs binary reference images, a series of non-noisy IR intensity reference images are utilized as a lookup table for the sub-pixel refinement step. More explicitly, reference images of a flat wall at various depths are pre-processed, which are determined using (2.1) and all integer disparity values between the minimum and maximum disparities. Given the IR intensities and dot pattern projection from the reference image corresponding to the initial integer disparity estimate Δd_0 , an approximate prediction of all inter-window pixel split possibilities can then be supplied [e.g. Figure 3.8(A)]. The total pixel displacement is ultimately computed as the aggregate of the two disparity estimates, i.e.

$$\Delta d = \Delta d_0 + \Delta d_s. \quad (3.19)$$

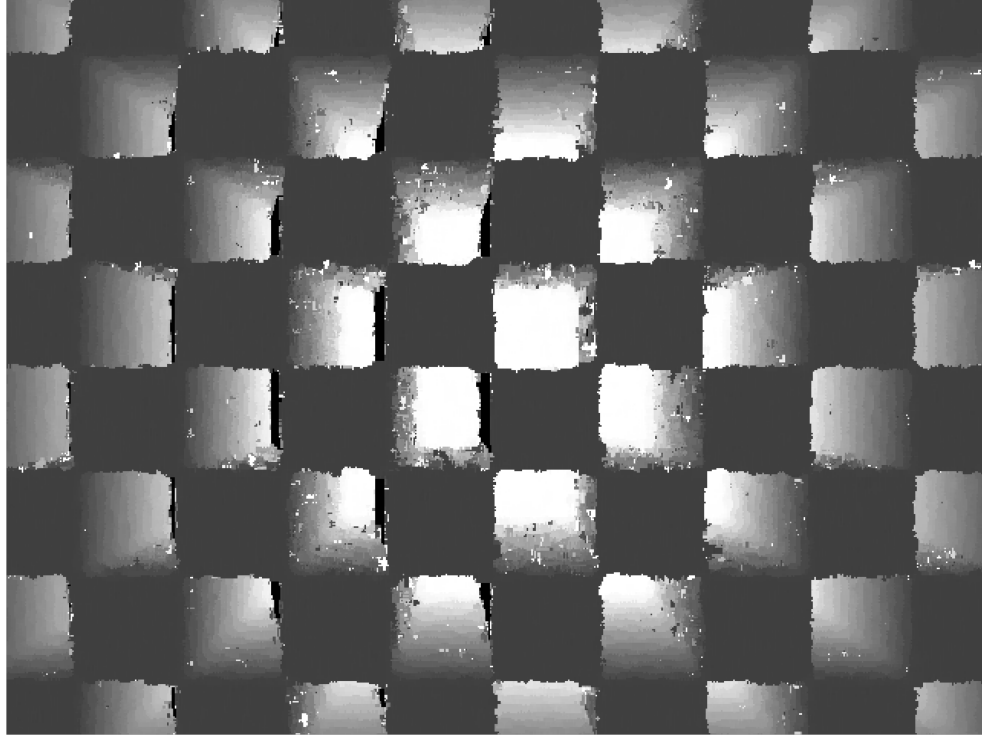
It is worth noting that the non-linear spacing of reference images generated for the depth image simulator does in fact agree with the increase in spacing mentioned in [38].

In order to compute depth from the total disparity estimate Δd , (2.1) was derived to estimate the change in depth with respect to a given reference image. Thus, a pixel's depth estimate is computed as

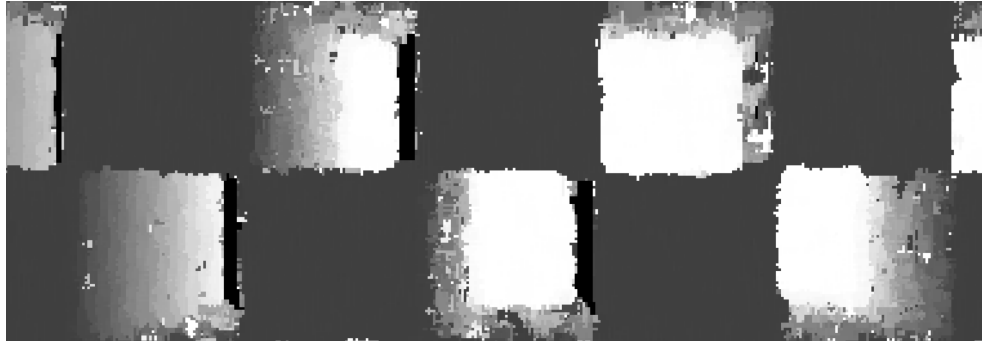
$$D = f_x \frac{b}{d_{off} + \Delta d}, \quad (3.20)$$

where d_{off} represents the offset disparity of a reference image. Since the transmitter is physically positioned to the right of the receiver (Figure 2.2), a positive Δd implies a pattern shift to the right, which translates to an object mapped closer to the sensor when compared to the reference depth. Alternatively, a negative Δd implies a shift to the left and an object mapped further away. It should be noted that by limiting total disparity estimation to an $\frac{1}{8}^{th}$ of a pixel with equal intervals, allowable output disparity is in essence quantized similar to the way Kinect quantizes depth images into 11-bit values (Figure 2.5). The simulated depth error due to disparity quantization is therefore comparable to the real-valued depth errors from the Kinect system.

The resulting depth image of the simulated block grid is displayed in Figure 3.10, which portrays key qualitative features seen in real depth images. For instance, edges of the blocks with straight and flat sides are imaged as the distinctly wavy edges. Artificial bridges and gaps are also introduced where block corners meet, which should otherwise intersect at a single point. These characteristics exist in Kinect because of imperfect matching estimates due to noise and significant depth variation within a processing window. Moreover, depth shadows are observed when there aren't any intersecting dots within a tested 9×9 correlation window, and as such, the algorithm cannot infer depth.



(A)



(B)

FIGURE 3.10: The simulated depth image of a block grid propagates the shadows and ‘wavy’ edges by processing the noisy IR image. Other distinctive characteristics of real structured-light depth images are also present, such as artificially induced bridges and gaps near the corners of each block.

3.5 Model Validation

In addition to a decisive qualitative comparison between simulated and real IR and depth images, quantitative comparisons of axial and lateral error are provided for three different simulated and experimental data sets. These data sets consist of 100 frame depth image streams from 1) a series of flat walls parallel to the focal plane at various depths, 2) a series of flat walls with varying tilt angles, and 3) a flat-edged, square object placed at two distances from the background wall and at varying distances from the sensor. Errors obtained from the simulated scenes are also compared to the three models discussed in

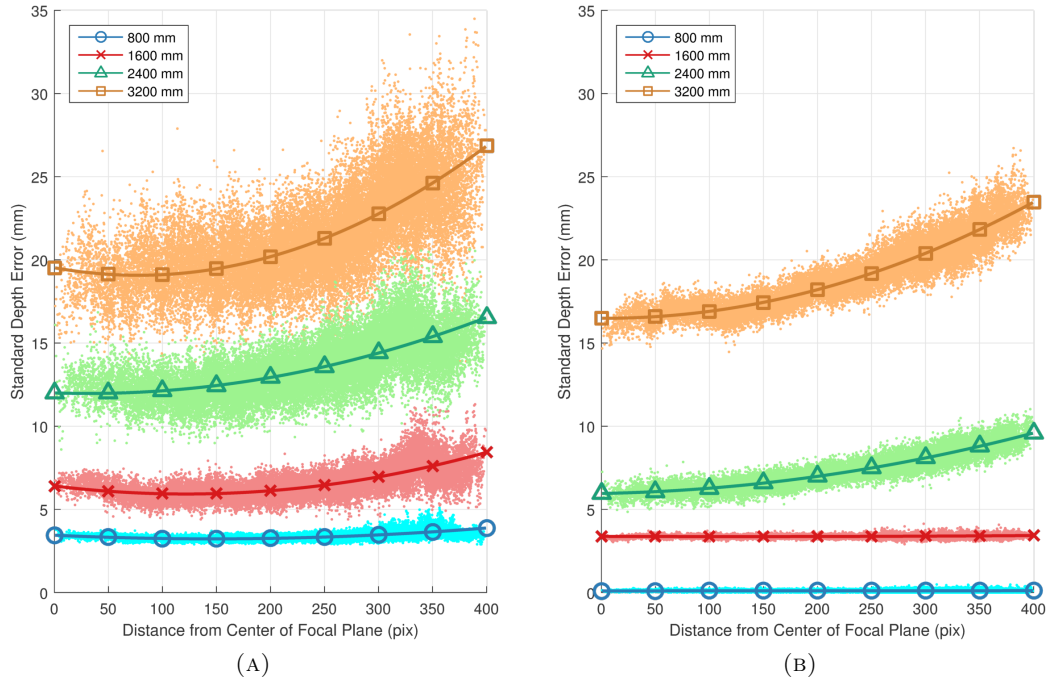


FIGURE 3.11: The standard error in depth estimation (mm) as a function of radial distance (pix) is plotted for the (A) experimental and (B) simulated data sets of flat walls at various depths (mm). The experimental standard depth error increases faster with an increase in radial distance due to lens distortion.

Sections 3.1 and 3.2 to further support validation. Since the experimental results derived from these models are highly dependent on the properties of the actual environment and sensor deployed, the resulting empirical models are not expected to perfectly align with the simulated results. However, the environment present in the experiments does coincide with the setup utilized to determine the speckle and detector noise distribution statistics in Section 3.3. It is also important to note that while some experimental results may align with previously constructed error models, these models incorrectly assume homogeneity and only predict the average error statistics. The proposed model, on the other hand, captures these error statistics by replicating salient and inhomogeneous features of the system.

The standard deviation of depth error was first examined for the series of flat walls parallel to the focal plane. The same best-fit planes constructed for the error models in Section 3.2.2 were again used as the basis for depth ground truth. In Figure 3.11, the standard depth errors are plotted separately for each pixel, which were computed using the 100 frame data sets. As seen here, the standard depth error as a function of radial distance increases at a faster rate and is more widely spread in the experimental data. Since pincushion distortion stretches the dot pattern at further distances, it follows that Kinect's correspondence algorithm suffers when matching templates to the reference IR images. Focus was therefore placed on the standard depth error for the collection of pixels

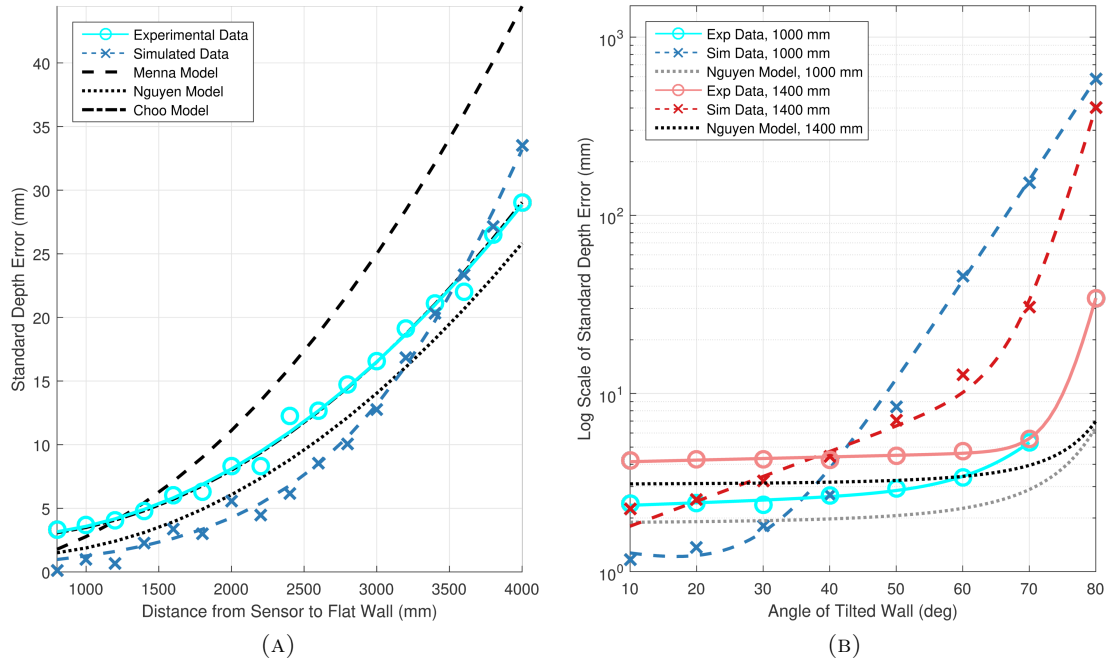


FIGURE 3.12: The standard depth error (mm) as a function of (A) flat wall depth (mm) and (B) tilted wall angle (deg) is plotted for the experimental and simulated data sets. Three models are also plotted in (A), whereas only the Nguyen model can be plotted in (B).

closer to the optical center (closest 10%) for the subsequent quantitative comparisons to avoid having to account for lens distortion.

The standard depth error of the center pixels is plotted for each data set in Figure 3.12(A) along with the Menna, Nguyen, and recalibrated Choo models for flat wall depths between 800 and 4000 mm at intervals of 200 mm. As seen in this figure, the simulated results closely comply with the experimental results for the range of depths tested, though standard error is perhaps slightly underestimated for closer flat walls. The recalibrated Choo model fits to the experimental depth error nearly exactly, as expected. Additionally, the Nguyen model appears to fit closely with the measured standard errors, whereas the Menna model noticeably overestimates. Nonetheless, the overall quadratic axial error trend is present in each curve of the figure, which can certainly be attributed to the stereo system. Flat surfaces do not exhibit dot occlusion nor dot spreading, however, so depth errors arise mainly when noisy IR intensities lead to imperfect matches from the SAD algorithm in the simulation. And since Chow *et al.* [25] demonstrate that error due to disparity quantization is minimal, the experimental standard errors near the center of the focal plane are decidedly attributed to the sub-pixel refinement step.

Next, the experimental and simulated standard depth errors of close pixels for each tilted wall data set are plotted in Figure 3.12(B). Two data setups were constructed to fix the distance between the sensor and the center of the wall at 1000 mm and 1400 mm, and consist of the plane's normal varying between nearly orthogonal to nearly parallel to the

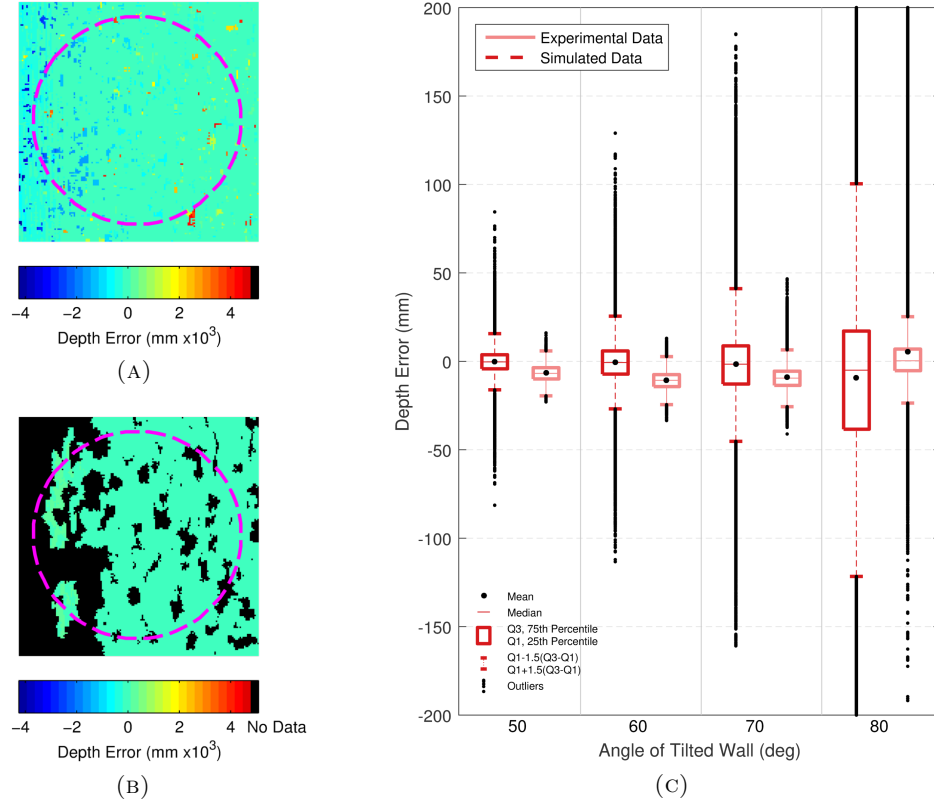


FIGURE 3.13: The (A) simulated and (B) experimental depth error heat maps of pixels close to the optical center (within the dashed magenta circles) for an 80° tilted wall, 1400 mm from the sensor are displayed, which demonstrate that simulated depth images contain many outlier depth estimates from mismatched correlation windows. The Tukey box plots for $50^\circ - 80^\circ$ tilted walls, 1400 mm from the sensor are shown in (C), which demonstrates the similarity in depth estimates for both image sets.

optical axis. Ground truth was again acquired by finding the root-mean-square error (RMSE) solution of the best-fit plane to the experimental depth streams. Since only the model provided by Nguyen *et al.* accounts for surface tilt, their model is also plotted for comparison and validation. The figure shows that the experimental and simulated errors follow closely together along with the Nguyen model at tilt angles below 50° . The Nguyen model underestimates the experimental results at steeper angles, though, which is likely attributed to the difference in actual reflective properties (albedo) between their tested surface and the surface imaged for this study. The simulated results, on the other hand, tend to overestimate standard depth errors at steep angles. The discrepancy can be explained with Figures 3.13(A) and 3.13(B), which depict errors plotted as a focal plane heat map collected for a surface at 80° and 1400 mm from the sensor. Outlier values (yellow and blue pixels) peppered in the simulated images tend to skew the standard error, whereas the experimental depth errors tend to stay more consistent but contain significant regions with no result. As explained earlier, depth error becomes more appreciable for closer surfaces with steeper angles because dots are horizontally spread. It is postulated that a region growing algorithm, believed to be employed in the Kinect

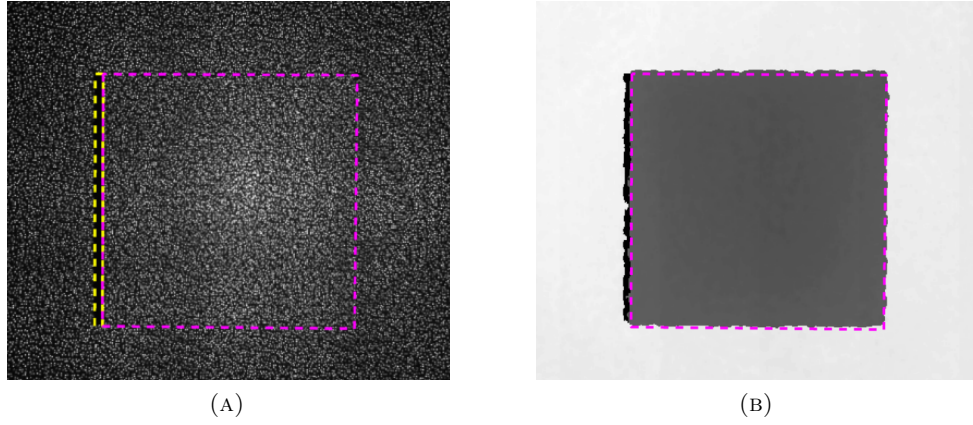


FIGURE 3.14: (A) A novel method to fit the edges of a block is presented, which utilizes the IR shadow caused by the surface of the block obstructing a section of the background wall. (B) The resulting fit clearly shows a bias in the estimated edges when overlaid on the corresponding experimental depth image.

system, is able to smooth poor matches from the correlation-based disparity estimation step (though not in all cases). The simulator also occasionally finds poor matches when the correlation peak is broad, but does not smooth pixels nor set them to a missing depth value. Upon examination of the Tukey box plots for the experimental and simulated depth errors from the 1400 mm tilted wall data sets in Figure 3.13(C), it is clear that the histograms are similar, and primarily differ in the tails of the distributions represented by outlier values. Here, the tails in the experimental distributions are truncated due to smoothed or ‘no result’ pixels that produced correlation peaks below a threshold.

Lastly, the lateral error component of Kinect is examined by determining the distances between the true and estimated edges of a square object placed at different distances from the wall and the sensor. Here, a new method to determine the ground truth edges is presented. The experimental IR data in Figure 3.14(A) is used to obtain ground truth by first fitting planes to the block object and wall surfaces, and then by fitting a boundary to the object. To compute the box boundary, a thin boarder is first fitted around the shadow (dashed yellow line) caused by the left edge of the object, and the right edge of the shadow box then initializes the left edge of the object box. The left corners of the object box combined with the knowledge of the object width and RMSE-fitted plane depth finally determine the remainder of the box boundary (dashed magenta line). This fitted boundary is overlaid on the corresponding depth image in Figure 3.14(B).

Upon visual inspection of Figure 3.14(B), it is clear there is a shift between the boundary determined from the IR image (used as truth) and the estimated edges derived from the experimental depth image. Thus, another metric referred to as *bias of edge error* is provided, which refers to the constant offset of the edge location errors. Konolige *et al.* [56] suggest this shift is the result of an IR camera-to-depth offset. After removing this offset from the experimental data sets, however, there remains a bias of error, as

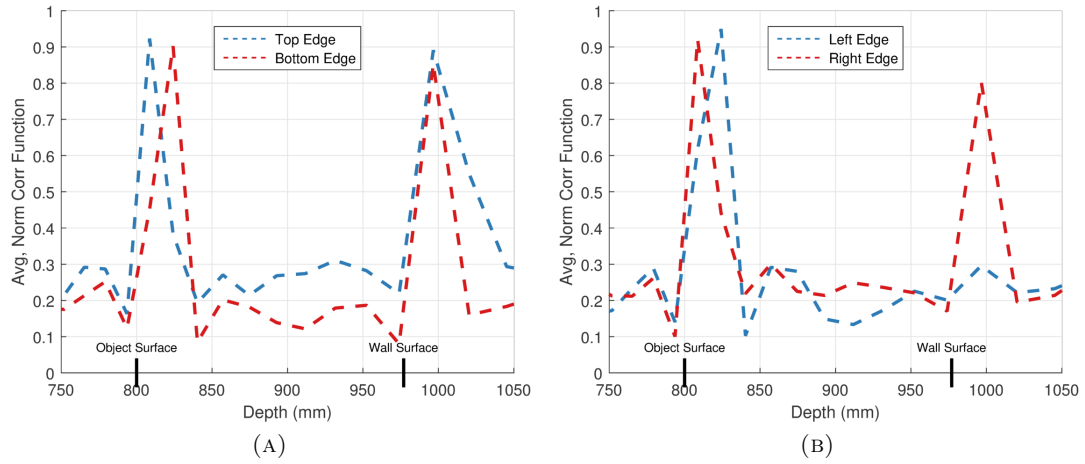


FIGURE 3.15: The normalized and averaged correlation functions for simulated object edges are displayed. Note the two peaks for the top, bottom, and right edges, which cause depth ambiguity that result in the bias of edge error.

supported by the solid line plots in Figures 3.16(A), 3.16(B), 3.16(C), and 3.16(D). More precisely, each experimental edge generally falls outside of the ground truth border (represented as a positive bias), which is consistent with the simulated data sets (dashed lines) in the respective figures. It is therefore proposed that the bias is the result of an imperfect stereo matching algorithm. This is easily explained at the left edge of the box when the correlation window operates in the shadow of the IR image and interpolates depth estimates. The top, bottom, and right edges have alternatively been determined to produce two correlation peaks corresponding to dot pattern segments on the object versus wall surfaces. This is demonstrated by the correlation functions displayed in Figure 3.15. Here, the correlation values computed separately for each edge pixel were normalized and averaged for the simulated edges of the 800 mm sensor-to-object, 177 mm object-to-wall data set. The higher peak manifests semi-randomly, which results in a mix of disparity estimates that are biased towards either the object or wall surface at the boundaries.

Imperfect matches also contribute to the standard errors of the estimated horizontal [Figure 3.16(E)] and vertical [Figure 3.16(F)] edges. These standard edge errors give a quantitative measure of the wavy edges observed in Figures 3.9 and 3.10. The Nguyen and Menna (not displayed) models tend to underestimate lateral error for closer objects, and the recalibrated Choo model appears to overestimate lateral error for further objects. More importantly, these models do not account for varying object-to-background surface distances, which is a clear influence on edge estimation and lateral error.

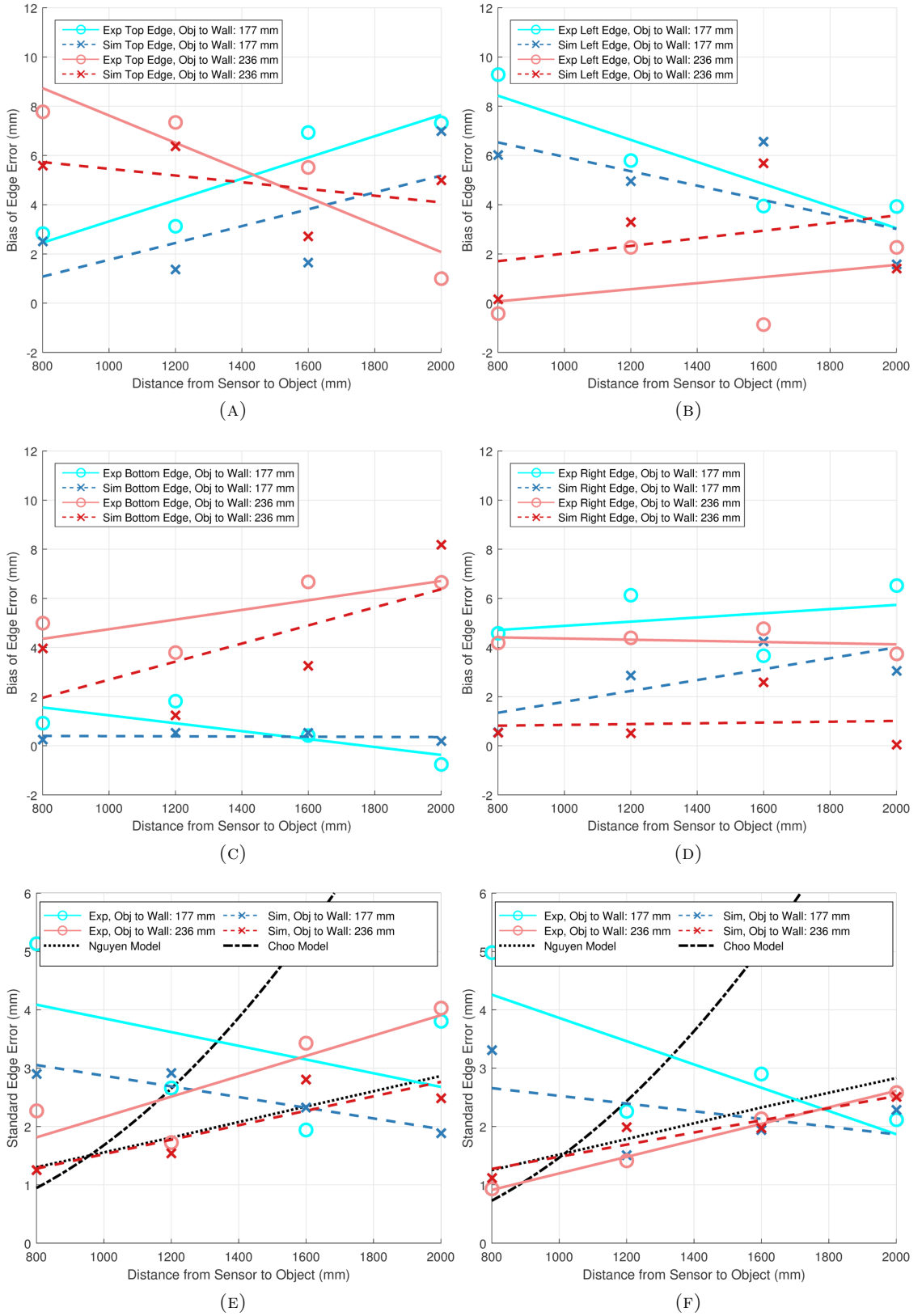


FIGURE 3.16: The horizontal bias of edge errors (mm) for the (A) top and (C) bottom edges, and the vertical bias of edge errors (mm) for (B) left and (D) right edges of the block are displayed. The standard edge errors (mm) for the (E) horizontal and (F) vertical edges are also displayed.

4

OPTIMAL POSE ESTIMATOR CRITERIA

The theory of information became the cornerstone of cybernetics because the latter deals with “the study of systems of any nature that are capable of receiving, storing, and processing information and utilizing it for control”.

– Aleksandr Kondratov (1969)

This chapter is structured as follows: Section 4.1 establishes an information theoretic framework and formulates the proposed method to compute the Fisher information and corresponding lower bound on the variance of any unbiased pose estimator for unique structured-light sensor measurement data. Section 4.2 provides the theory behind the loss of information when IR images are processed into depth images. Finally, Section 4.3 defines the criteria for an optimal, i.e. best mean square error estimator for structured-light sensors.

4.1 Cramér-Rao Bound of Structured-Light Sensors

In classical parameter estimation theory, the amount of information contained in a random observation is often quantified to establish the fundamental limit in performance of a potential estimator. For this dissertation, the Fisher information is calculated to measure the amount of information that observed structured-light IR images of an object carry about an unknown 6D object pose, and utilize its relationship to the Cramér-Rao bound (CRB) to express a lower bound on the variance of any unbiased pose estimation scheme. This bound is later shown to be approximately attained by the newly proposed parameter estimation algorithm in Chapter 6 that operates on the raw IR images – even when only sparse measurement sets are available – thereby supporting near optimality.

Studies have been performed where analytical expressions for the CRB of estimators that use point clouds from monostatic laser imaging systems (e.g. ToF and lidar) have been

provided [10, 75, 82]. The range error statistics of these sensors are nearly independent and homogeneous [36], which allow point set registration (PSR) methods to utilize approximate joint PDFs of the transformed point clouds. However, due to the complexity of depth processing required for structured-light sensors such as the Kinect, and because the error in depth pixels is inhomogeneous and correlated from spatial-multiplexing light coding, it is not possible to compute the true joint density of the depth images or corresponding pseudo-measurement point clouds without highly computational and memory intensive Monte Carlo methods [100]. One would need to non-parametrically generate the joint PDF histogram for thousands of pixels from a sizable ensemble of noisy depth images for each object and pose. It is therefore prohibitive to compute the Fisher information for structured-light depth images and consequently for the corresponding pseudo-measurement point clouds.

On the other hand, because the PDFs for the IR pixels have common independent speckle and thermal noise statistics, the Fisher information and Cramér-Rao bounds of the IR images can be computed without the need for Monte Carlo techniques. As will be shown, the joint PDF for the IR image is understood because each pixel's density is a member of the same family of distributions with a mean that depends on the sensor and its interaction with the object. Though an explicit analytic expression for the Fisher information of the pose cannot be written, it can be expressed as a function of the mean image and its sensitivity with respect to the pose via the IR image simulation model. A method to compute the Fisher information matrix for the pose vector given the collection of IR pixels is thus provided, which utilizes the simulator, developed speckle and thermal noise statistics, and IR pixel independence.

4.1.1 Fisher Information Matrix for IR Image

Let the pose of an object be defined as the augmentation of the 3D rotation and 3D translation the object presents to the sensor, $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_6]^\top$, where θ_1, θ_2 , and θ_3 refer to the three Euler angles with respect to the initial orientation in the model coordinate system, and θ_4, θ_5 , and θ_6 refer to the three Euclidean distances with respect to the origin of the sensor coordinate system. The pose domain is defined as the augmentation of the entire group of Euclidean rotations, denoted by $SO(3)$, with the volume the object may occupy within the confines of the FOV, denoted by V_{FOV} . Next, let the IR image be defined as the collection of P pixels at the receiver's focal plane, $\mathcal{Z} = [Z_1, Z_2, \dots, Z_P]^\top$. When the image contains enough informative pixels, the variance of any unbiased estimator $\hat{\boldsymbol{\theta}}(\mathcal{Z})$ of the data set is known to be bounded below by the inverse of the Fisher information matrix (FIM)

$$\text{CRB}_{\mathcal{Z}}(\boldsymbol{\theta}) \triangleq \mathcal{I}_{\mathcal{Z}}^{-1}(\boldsymbol{\theta}) \leq \text{Cov}_{\mathcal{Z}}(\hat{\boldsymbol{\theta}}(\mathcal{Z})), \quad (4.1)$$

where the matrix inequality $\text{Cov}_{\mathcal{Z}}(\hat{\boldsymbol{\theta}}(\mathcal{Z})) \geq \mathcal{I}_{\mathcal{Z}}^{-1}(\boldsymbol{\theta})$ implies that $\text{Cov}_{\mathcal{Z}}(\hat{\boldsymbol{\theta}}(\mathcal{Z})) - \mathcal{I}_{\mathcal{Z}}^{-1}(\boldsymbol{\theta})$ is positive semidefinite. This limit is referred to as the Cramér-Rao bound, and is derived using the Cauchy-Schwarz inequality [106]. The 6×6 FIM is defined by

$$\mathcal{I}_{\mathcal{Z}}(\boldsymbol{\theta}) = \mathbb{E} \left[s_{\mathcal{Z}}(\mathbf{z}; \boldsymbol{\theta}) s_{\mathcal{Z}}(\mathbf{z}; \boldsymbol{\theta})^{\top}; \boldsymbol{\theta} \right], \quad (4.2)$$

where s is the score function

$$s_{\mathcal{Z}}(\mathbf{z}; \boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} \ln f_{\mathcal{Z}}(\mathbf{z}; \boldsymbol{\theta}). \quad (4.3)$$

Note, the notation $f(\cdot; \boldsymbol{\theta})$ implies that the function f is a function of the non-random, unknown pose $\boldsymbol{\theta}$. The joint likelihood function of the pixel measurements $Z \in \mathcal{Z}$ given the unknown pose parameter vector $\boldsymbol{\theta}$ is defined as the product of the individual measurement functions

$$f_{\mathcal{Z}}(\mathbf{z}; \boldsymbol{\theta}) = \prod_{p \in P} f_{Z_p}(z_p; \mu_p(\boldsymbol{\theta})) \quad (4.4)$$

since the measurements are assumed as independent. Note, when pixels don't receive any reflected IR energy, $f_{\mathcal{Z}}(\mathbf{z}; \mu(\boldsymbol{\theta}))$ reverts to the thermal noise distribution (3.14). The expected intensity $\mu(\boldsymbol{\theta})$ of a pixel that images the surface of an object under pose $\boldsymbol{\theta}$ is given by an extension of (3.10)

$$\mu(\boldsymbol{\theta}) = c\rho \iint_{pix} \frac{\mathbb{1}((\tilde{u}, \tilde{v}), \boldsymbol{\theta}) [\mathbf{n}((\tilde{u}, \tilde{v}), \boldsymbol{\theta}) \bullet \mathbf{l}]}{r((\tilde{u}, \tilde{v}), \boldsymbol{\theta})^2} d\tilde{u}d\tilde{v}, \quad (4.5)$$

where (\tilde{u}, \tilde{v}) represents a point in receive pixel coordinates (i.e. a direction), and pix represents the region of directions for a pixel. The indicator function $\mathbb{1}$ takes on a value 1 when (\tilde{u}, \tilde{v}) is a direction in which a beam's partial reflected power is captured by pix . Alternatively, the indicator function takes on a value 0 when the partial reflected energy is occluded or when the receive direction was not illuminated due to the distribution and spacing of the light pattern. $\mu(\boldsymbol{\theta})$ can then receive a fraction of the full dot energy, depending on the depth and slope of the object surface. More specifically, a one-pixel-wide transmit IR dot can either fully intersect a single receive pixel or split adjacent receive pixels when projected on a flat surface, or can be spread over two or more receive pixels when projected on a tilted surface. If pix captures no dot energy, then $\mu(\boldsymbol{\theta}) = 0$ and would therefore only be corrupted by thermal noise. It is important to note that all of these effects are captured by the simulator presented in Chapter 3.

The assumption that intensity measurements are independent is supported by a theoretical analysis on the roughness of surfaces, as well as an experimental evaluation performed

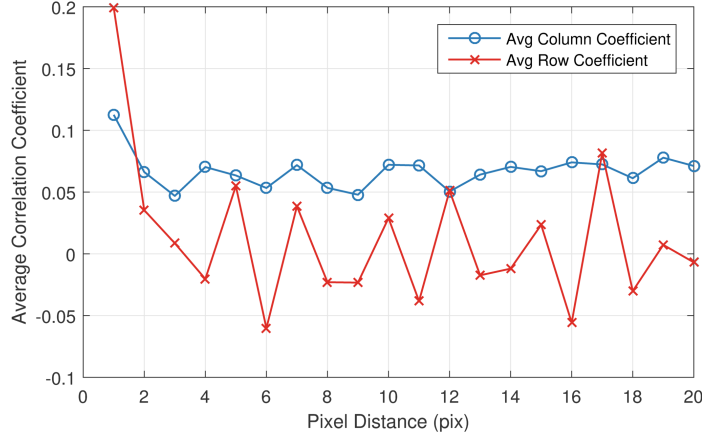


FIGURE 4.1: The average Pearson correlation coefficients are computed for pixel pairs containing dot energy along neighboring rows and columns for a 100 frame sequence of real IR images of a flat wall at a depth of 2400 mm.

on real IR images recorded by a Kinect for Windows sensor. It can be shown that IR dots are spaced far enough at typical operating ranges so that the autocorrelation of intensities are near zero. As Kayahan *et al.* [54] explain, speckle size is directly related to the wavelength λ of the laser beam when the wavelength is significantly shorter than the surface roughness, and the average speckle size can be approximated by

$$d_{\text{speckle}} = 1.2\lambda \frac{D}{d_{\text{dot}}}, \quad (4.6)$$

where D is the depth of the reflecting surface and d_{dot} is the diameter of the dot. In the case of the Kinect, the illuminator uses an 830nm laser diode [3], and measures a 58.5° horizontal FOV [4] with a 571.4 pixel focal length, denoted by f_x . Though the diameter of the dot increases with an increase in depth, the ratio remains constant, and after a 2×2 binning a dot diameter fills a single pixel when reflected on an ideal, flat surface [120]. Since the focal length measured in pixel widths is given by $f_x = \frac{D}{d_{\text{dot}}}$, the speckle correlation distance is approximately $d_{\text{speckle}} = 0.57$ mm, which corresponds to 0.41 pixels at Kinect's nearest operating range of 800 mm and 0.14 pixels at 2400 mm. Since the closest distance between unique dots in the projected pattern is at least two pixels, this corresponds to a minimum of $\frac{2}{d_{\text{speckle}}} = 4.9$ and 14.2 correlation distances, respectively, and therefore two unique dots are expected to be uncorrelated. This low-level correlation is confirmed in Figure 4.1, which plots the average Pearson correlation coefficients for IR dot pairs that were reflected off a flat surface at 2400 mm. Here, individual correlation coefficients were computed for each pair of pixels containing dot energy along neighboring rows or columns from a 100 frame sequence of recorded IR images. The highest correlation coefficients correspond to the cases when a single, pixel-wide dot illuminates a patch that straddles two adjacent pixels. However, these coefficients are also relatively low since one pixel width/height corresponds to $\frac{1}{d_{\text{speckle}}} = 7.1$ correlation distances, and because random detector noise is present in each pixel separately. While adjacent pixels may

provide slightly more sub-pixel shift information from higher degrees of speckle correlation at closer distances via the intensity ratio, the majority of information is derived from captured energy of dots that are on average separated by many pixels. It should also be noted that though the composite Fisher information estimate depends on all pixels in the IR image, many informative pixels exist near the border of objects, as is shown later in Section 4.1.2. Therefore dots straddling pixels within the cross-sectional area of the object contribute a relatively small amount to the composite Fisher information estimate for most object shapes.

Pixel independence and the natural logarithm operation in (4.3) then turn the product in (4.4) into a sum. Thus, the information is additive, and the FIM can be computed as

$$\mathcal{I}_{\mathbf{Z}}(\boldsymbol{\theta}) = \sum_{p \in P} \mathcal{I}_{Z_p}(\boldsymbol{\theta}). \quad (4.7)$$

Though the statistics for the speckle and thermal noise random variables are assumed to be identical across the focal plane, the mean of the intensity distributions for each pixel is relegated by the IR pattern and the position and orientation of the object and background surface. Moreover, each pixel observes a different subset of the object along a different direction defined by the pixel location. Therefore the joint PDF of each IR image – the observable random vector applied to the information metric – is unique to the object’s pose. Monte Carlo methods could potentially be used to approximate (4.7) due to the reduced complexity from IR pixel independence, though it would still be costly, and unnecessary, to generate non-parametric PDFs of the multivariate functions $f_Z(z; \mu(\boldsymbol{\theta}))$ for each pixel in the IR image. The known statistics and the simulation model can instead be made of use to generate a mean or non-noisy image given the object and pose. The FIM for the pose can then be reparameterized for each pixel using the chain rule, and represented as a function of the mean via

$$\mathcal{I}_{\mathbf{Z}}(\boldsymbol{\theta}) = \mathcal{I}_{\mathbf{Z}}(\mu) \frac{\partial \mu(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \frac{\partial \mu(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}^{\top}, \quad (4.8)$$

where $\frac{\partial \mu(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ is the Jacobian, i.e. the sensitivity of the mean intensity to the pose. Note, for one pixel Z , this matrix is rank one and is not invertible for a CRB expression. However, when $\dim(\boldsymbol{\theta}) = 6$ or more pixels with dot energy are provided, each with a unique view of the object, (4.7) is in fact invertible.

In order to define the Fisher information for the mean of a pixel $\mathcal{I}_Z(\mu)$, let us first consider two limiting cases for the shape of the distribution defined in (3.8): when additive thermal noise is zero, the detector output is gamma, and when the speckle noise is constant (which occurs when the shape parameter k goes to infinity and $\sigma_{\gamma}^2 = 0$), the detector output is Gaussian shifted by the mean μ . Both of these distributions are members of the natural exponential family (NEF) class, a subset of the exponential

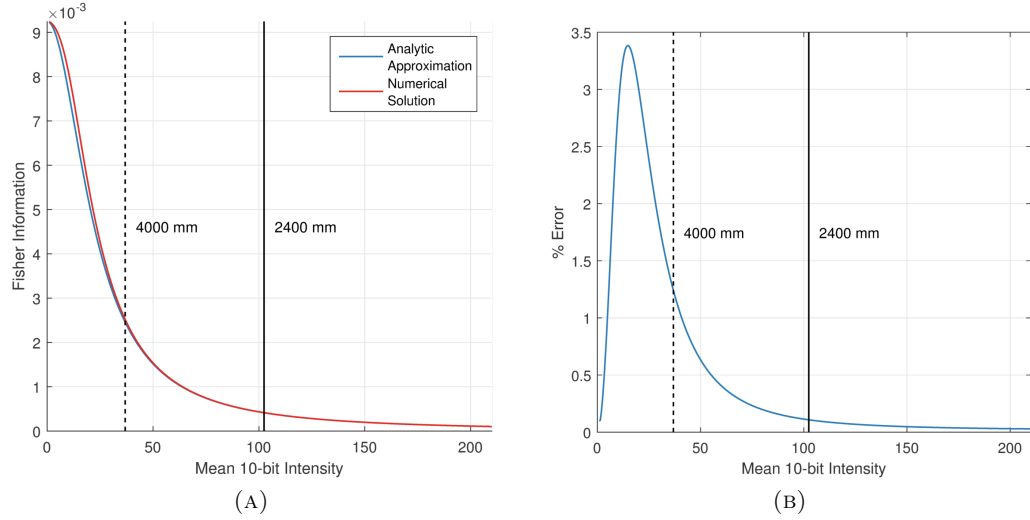


FIGURE 4.2: The analytic approximations from (4.10) and the numerical solutions for the Fisher information of Γ MGs scaled by different mean intensities (i.e. at different ranges) are shown in (A). The approximation produces a difference of less than 1% error for the square root of the CRB at typical operating ranges, as shown in (B).

family. As can be shown, the Fisher information of a NEF parameterized by the mean is given by the inverse of its variance [45]. Therefore,

$$\begin{aligned}\mathcal{I}_Z(\mu) &= \frac{1}{\frac{\mu^2(\theta)}{k}}, \text{ when } \sigma_n^2 = 0 \\ \mathcal{I}_Z(\mu) &= \frac{1}{\sigma_n^2}, \text{ when } k = \infty\end{aligned}\tag{4.9}$$

Note, since the Γ MG distribution is a convolution of two different NEFs, it is not strictly a member of the family. Alternatively, Stein *et al.* [101] prove that the Fisher information for the mean of any distribution is bounded below by the inverse of its variance

$$\mathcal{I}_Z(\mu) \geq \frac{1}{\sigma_Z^2} = \frac{1}{\frac{\mu^2(\theta)}{k} + \sigma_n^2}.\tag{4.10}$$

This bound has the potential to be tight for many unimodal distributions in which the first two central moments are available.

The effectiveness of the near equality in (4.10) is demonstrated in Figure 4.2(A), which compares the Γ MG Fisher information approximation to the respective numerical solution evaluated over a range of mean intensity values. Here, generating the red curve in Figure 4.2(A) is made possible by evaluating the convolution of the thermal noise distribution in (3.14) and the scaled speckle distribution in (3.12) at different mean intensities – e.g. the blue curves in Figures 4.3(A) and 4.3(B), which are provided for

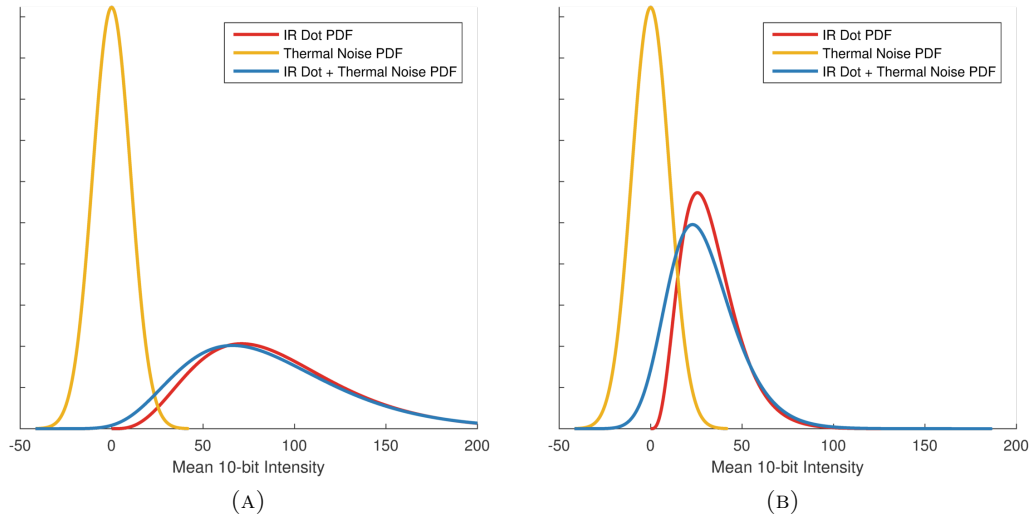


FIGURE 4.3: The differences in PDFs are shown for an IR dot with speckle noise and an IR dot with speckle and thermal noise (Γ MG) at a range of (A) 2400 mm and (B) 4000 mm.

mean intensities at ranges of 2400 mm and 4000 mm, respectively. Given the convolution at a mean μ and at a perturbed mean $\mu + \Delta\mu$, the Fisher information score is numerically computed by

$$s_Z(z; \mu) \approx \frac{\ln(f_Z(z; \mu + \Delta\mu)) - \ln(f_Z(z; \mu))}{\Delta\mu}. \quad (4.11)$$

The Fisher information of the Γ MG is then calculated as

$$\mathcal{I}_Z(\mu) \approx \Delta z \sum_k s_Z^2(z_k; \mu) f_Z(z_k; \mu), \quad (4.12)$$

which is essentially the discretization of the scalar version of (4.2). The approximation in (4.10) results in less than a 1% error for the square root of the CRB of the Γ MG at typical operational ranges, as shown in Figure 4.2(B) when 100,000 evaluations are used for the sum in (4.12).

Lastly, by substituting (4.8) and (4.10) in (4.7), the final expression for the FIM for the pose given a structured-light sensor's IR image is

$$\mathcal{I}_Z(\theta) = \sum_{p \in P} \frac{1}{\frac{\mu_p^2(\theta)}{k} + \sigma_n^2} \frac{\partial \mu_p(\theta)}{\partial \theta} \frac{\partial \mu_p(\theta)}{\partial \theta}^\top. \quad (4.13)$$

Note, the FIM expression provided above is expressed as a function of the known speckle and thermal noise parameters, k and σ_n , and the mean and sensitivity images, defined as

the set $\{\mu_p(\boldsymbol{\theta}), \frac{\partial \mu_p(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}; p = 1, 2, \dots, P\}$. It should also be noted that the approximation given by (4.13) is accurate, though conservative and biased lower because of Stein's inequality in (4.10), and because of the independence assumption applied to adjacent pixels with dot energy.

4.1.2 Sensitivity Images

Given the simulator that provides the noise-free mean image of the object's CAD, the six sensitivity images for each pose parameter can be approximated by

$$\frac{\partial \mu(\boldsymbol{\theta})}{\partial \theta_k} \approx \frac{\mu(\boldsymbol{\theta} + \Delta \theta_k \cdot \mathbf{e}_k) - \mu(\boldsymbol{\theta} - \Delta \theta_k \cdot \mathbf{e}_k)}{2\Delta \theta_k}, \quad (4.14)$$

where $k = 1, 2, \dots, 6$, and each \mathbf{e}_k is the k th Cartesian unit vector. A $\Delta \theta$ is selected that is similar to [94], which is small enough to be accurate but large enough to have a non-zero result when considering the simulator's need to model a dot as a finite collection of sub-rays.

Sensitivity images of a teapot with roughly 5000 pixels on target are shown in Figure 4.4, which demonstrate a perturbation in rotation about the (A) X, (C) Y, and (E) Z-axis with respect to the model coordinate system, and a translation in the (B) X, (D) Y, and (F) Z-axis with respect to the sensor coordinate system. When the display ranges are kept consistent, separately within both sets of images, a few interesting properties are noticeable. For instance, when the object is rotated about its X or Y-axis, there are more informative (bright white/black) pixels near the top and bottom or left and right edges of the 2D projection of the object. This is due to a greater change in intensity [from greater dot disparity movement and $\frac{1}{r^2}$ change in (4.5)] with pixels further from the axis of rotation, most notably when IR dots straddle the object/background surface boundary. Similarly, informative pixels exist along the entire boundary when the object is rotated about its Z-axis, which results in more information overall for orientation estimation.

More informative pixels also exist near the boundary when the object is translated in the X and Y-axes; however a translation in the Z-axis provides less overall information. A change in intensity of pixels with respect to X and Y-axis translation is primarily due to dot energy shifting among adjacent pixels near the object edge (i.e. where there is a step in depth), while the intensity of pixels within the cross-sectional area change moderately from a change in surface depth due to the shape of the teapot. Note, the position and intensity of pixels on a flat surface would not change with X or Y-axis translations. On the other hand, all dots on the surface of any object shift by an equivalent disparity with Z-axis translations. These large intensity changes are dampened, however, because larger translations in depth are required for equivalent object shifts in the pixel coordinate system when compared to lateral translations due to the standard stereo-triangulation equation [103]. More specifically, a pixel shift from a change in X or Y-axis translation

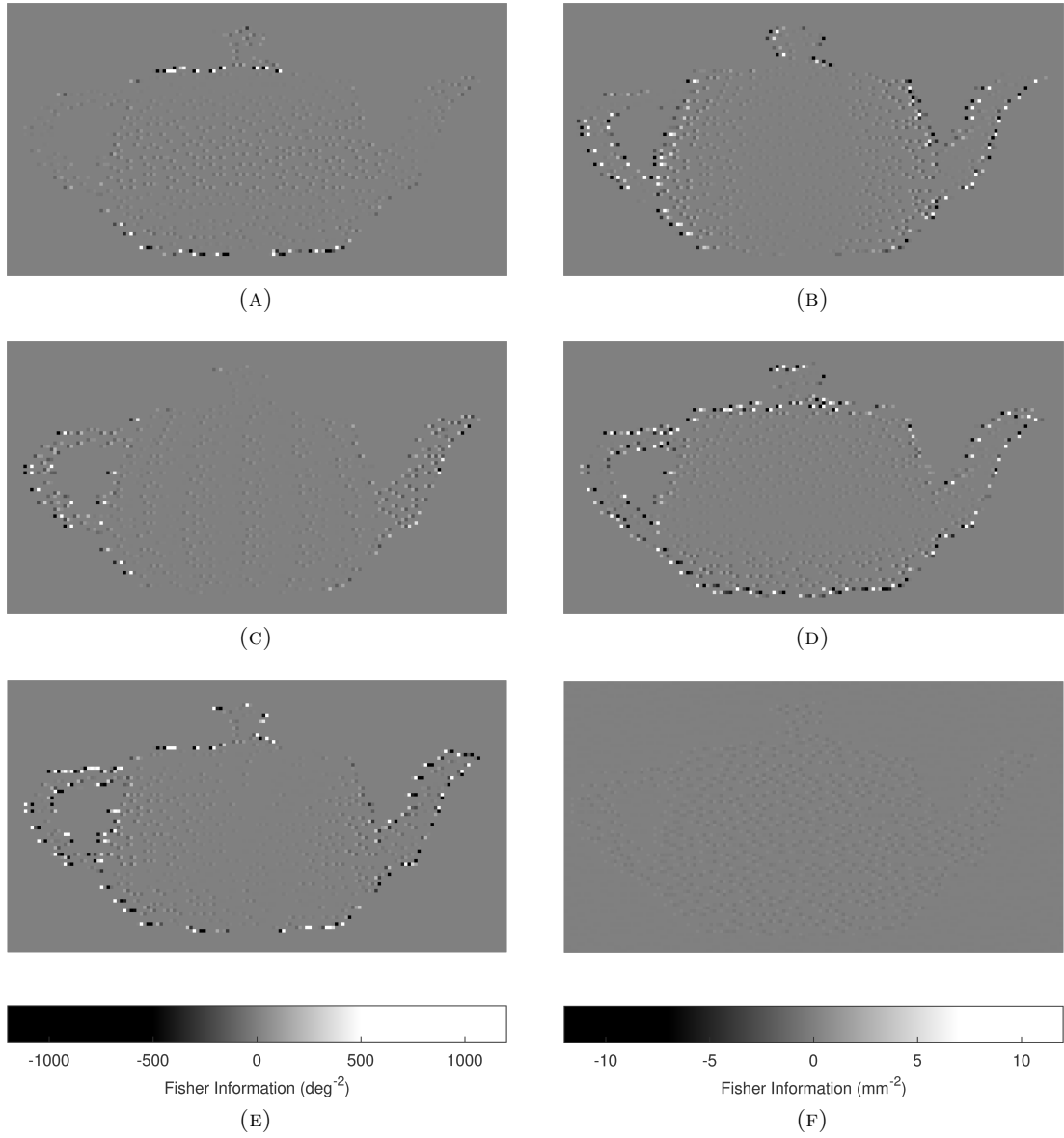


FIGURE 4.4: The sensitivity images of the teapot with roughly 5000 pixels on target demonstrate a perturbation in rotation about the (A) X, (C) Y, and (E) Z-axis with respect to the model coordinate system, and a translation in the (B) X, (D) Y, and (F) Z-axis with respect to the sensor coordinate system.

$(\Delta\theta_4$ or $\Delta\theta_5)$ that is proportional to $\frac{D}{f_x}$ is equivalent to a pixel shift from a change in Z-axis translation ($\Delta\theta_6$) that is proportional to $\frac{D^2}{bf_x}$, where b denotes the baseline distance between the transmitter and receiver. This results in larger denominators in (4.14) and less informative sensitivity images for Z-axis translations, especially for objects placed at further depths. These trends coincide with the CRBs in Section 4.1.3 and pose estimation results shown later in Section 6.3. Note, the sensitivity images for the X and Y-axis rotations and X and Y-axis translations generate a comparable amount of informative pixels, and thus result in similar CRBs.

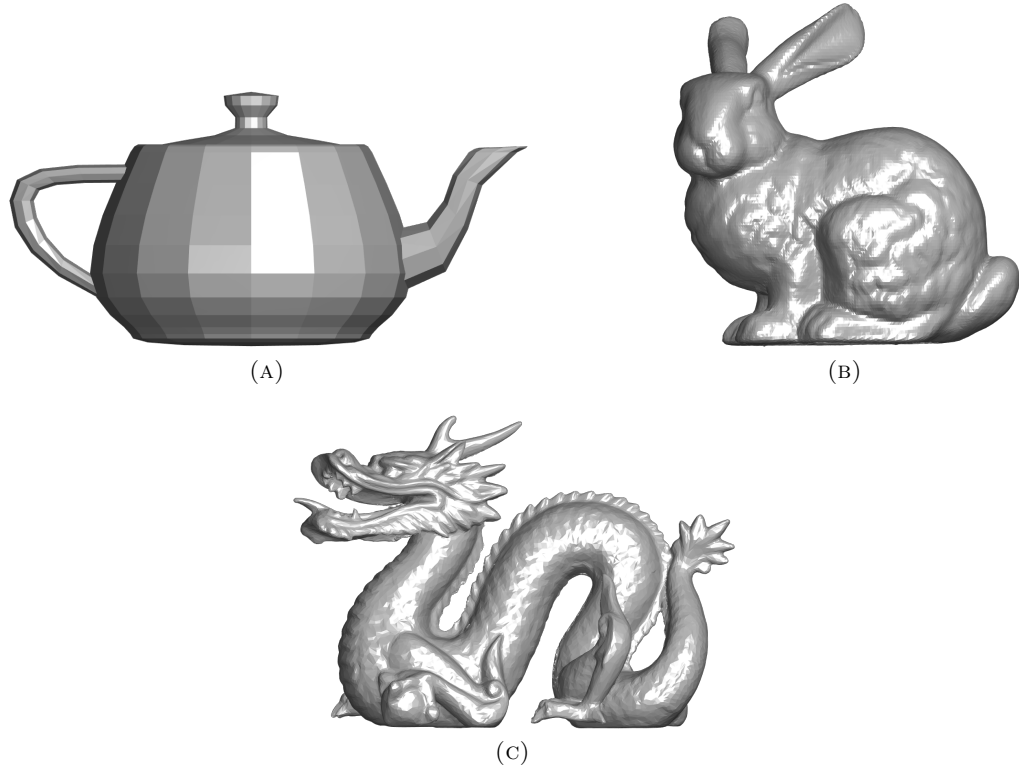


FIGURE 4.5: The three CAD models utilized in this dissertation are the (A) standard Utah teapot, (B) the Stanford bunny, and (C) the Stanford dragon, each of which contain interesting, complex, and small/thin features.

4.1.3 Cramér-Rao Bound Examples

For this section and for the results in Sections 5.4 and 6.3, the effects that shape complexity and the number of pixels on target have on the accuracy of 6D pose estimators are primarily studied. The effect that signal-to-noise ratio (SNR) and various illumination patterns have on the tightness of the error bounds are also explored in this section. As mentioned previously, the three objects utilized in this dissertation are the standard Utah teapot [Figure 4.5(A)], as well as the Stanford bunny [Figure 4.5(B)] and dragon [Figure 4.5(C)], each of which contain interesting, complex, and small/thin features.

For each of the simulated experiments, a CAD of a flat background wall was included behind the object CAD at a distance of 100 mm from the furthest facet. This is an important distinction to make because the surrounding scene, specifically the distance between the silhouette of the object and immediate background surface, significantly affects the amount of information engendered from pose parameter perturbations. As noted in Section 4.1.2, most informative pixels congregate near the object boundary where IR dots straddle the object and background surface. Thus, when the background surface is at a further depth, the changes in intensity from perturbations are greater, resulting in more Fisher information. In order to moderate this effect, a reasonably close

background surface depth was chosen as to avoid artificially exaggerated composite FIMs for the simulated images.

In order to condense the results for each set of experiments, two metrics are provided: the root CRB (RCRB) of orientation and position estimation, which are respectively computed as the square root of the trace of the 3×3 rotation component of the full 6×6 CRB, and the square root of the trace of the 3×3 translation component

$$\begin{aligned} \text{RCRB}_{\text{Orientation}} &= \sqrt{\text{Tr}(\text{CRB}_{\text{Rotation}})}, \\ \text{RCRB}_{\text{Position}} &= \sqrt{\text{Tr}(\text{CRB}_{\text{Translation}})}. \end{aligned} \quad (4.15)$$

Note, since the CRB levels out at very high resolutions and expectedly approaches infinity as the number of pixels on target approaches zero (i.e. the Fisher information goes to zero), the best-fit, two term rational function is found for each CRB metric. More specifically, a nonlinear minimization routine is applied to the data samples to fit a function where the second term has an irrational degree $-n$, i.e. $y = a_0 + a_1 x^{-n}$.

In the first set of CRB experiments [Figure 4.6(A)], the teapot was positioned at three fixed depths from the sensor, and the size was varied to generate between 50 and 5000 pixels on target. As expected, IR images generally contain more information when the SNR is greater, where SNR is defined as the ratio μ/σ_n . When the number of pixels on target is fixed [Figure 4.6(B)], the shape of the CRB curves when plotted against sensor range generally approaches a quadratic trend, especially for lower ranges, which is most likely due to the stereo triangulating system and the IR intensity's inverse square falloff rate [23]. These curves also level out at lower ranges (i.e. at higher SNRs), which is due to a canceling effect where there is an increase in speckle variance from large magnitudes of the mean in the denominator of (4.13), and magnified sensitivities in the numerator. Note, this trend is more obvious when the curves are plotted on a linear scale.

Another interesting factor to consider when predicting CRBs is the selected IR light pattern that is projected onto a scene. Though most informative pixels typically occur near the object boundary of the 2D projected image, modest increases in information can also be extracted from pixels within the cross-sectional region of the object when a more regular, fixed pattern is employed, such as the dot patterns in Figure 4.7. In these cases, pose perturbations of images with denser dot patterns allow for more pixels with no direct dot energy (black pixels) to inherit a change in intensity from neighboring pixels with dot energy (white pixels). This results in an increase in orientation estimation accuracy for both patterns [Figure 4.6(C), left], and an increase in position estimation accuracy for Pattern 1 [Figure 4.6(C), right] when compared to the pseudo-random Kinect pattern with 11% of the transmit pixels containing dot energy [88]. However, when the image is fully illuminated, the pose perturbations experience the least amount of intensity change

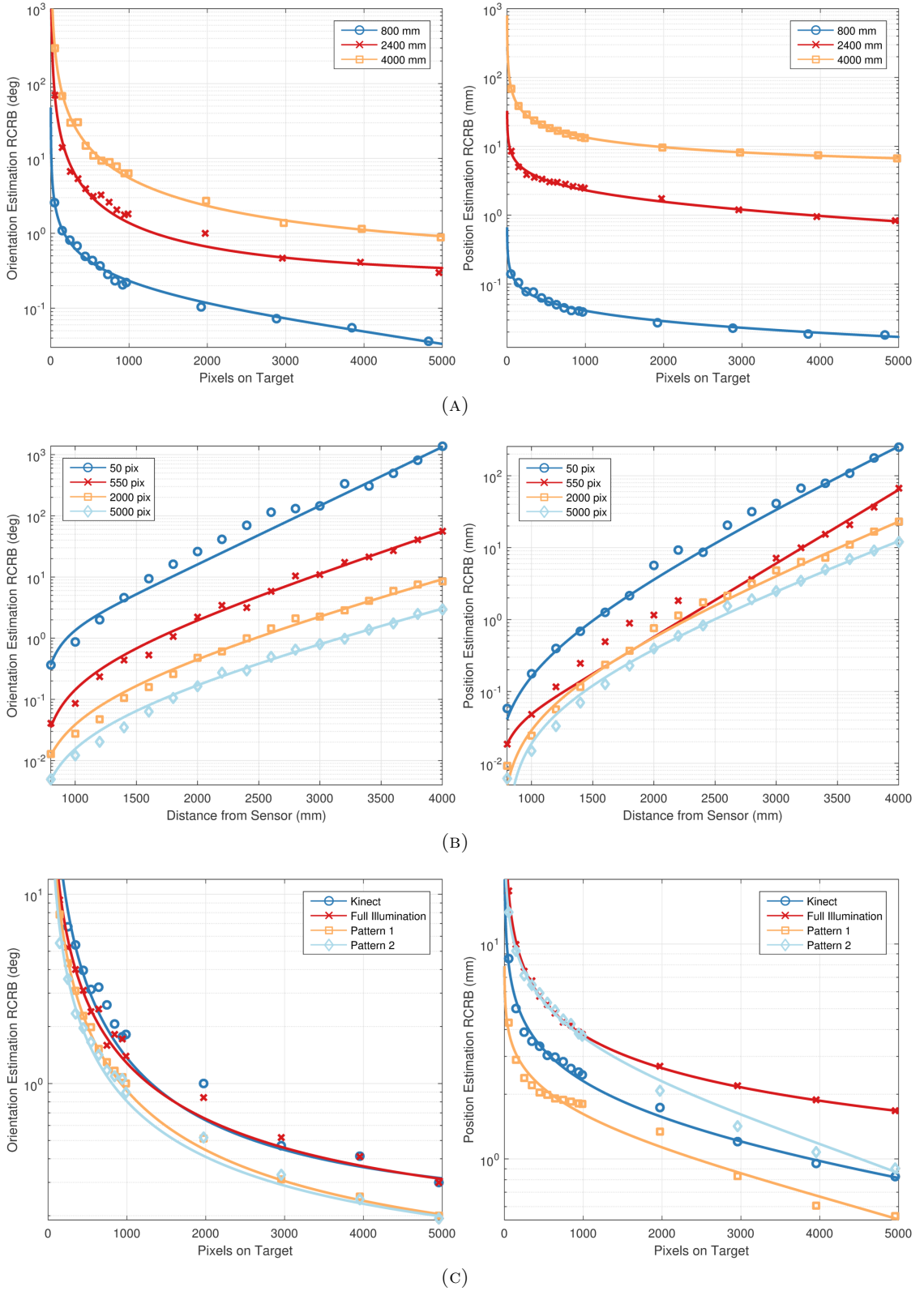


FIGURE 4.6: The log scale of theoretical RCRBs compare (A) the teapot at different distances, (B) the teapot with different amounts of pixels on target, and (C) the teapot at 2400 mm with different IR dot patterns.

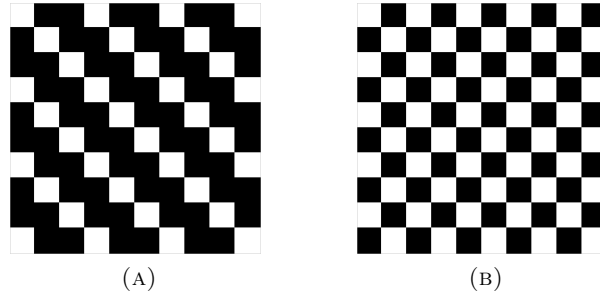


FIGURE 4.7: (A) IR Dot Pattern 1 and (B) IR Dot Pattern 2 are displayed that moderately increase the FIM of IR images, which could theoretically be implemented in structured-light sensors for more accurate object classification, shape inspection, pose estimation, etc.

overall, which produces the highest error bounds for estimation of both orientation and position. This is an important realization because though certain ToF cameras also provide access to the raw IR images without time delay information (e.g. the Kinect v2), these data sets would result in a decrease in potential pose estimation accuracy since the entire scene is illuminated by pulses of laser light. Moreover, a ToF transmitter and receiver are nearly collocated, which removes information from triangulation. On the other hand, algorithms that process IR images of a dot pattern projected from an offset laser do benefit from triangulation, allowing for higher accuracy model-based shape matching.

It is important to recognize, however, that Patterns 1 and 2, and full illumination can only be used for model-based shape matching, and are not suitable for spatial-multiplexing coding depth reconstruction. In other words, a fixed grid of dots would not provide unique local patches to match against a series of reference images, and therefore disparity estimation is not possible. On the other hand, if a CAD model of an object is provided, then a more accurate structured-light depth sensor could theoretically be constructed to perform object classification, shape inspection, pose estimation, etc.

Finally, the ability to estimate orientation and position is compared for objects of varying shape and complexity, where the ‘complexity’ of an object is defined by its composition of elaborately interconnected parts [14]. More explicitly, CAD models consist of tiled polygons (facets), usually triangles, to describe the 3D surfaces, where each triangle is defined by three vertices and a unique normal vector. Thus, a CAD with a higher complexity is discretized into a denser model point cloud composed of the center points of each facet. Based on this criteria, the teapot is least complex with 992 facets, then the bunny with 69,666 facets, and lastly the dragon with 100,000 facets. As Wang and Shen [110] postulate, when an object presents more voxels in the 2D projection of a certain viewpoint, the resulting rendered image carries more information. The order of the three objects’ complexity and its correlation to information is highlighted in the X and Y-axis rotation CRBs [Figures 4.8(A) and 4.8(C)] that dominate the estimation of

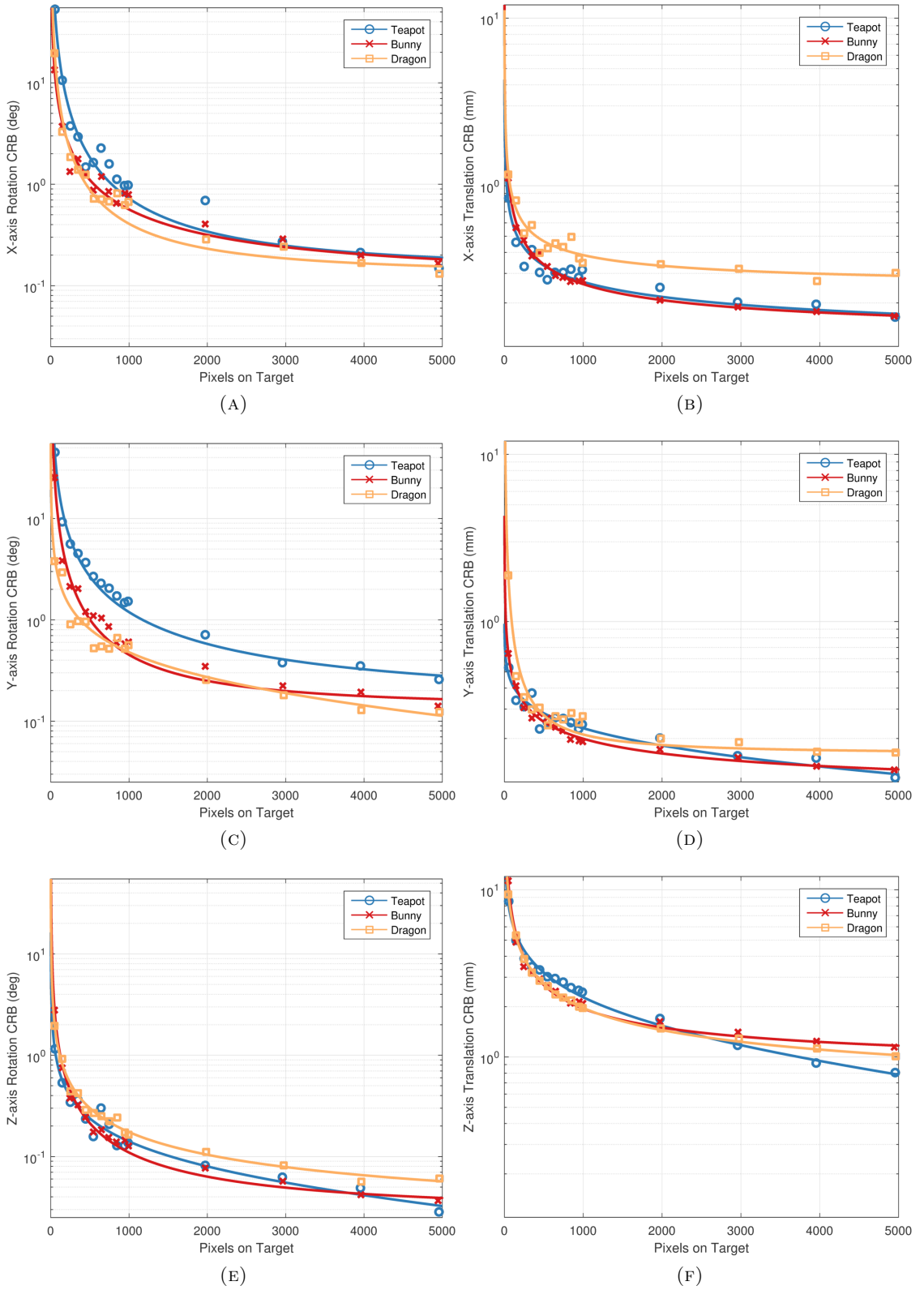


FIGURE 4.8: The log scale of theoretical CRBs compare the teapot, bunny, and dragon positioned at the center of the focal plane and at a depth of 2400 mm with varying amounts of pixels on target.

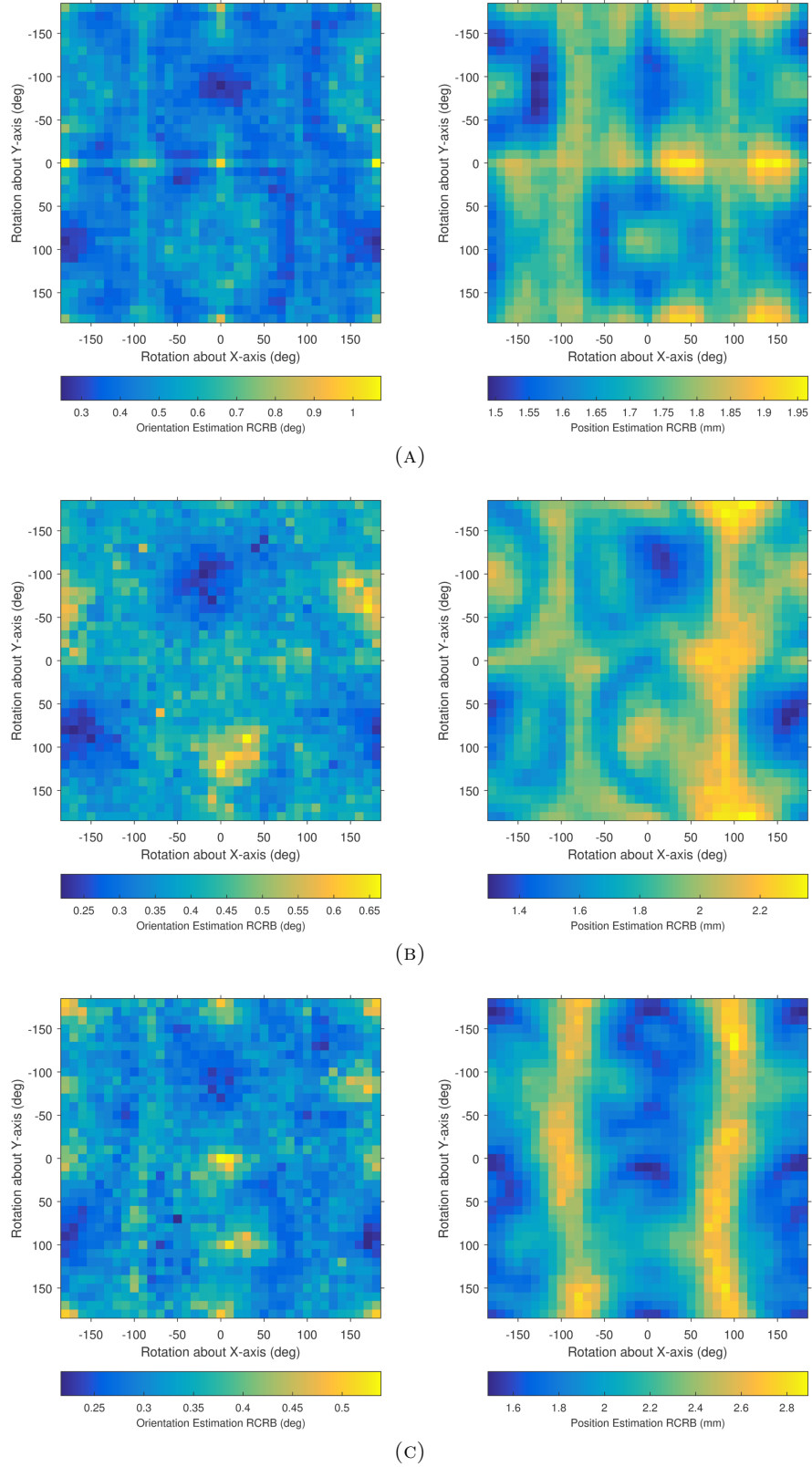


FIGURE 4.9: The heat maps of orientation and position estimation RCRBs are generated by varying the X and Y-axis rotation pose parameters for the (A) teapot, (B) bunny, and (C) dragon at a depth of 2400 mm with 2000 pixels on target.

orientation error, where the lowest CRB is estimated for the dragon when compared to the bunny and teapot with a similar number of pixels on target. This trend also exists in the Z-axis translation CRB plot [Figure 4.8(F)] for the objects with 2000 or less pixels on target. The teapot outperforms the other objects at higher resolutions, however, which is most likely due to a greater depth diversity since Z-axis translation dominates the position estimation error. It should be noted that different object poses that generate more or fewer pixels on target in the 2D projections on the receiver focal plane result in different FIMs, as depicted in Figure 4.9. Here, the heat maps of orientation and position estimation RCRBs were generated by varying pose over all X and Y-axis rotations for the three objects at 2400 mm with 2000 pixels on target, which demonstrate a modest dependency of estimation error on pose since the RCRBs fall in a fairly narrow range. Note, a rotation about the Z-axis (as well as a translation in the X or Y-axis) would not provide a change in the 2D cross-sectional shape or area of the object, and therefore results in equivalent information content and even narrower RCRB ranges.

4.2 Criteria for Informative Measurements

In general, structured-light 3D scanners are a class of range imaging devices that reconstruct the surface depth of a scene by processing images of a distorted IR pattern, where reconstruction is accomplished with a match filter that depends on the type of light pattern projected on the scene. An important property of Fisher information is that any transformation $T(\mathcal{Z})$ applied to the original data \mathcal{Z} will generally result in a loss of information [30]

$$\mathcal{I}_{T(\mathcal{Z})}(\boldsymbol{\theta}) \leq \mathcal{I}_{\mathcal{Z}}(\boldsymbol{\theta}). \quad (4.16)$$

In the case of this report, $T : \mathcal{Z} \rightarrow \mathcal{D}$ represents the processing transformation of the IR image \mathcal{Z} into the depth image \mathcal{D} , and $\mathcal{I}_{T(\mathcal{Z})}(\boldsymbol{\theta})$ denotes the Fisher information of the depth image of an object about pose $\boldsymbol{\theta}$. The equality in (4.16) holds if and only if the statistic $T(\mathcal{Z})$ is sufficient for the unknown parameter, in that the transformation contains all of the information about $\boldsymbol{\theta}$ that is available in the original data set \mathcal{Z} . More formally, the ratio of the joint PDFs defined by the IR image and depth image

$$\frac{f_{\mathcal{Z}}(\mathbf{z}; \boldsymbol{\theta})}{g(T(\mathbf{z}); \boldsymbol{\theta})} \quad (4.17)$$

must not depend on the unknown pose for the entire set of possible IR intensity values. This is a result of the Fisher-Neyman factorization theorem [65] (Theorem 6.5), in which the joint PDF of an IR image can be written as

$$f_{\mathbf{z}}(\mathbf{z}; \boldsymbol{\theta}) = h(\mathbf{z}) \cdot g(T(\mathbf{z}); \boldsymbol{\theta}), \quad (4.18)$$

where h depends only on the data and not on the unknown parameter. Note, this factorization is usually used to guide the compression of independent and identically distributed (IID) random variables – for instance, an average IR image of a sequence of frames is sufficient for the mean image of a non-moving object. Recall that the IR measurement PDF has unique mean pixel values that depend on the known transmit dot pattern, the known CAD model, and the unknown pose. Since the pixel means are unique, the pixel PDFs are not identical and lower order sufficient statistics are not achievable. In fact, the only feasible way to compress the IR data is through an appropriate pose estimator.

To demonstrate this concept, consider a simple transformation that replaces an $N \times N$ subgrid of pixels with its window average. The new downsampled IR image would clearly be less informative because of the loss in resolution. This type of information loss is demonstrated in Figure 4.6(A) when object size was varied to decrease the number of pixels on target, which is similar to a loss from lowered sensor resolution. Continuing the example, consider sliding the averaging window instead of downsampling so that the image size is maintained. Deconvolution of the new IR image is then possible, where the original IR image is theoretically retrievable. However, due to the disparity estimation error sources summarized in Figure 2.4, the depth image is not invertible. Thus, any estimator of the mean IR image from the depth data would result in IR intensity errors that are correlated, inhomogeneous, and significantly larger than the noise in the raw measurements. This means that the depth data cannot be used to provide a statistically equivalent IR image, which is a necessary condition for writing (4.18). It is important to note, on the other hand, that the original data set is trivially sufficient since it doesn't lead to any data reduction.

In addition to sufficiency, the concepts of identifiability and completeness are important to this study. A family is identifiable when

$$f_{\mathbf{z}}(\mathbf{z}; \boldsymbol{\theta}_1) = f_{\mathbf{z}}(\mathbf{z}; \boldsymbol{\theta}_2), \text{ iff } \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2 \quad (4.19)$$

for every pose in the set of all poses, $\forall \boldsymbol{\theta} \in \boldsymbol{\Theta}$. This holds true if the object is not symmetric about any of its axes in the 3D rotation group $SO(3)$, or if a chosen subset $\boldsymbol{\Theta}' \subseteq \boldsymbol{\Theta}$ is exclusive of any symmetry. In other words, the problem is identifiable when every mean image is unique for all $\boldsymbol{\theta}$ in the chosen domain. Additionally, the problem is not identifiable if the sensitivity gradient isn't full rank, i.e. when the rank is less than $\dim(\boldsymbol{\theta}) = 6$. Otherwise, there exists a hyperplane in $\boldsymbol{\theta}$ in which the mean image and joint PDF is constant.

It is well known that an exponential family of a minimally sufficient (non-compressible) statistic is complete, provided that it is full rank [65] (Theorem 6.22). Thus, in either limit when $\sigma_n^2 = 0$ or $k = \infty$, the resulting exponential families are complete if the vector of sensitivity images is full rank, i.e. when the FIM is nonsingular and invertible. This logic can then be extended to the FMG family for IR images that also depend on the pose parameter through the mean image. Thus, the information provided by \mathcal{Z} can be considered complete for the pose if a unique estimate of θ is possible, and there are at least $\dim(\theta) = 6$ pixels on target with dot energy (principally at high SNR). The implications of sufficiency and completeness are important for establishing optimality, and ultimately motivate the exploitation of raw IR images in the pursuit of an optimal pose estimator.

4.3 Criteria for an Optimal Estimator

An optimal estimator is defined as the best possible estimator for an unknown parameter given a data set from a family of distributions, where the term ‘best’ depends on the choice of a loss function. For this dissertation, the best estimator is defined as minimizing the mean square error (MSE) for all possible object poses in the 6D domain of the structured-light image data. Thus, the quadratic loss function that measures the quality of any estimator $\hat{\theta}(\mathcal{Z})$ is

$$\text{MSE}_{\hat{\theta}(\mathcal{Z})} = \text{E}[(\hat{\theta}(\mathcal{Z}) - \theta)(\hat{\theta}(\mathcal{Z}) - \theta)^\top; \theta]. \quad (4.20)$$

Since a full rank IR image is assumed sufficient and complete, it could potentially be used to devise the uniformly minimum variance unbiased estimator (UMVUE) according to the Lehmann-Scheffé theorem [65] (Theorem 1.11). If an unbiased estimator existed for the structured-light data and satisfied the CRB established in (4.1) with equality for all $\theta \in \Theta$, it would, by definition, be considered the *efficient* estimator. This is the sought-after gold standard since no other unbiased estimator can perform better in terms of the MSE.

In practical terms for any real-world optimization problem, however, a truly unbiased estimator may never exist due to a number of natural stochastic, unaccountable error sources. The best design should therefore minimize the consequences of mismodel while maintaining feasible computational expense via appropriate simplifications and assumptions. For instance, in this study the responsivity and size of each pixel are assumed uniform, and the speckle shape parameter and thermal noise variance are assumed sufficiently calibrated. The uniformly minimum MSE (UMMSE) estimator must then minimize the error metric

$$\text{MSE}_{\hat{\boldsymbol{\theta}}(\mathcal{Z})} = \text{Cov}_{\mathcal{Z}}(\hat{\boldsymbol{\theta}}(\mathcal{Z})) + \mathbf{b}(\boldsymbol{\theta})\mathbf{b}(\boldsymbol{\theta})^\top \quad (4.21)$$

for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, where the bias of an estimator is defined as

$$\mathbf{b}(\boldsymbol{\theta}) = \text{E}[\hat{\boldsymbol{\theta}}(\mathcal{Z})] - \boldsymbol{\theta}. \quad (4.22)$$

Any MSE estimator is then lower bounded by the generalized CRB

$$\left(1 + \mathbf{b}(\boldsymbol{\theta})'\right) \mathcal{I}_{\mathcal{Z}}^{-1}(\boldsymbol{\theta}) \left(1 + \mathbf{b}(\boldsymbol{\theta})'\right)^\top + \mathbf{b}(\boldsymbol{\theta})\mathbf{b}(\boldsymbol{\theta})^\top, \quad (4.23)$$

which is further lower bounded by the covariance $\text{Cov}_{\mathcal{Z}}(\hat{\boldsymbol{\theta}}(\mathcal{Z}))$ when $\frac{\partial}{\partial \boldsymbol{\theta}} \text{E}[\hat{\boldsymbol{\theta}}(\mathcal{Z})] > 1$. Since it is generally difficult to know how an estimator's bias varies with the unknown parameter, it is useful to compare the MSEs of different methods to each other and the calculated CRB given by (4.1).

Several methods have been developed that estimate the unknown parameters of a distribution, such as the method of moments. However, the maximum likelihood estimation (MLE) method – originally developed by Ronald Fisher – regularly outperforms the respective counterparts, and have a number of properties that make it the most widely used method in statistics. For instance, in the limit when the sample size increases to infinity, MLE methods are consistent, asymptotically normal, and asymptotically efficient [53]. More explicitly, consistency implies that both the bias and variance asymptotically approach zero in the limit as the number of measurements, P , increases to infinity, and asymptotic normality implies that the shape of the limiting error distribution approaches a Gaussian distribution, $\mathcal{N}(0, \mathcal{I}_{\mathcal{Z}}^{-1}(\boldsymbol{\theta})/P)$.

5

POINT SET REGISTRATION

To me programming is more than an important practical art. It is also a gigantic undertaking in the foundations of knowledge.

– David Sayre (1962)

This chapter is structured as follows: Section 5.1 gives an overview of current PSR methods including improvements made to the original ICP method. The PSR data model and the proposed PSR-MLE method are then described in detail in Section 5.2 and Section 5.3. Finally, in Section 5.4, the performance of PSR-MLE is evaluated and compared to the performance of the ICP cost function, an ICP variant, and a cutting edge, soft point-pair assigning PSR method.

5.1 Background

An early adaptation of point set registration (PSR) is known as the iterative closest point (ICP) algorithm, which was first introduced by Besl and McKay [13] in 1992, and is generally considered the most popular approach for the alignment of range data to 3D shapes represented by point clouds. Typically, ICP-like algorithms iterate two steps, where a correspondence between each measurement point and its closest model point is first established based on the smallest (Euclidean) distances, and then a transformation is performed in closed form or via numerically stable conditions such as singular value decomposition. This simplified ICP model can perform poorly on degenerate measurements however, especially when a good initial estimate of the pose transformation isn't provided, the noise isn't isotropic, an exact correspondence between measured and model points doesn't exist, or when single sensor measurements result in sparse and incomplete measurement point clouds. Several variants of the ICP method have therefore been introduced in order to address these issues, primarily the first three.

Rusinkiewicz and Levoy [89] provide an extensive examination of the ICP algorithm, including the effect of the selection of point pair assignment and weighting, outlier rejection, and error metrics. The focus of their paper is more on improvements to convergence speed while maintaining comparable performance, and less on stability and robustness. Gelfand *et al.* [39] address the issue of automatic alignment without an assumption of the initial estimate of a transformation by the use of global registration. Similarly, Brown and Rusinkiewicz [19] present an algorithm for global non-rigid alignment of 3D point clouds, which is robust for noise and non-linear warp. Maier-Hein *et al.* [70] present the anisotropic ICP (A-ICP) method that accounts for anisotropic Gaussian noise by modifying a distance metric via a non-identity and varying covariance matrix, which is shown to be more robust for partially overlapping surfaces. The issue of misalignment due to measurement error and uncertainty in initial guess is also considered by Hara *et al.* [47], where they introduce a prior error distribution for the pose, and maximize the a posteriori likelihood.

The common mechanism for the aforementioned ICP variants requires an exact correspondence between the measured point cloud and the discretized CAD model point cloud. This happens because multiple points on a CAD model can contribute to a single pseudo-measurement. Also, in reality objects are continuously extended and sensors form measurements via pixelation, and therefore an exact correspondence does not theoretically exist. Liu and Chen [66] introduce a shape descriptor for 2D point sets by histogramming points into polar bins, attempting to relieve the need for a strict correspondence. This has proven to make the algorithm more robust for distortions, but the size of the bins are user defined, i.e. not analytically determined. The kernel correlation (KC) method proposed by Tsin and Kanade [107] is an earlier example of a method that calculates correlation weights that provide information on how close model points are to each pseudo-measurement with a Gaussian kernel. Also, the coherent point drift (CPD) algorithm proposed by Myronenko and Song [78] jointly estimates the pose and variance of an assumed isotropic noise distribution in an expectation-maximization (EM) framework. Both of these methods are shown to be more robust for noise and outliers, though they implicitly assume isotropic errors. Moreover, all point set registration methods assume homogeneity, which is invalid since scene-specific errors from each depth processing step are absorbed into the pseudo-measurement point clouds. These limits motivate the method presented in this chapter, which determines the probability distribution of each measurement generated from the collected surface of transformed model points based on a Bayesian approach, and begins to address the inhomogeneous error sources by predicting scene and pose dependent facet visibility.

5.2 Data Model

In the setup of PSR methods, the model and pseudo-measurement point clouds are respectively denoted as $\mathbf{M} \in \mathcal{M}$ and $\mathbf{S} \in \mathcal{S}$, similar to the measurement model in Section 3.2. However in this chapter, the surface model can include the CAD model of the rigid object provided for pose estimation, a model of the background surface, or any other object known to be present within the FOV. A point on the surface of the model \mathbf{M} then represents the coordinates of the centers of each facet with respect to the sensor. Also, each pseudo-measurement is modeled as $\mathbf{S} = \boldsymbol{\theta} \cdot \mathbf{M} + \mathcal{N}$, where pose $\boldsymbol{\theta}$ represents the translation and rotation that transforms the model point \mathbf{M} that ideally corresponds to \mathbf{S} . Note, the transformed surface point $\boldsymbol{\theta} \cdot \mathbf{M}$ is corrupted by a unique measurement error \mathcal{N} as before, which generally depends on the location of the pixel coordinate and depth of the object the pixel measured.

The conditional probability density function $f_{\mathbf{S}|\boldsymbol{\theta},\mathbf{M}}(s|\boldsymbol{\theta},m)$ that describes the likelihood that a measurement point \mathbf{S} was generated from a transformed model point \mathbf{M} can be described by a function depending on the distribution of \mathcal{N} . In this chapter, it is again assumed that each pseudo-measurement can be represented as a unique multivariate Gaussian distribution. Therefore $\mathbf{S} \sim \mathcal{N}(\boldsymbol{\theta} \cdot \mathbf{M}, \Sigma_{\mathbf{S}})$, where $\Sigma_{\mathbf{S}}$ is a 3×3 diagonal covariance matrix when measurement error is assumed as independent in each dimension. The conditional probability density function that describes the likelihood that a measurement point \mathbf{S} was generated from a transformed model point \mathbf{M} is then

$$f_{\mathbf{S}|\boldsymbol{\theta},\mathbf{M}}(s|\boldsymbol{\theta},m) = \frac{1}{(2\pi)^{\frac{3}{2}} |\Sigma_{\mathbf{S}}|^{\frac{1}{2}}} e^{-\frac{1}{2}(s-\boldsymbol{\theta} \cdot m)^{\top} \Sigma_{\mathbf{S}}^{-1} (s-\boldsymbol{\theta} \cdot m)}, \quad (5.1)$$

where larger incremental likelihoods are inherited from closer correspondences.

5.3 Pose Hypothesis Evaluation

In this section, three cost functions are studied: the ICP [13] and A-ICP [70] methods, which make a *hard assignment* between the point clouds, and the proposed method, referred to as the PSR maximum likelihood estimation (PSR-MLE) algorithm, which *soft assigns* all measurement points to all model points. A fast method for determining the visibility of a model point by utilizing the normals to the facets is also presented, along with two other methods to determine point priors. Lastly, a Bayesian method that inherently handles outlier rejection of measurement points is presented, which is also integrated in the PSR-MLE cost function.

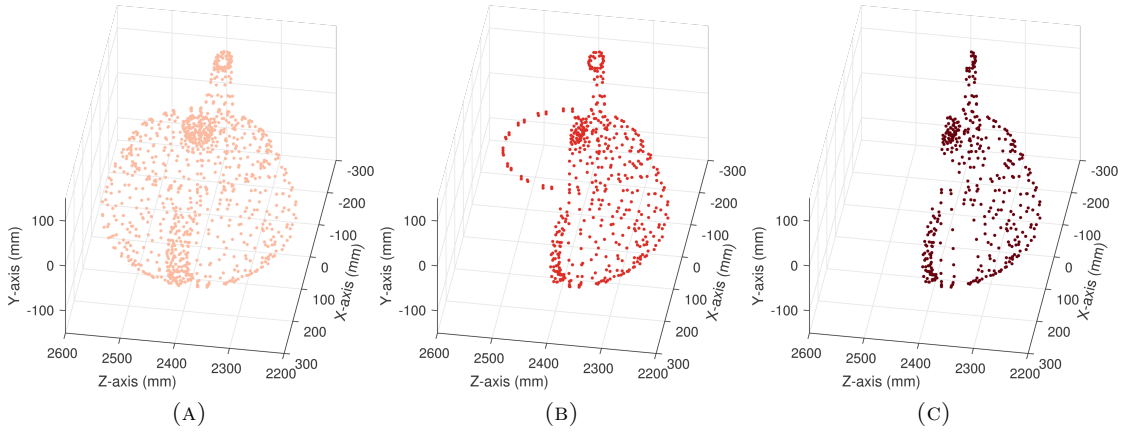


FIGURE 5.1: Model point clouds plotted in the sensor coordinate system of (A) unfiltered center points, (B) filtered center points using the direction of each normal vector with respect to the sensor direction, and (C) the set of filtered center points using ray casting are displayed.

5.3.1 Point Prior

Prior 1: Unfiltered Model Point Cloud

The prior, or selection distribution, of the model points is the probability density $f_{\mathbf{M}|\boldsymbol{\theta},\mathcal{M}}(m|\boldsymbol{\theta},\mathcal{M})$, conditioned on a given pose $\boldsymbol{\theta}$. In general, the density varies with $\boldsymbol{\theta}$ because geometrical factors such as obscuration and perspective projection affect the sensor's ability to observe each 3D model point. When facet occlusion is ignored, the prior selection probability for the facets is assumed to be distributed based on facet size, regardless of pose, which tacitly assumes that the object is transparent [Figure 5.1(A)]. Therefore,

$$f_{\mathbf{M}|\boldsymbol{\theta},\mathcal{M}}(m|\boldsymbol{\theta},\mathcal{M}) = \frac{A_m}{\sum_{m_j \in \mathcal{M}} A_{m_j}}, \quad (5.2)$$

where A_m is the area of the facet that defines model point m . This assumption is appropriate for a model of the background surface, for example, especially if the surface is flat and each point in the model is observed by the sensor.

Prior 2: Filtered Model Points from Facet Normals

The points on a solid CAD model, however, may be observable to, or occluded from the sensor, depending on pose $\boldsymbol{\theta}$. The prior probability for each point \mathbf{M} may then take on a binary value, depending on the direction of the respective normal vector \mathbf{N}_m in relation to vector \vec{m} that points from the sensor's origin to \mathbf{M} . Thus, the indicator function for the model point prior is defined by the dot product of both vectors

$$\mathcal{J}(m, \boldsymbol{\theta}) = \begin{cases} 1 & \text{if } \mathbf{N}_m \bullet \vec{m} < 0 \\ 0 & \text{if } \mathbf{N}_m \bullet \vec{m} \geq 0 \end{cases}. \quad (5.3)$$

The model point probability density then becomes

$$f_{\mathbf{M}|\boldsymbol{\theta}, \mathcal{M}}(m|\boldsymbol{\theta}, \mathcal{M}) = \frac{\mathcal{J}(m, \boldsymbol{\theta}) A_m}{\sum_{m_j \in \mathcal{M}} \mathcal{J}(m_j, \boldsymbol{\theta}) A_{m_j}}. \quad (5.4)$$

This assumption holds true if the shape of the CAD model is convex, i.e. a straight line segment that connects every pair of points within the object is also within the object. However, the object used for this paper is not fully convex, and some points that are truly self-obstructed produce a negative dot product under certain poses, as shown in Figure 5.1(B).

Prior 3: Filtered Model Points from Ray Casting

Self-occlusions may also be determined by a ray casting method to produce a fully filtered model point cloud [Figure 5.1(C)]. Here, rays are constructed that emanate from the origin of the sensor coordinate system and end at the centers of each facet on the object CAD given a pose hypothesis. If a ray intersects the object surface at a closer distance before reaching the initialized 3D center point, the facet is considered as self-occluded and filtered from the model point cloud set for that iteration of cost function computation. Note, these methods can be very computationally expensive; however, in order to allow for faster processing times, the Matlab wrapper for the optimized algorithm developed by Terdiman [104] (utilized in Section 3.4.1 to simulate the transmitted and received IR dot pattern) is implemented later in Section 5.4 in conjunction with the PSR-MLE algorithm.

5.3.2 Point-Pair Assignment

Point-Pair 1: ICP

As mentioned in Section 5.1, ICP begins each iteration of its cost function reduction by establishing a correspondence between the point clouds via hard assigning each pseudo-measurement point to the model point in \mathcal{M} with the smallest Euclidean distance. Hence, ICP maximizes the conditional probability of a measurement \mathbf{S} that was generated from the closest surface model point \mathbf{M} transformed by a hypothetical pose $\boldsymbol{\theta}$

$$f_{\mathbf{S}|\boldsymbol{\theta}, \mathcal{M}}(s|\boldsymbol{\theta}, \mathcal{M}) = \max_{m \in \mathcal{M}} f_{\mathbf{S}|\boldsymbol{\theta}, \mathbf{M}}(s|\boldsymbol{\theta}, m), \quad (5.5)$$

where the point prior distribution $f_{\mathbf{M}|\boldsymbol{\theta},\mathcal{M}}(m|\boldsymbol{\theta},\mathcal{M})$ is assumed uniform, and the input measurement set is subject to an assumed isotropic zero-mean Gaussian noise distribution, i.e. $\Sigma_{\mathbf{S}}$ is chosen as a 3×3 identity matrix. Note, this is equivalent to minimizing $(s - \boldsymbol{\theta} \cdot m)^\top (s - \boldsymbol{\theta} \cdot m)$ from (5.1).

Point-Pair 2: A-ICP

The A-ICP method alternatively makes use of an established independent, anisotropic measurement error distribution, and hard assigns each measurement \mathbf{S} to the closest model point in \mathcal{M} with the smallest normalized (Mahalanobis) distance prescribed by the exponent in (5.1). Hence, A-ICP maximizes (5.5) when $\Sigma_{\mathbf{S}}$ is chosen as a diagonal matrix with elements $\text{diag}(\sigma_x^2, \sigma_y^2, \sigma_z^2)$, which is equivalent to minimizing $(s - \boldsymbol{\theta} \cdot m)^\top \Sigma_{\mathbf{S}}^{-1} (s - \boldsymbol{\theta} \cdot m)$ from (5.1).

In the traditional ICP and A-ICP architectures, after point-pair correspondences are established via a hard assignment, a rigid transformation is estimated by minimizing the normalized distance squared between corresponding points. Once the transformation is applied to the model point set, the next iteration is initiated by again finding a correspondence as a new incremental likelihood for the unknown pose, which is repeated until a stopping criteria is met. It is important to note that these methods typically don't converge exactly to the local optima, which is due to a changing cost function of registration after the transformation is applied at the end of each iteration.

Point-Pair 3: PSR-MLE

Given the conditional densities of the selection distribution and the likelihood a measurement point \mathbf{S} was generated from a transformed model point \mathbf{M} , the probability distribution of \mathbf{S} generated from the collected surface of transformed model points \mathcal{M} is determined based on a Bayesian approach [123]

$$f_{\mathbf{S}|\boldsymbol{\theta},\mathcal{M}}(s|\boldsymbol{\theta},\mathcal{M}) = \int_{m \in \mathcal{M}} f_{\mathbf{S}|\boldsymbol{\theta},\mathbf{M}}(s|\boldsymbol{\theta},m) f_{\mathbf{M}|\boldsymbol{\theta},\mathcal{M}}(m|\boldsymbol{\theta},\mathcal{M}) dm. \quad (5.6)$$

Since the surface of a solid CAD model is quantized into a countable list of model points, the measurement probability density is estimated as

$$\begin{aligned} f_{\mathbf{S}|\boldsymbol{\theta},\mathcal{M}}(s|\boldsymbol{\theta},\mathcal{M}) &\approx \sum_{m_i \in \mathcal{M}} f_{\mathbf{S}|\boldsymbol{\theta},\mathbf{M}}(s|\boldsymbol{\theta},m_i) f_{\mathbf{M}|\boldsymbol{\theta},\mathcal{M}}(m_i|\boldsymbol{\theta},\mathcal{M}) \\ &= \sum_{m_i \in \mathcal{M}} f_{\mathbf{S}|\boldsymbol{\theta},\mathbf{M}}(s|\boldsymbol{\theta},m_i) \frac{\mathcal{J}(m_i, \boldsymbol{\theta}) A_{m_i}}{\sum_{m_j \in \mathcal{M}} \mathcal{J}(m_j, \boldsymbol{\theta}) A_{m_j}}. \end{aligned} \quad (5.7)$$

Thus, weights depend on how close model points are to each pseudo-measurement, where higher weights are given to model points that are closer to a pseudo-measurement, which

has a similarity to image pixelation where many points on the surface of an object contribute to a single measurement.

If the given CAD model is sparse, and model points are far apart, then a hard assignment may perform nearly as well as the proposed PSR-MLE method. This is due to the PSR-MLE mechanism where very small weights are computed when model points are very far away. Theoretically, though, the A-ICP method is not as statistically accurate as the PSR-MLE method because in reality a pixel measurement depends on all model points in its neighborhood, monotonically from the center of the pixel in the depth image. In other words, the sensing mechanism is not a one-to-one/one-to-many association, it's a many-to-many association.

5.3.3 Decision Rule and Clutter Model

In any realistic observed scene, measurements in the respective depth image are generated not only by the target of interest, but from other sources such as the background surface or clutter. Clutter may include physical objects within the FOV that may or may not obstruct the target of interest, or spurious noise caused by camera jitter, material reflectance, etc. Since the object can be positioned anywhere within the FOV, each measurement $\mathbf{S} \in \mathcal{S}$ is determined as being generated from the object, or any other model within the scene. A minimum probability of error hypothesis test to decide whether an observed point was generated from the CAD model of interest under a certain pose $f_{\mathbf{S}|\boldsymbol{\theta}, \mathcal{M}}(s|\boldsymbol{\theta}, \mathcal{M}_{CAD})$, or another model such as clutter $f_{\mathbf{S}|\mathcal{M}}(s|\mathcal{M}_{clutter})$, can be expressed as a maximum over these conditional densities [28]:

$$l(s, \boldsymbol{\theta}) = \max \left\{ f_{\mathbf{S}|\boldsymbol{\theta}, \mathcal{M}}(s|\boldsymbol{\theta}, \mathcal{M}_{CAD}), \max f_{\mathbf{S}|\mathcal{M}}(s|\mathcal{M}_{other}) \right\}, \quad (5.8)$$

where

$$\max f_{\mathbf{S}|\mathcal{M}}(s|\mathcal{M}_{other}) = \max \left\{ f_{\mathbf{S}|\mathcal{M}}(s|\mathcal{M}_{clutter}), \dots \right\} \quad (5.9)$$

is preprocessed prior to pose estimation.

Instances do exist where multiple objects contribute to a single pixel, such as when the edge of an object straddles a portion of the pixel. A full Bayesian approach accommodating multiple object sets would require a weighted combination of points from all objects within view of each pixel, where ray casting can be used to determine if one object is fully occluding another. However, since only a single object contributes to the depth value of most pixels, the proposed decision model is nearly Bayesian, which alleviates the need for ray casting.

Since clutter may exist anywhere within the volume of the FOV of the sensor V_{FOV} , the probability density is modeled as a uniform distribution

$$f_{\mathbf{S}|M}(s|M_{clutter}) = \beta_{clutter} = \frac{E[N_{clutter}]}{V_{FOV}}, \quad (5.10)$$

where $\beta_{clutter}$ is the clutter density [61] and $E[N_{clutter}]$ is the expected value of the number of clutter points. The model for clutter can be adjusted depending on how confident the user is of making a correct measurement point classification. With this mechanism, measurements far away from *all* object model points are then rejected as outliers.

Finally, the composite likelihood for all measurements given the pose of the object is given by

$$\tilde{l}(\mathcal{S}, \boldsymbol{\theta}) = \prod_{s_k \in \mathcal{S}} l(s_k, \boldsymbol{\theta}), \quad (5.11)$$

which is applied to both the hard and soft assignment methods. When the prior distribution for pose $f_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ is uniform, the pose estimate $\hat{\boldsymbol{\theta}}$ is chosen as the value that maximizes (5.11), since it is proportional to the posterior of $f_{\boldsymbol{\theta}|\mathcal{S}}(\boldsymbol{\theta}|s)$. Note, the width of the peak of the 6D distribution $\tilde{l}(\mathcal{S}, \boldsymbol{\theta})$ in the neighborhood of $\hat{\boldsymbol{\theta}}$ is related to the uncertainty of the estimate, and for locally Gaussian shapes, can be described by a covariance $\Sigma_{\boldsymbol{\theta}}$.

It is also important to note that maintaining the same number of measurement points when evaluating pose hypotheses at each iteration of cost function computation is necessary. Moreover, the posterior becomes flatter from (5.8) and (5.9) when the environment is deemed to have more clutter. In the limit when clutter density is very large, every measurement is assigned the constant $f_{\mathbf{S}|M}(s|M_{clutter})$, where $\tilde{l}(\mathcal{S}, \boldsymbol{\theta})$ would be equal to a constant for all $\boldsymbol{\theta}$. It is therefore important to select an appropriate clutter model when the object is not well segmented within the depth image.

5.4 Results

For the simulated data sets in this section, the CAD model of the standard Utah teapot oriented as shown in [Figure 4.5(A)] was selected for its complex shape and symmetry, and because there is no ambiguity in pose over the entire set of Euclidean rotations. A commonly explored environment where measurement point clouds are simulated directly (i.e. when depth image formulation is bypassed) was first utilized to test the efficacy of the A-ICP cost function, as well as the proposed PSR-MLE cost function in conjunction with the fast facet visibility calculation presented in (5.4). Under these ideal conditions, a true correspondence exists between either a subset of the measurement or model point

clouds. In each experiment, 100 sparse or overloaded data sets were generated from the original CAD model center points, where independent X, Y, and Z-axis anisotropic Gaussian noise was introduced. The random error values introduced to each simulated data set correspond to the error model statistics provided by Section 3.2, which vary by depth and radial distance from the origin of the sensor.

In order to compare PSR-MLE to A-ICP, two measures of accuracy are utilized: *relative rotation error* (RRE) defined by [50], and *total translation error* (TTE)

$$\text{RRE} = \min \left\{ \|\hat{\theta}_{\bar{q}} - \theta_{\bar{q}}\|, \|\hat{\theta}_{\bar{q}} + \theta_{\bar{q}}\| \right\}, \quad (5.12)$$

$$\text{TTE} = \|\hat{\theta}_{x,y,z} - \theta_{x,y,z}\|, \quad (5.13)$$

where \bar{q} is the unit quaternion representing orientation, and the set x, y, z is the position coordinates of the object. The relative rotation error is calculated as the norm of the difference between the estimated and ground truth parameterizations of rotations. This dimensionless metric ranges between $[0, \sqrt{2}]$, and approaches a perfect score of 0 at a much slower rate than other metrics. In order to find the global maximum likelihood for the methods on the continuous 6D pose domain, the genetic algorithm, a built-in Matlab global optimization routine, is used.

Figure 5.2 presents results where the percentage of CAD points used to form measurements are varied. Performance as a function of range-dependent error is depicted where optimum alignment against the ‘full’ CAD point cloud defines the pose estimate for each of the 100 trials. These experiments that vary measurement quantity could represent different sensor resolutions or missing data due to jitter, etc. When the measurement quantity is greater than 100%, measurements are generated by resampling CAD points randomly as needed to resemble the redundancy that can occur in a step-stare sensor. An interesting trend occurs with A-ICP position error [Figure 5.2(C)] when the measurement noise reaches an axial standard deviation of 51 mm at a theoretical distance of 5000 mm from the Kinect sensor. This is referred to as the peaking phenomenon, as reported in [124] for shape recognition and object classification, where more measurements actually correlate with a decrease in performance. The peaking phenomenon occurs partly due to the ambiguity provoked when multiple measurement points are hard assigned to the same model point. The PSR-MLE algorithm, however, achieves a monotonic increase in performance with an increase in measurement quantity [Figure 5.2(D)]. This is also true for estimation of orientation, where Figure 5.2(B) demonstrates PSR-MLE outperforming A-ICP, which again achieves a monotonic increase in performance. Thus, the PSR-MLE method accurately determines the orientation and position of the object without the need of an exact correspondence to the given CAD model points, which is related to recognizing the shape of the object within the noisy measurement point cloud.

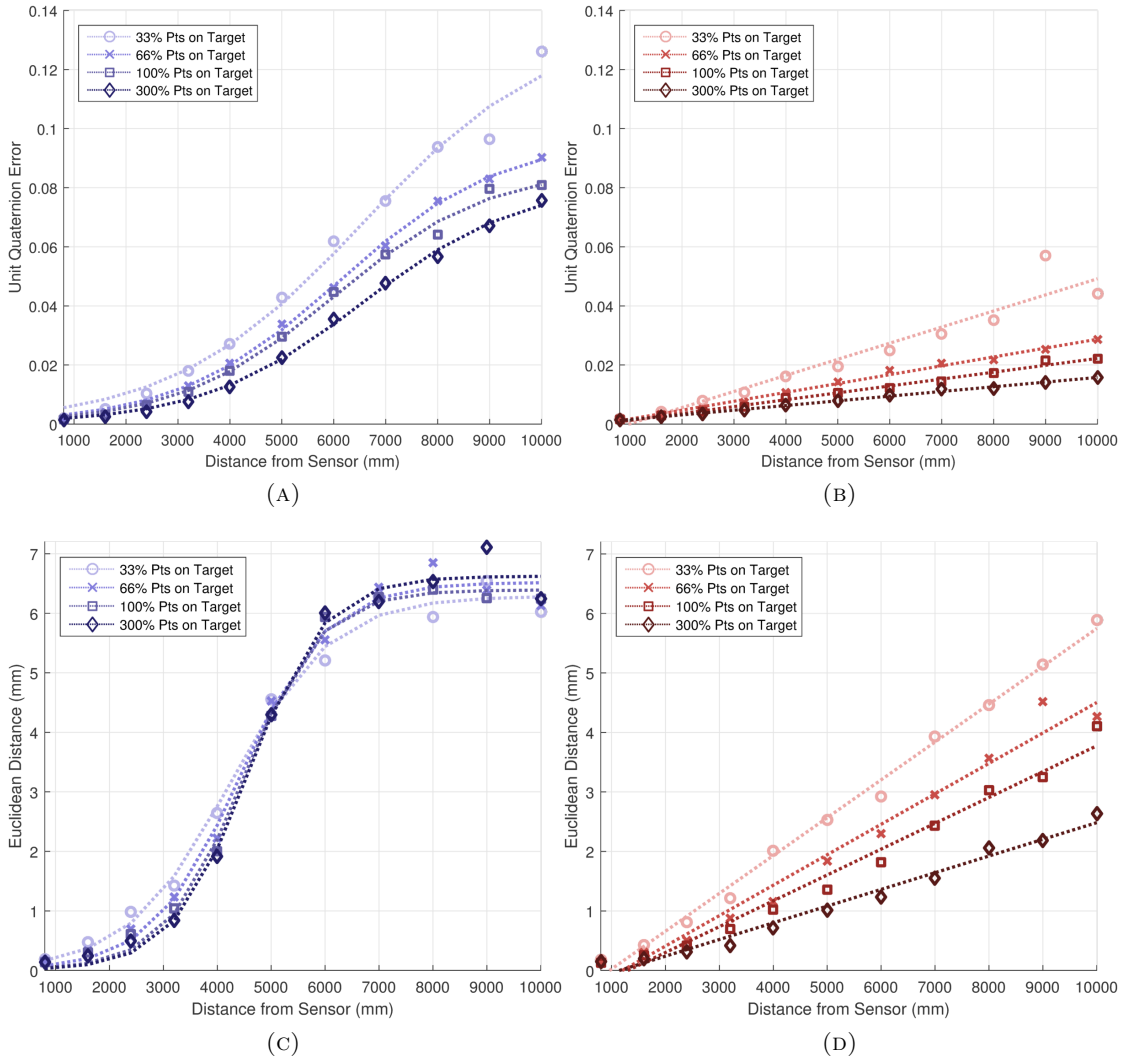


FIGURE 5.2: The number of points on target in simulated measurement point clouds is varied and measured against the full CAD model point cloud to compare the relative rotation error for the (A) A-ICP and (B) PSR-MLE cost functions, as well as the total translation error for the (C) A-ICP and (D) PSR-MLE cost functions.

In Figure 5.3, results are provided where the number of model points used by the algorithms are varied while keeping the measurement quantity fixed. These experiments test scenarios for sparse and more complex CAD models. For values larger than 100%, a denser CAD model is generated by interpolating between existing facets. The peaking phenomenon is again apparent in both the orientation and position errors when employing A-ICP near the edge of the Kinect's observable range [Figures 5.3(A) and 5.3(C), respectively]. In this case, measurements have a higher chance of being assigned to an incorrect model point as the CAD model density increases. As with the simulations presented in Figure 5.2, PSR-MLE performance monotonically increases with an increase in model points.

In the next sets of experiments, the simulator presented in Chapter 3 was utilized

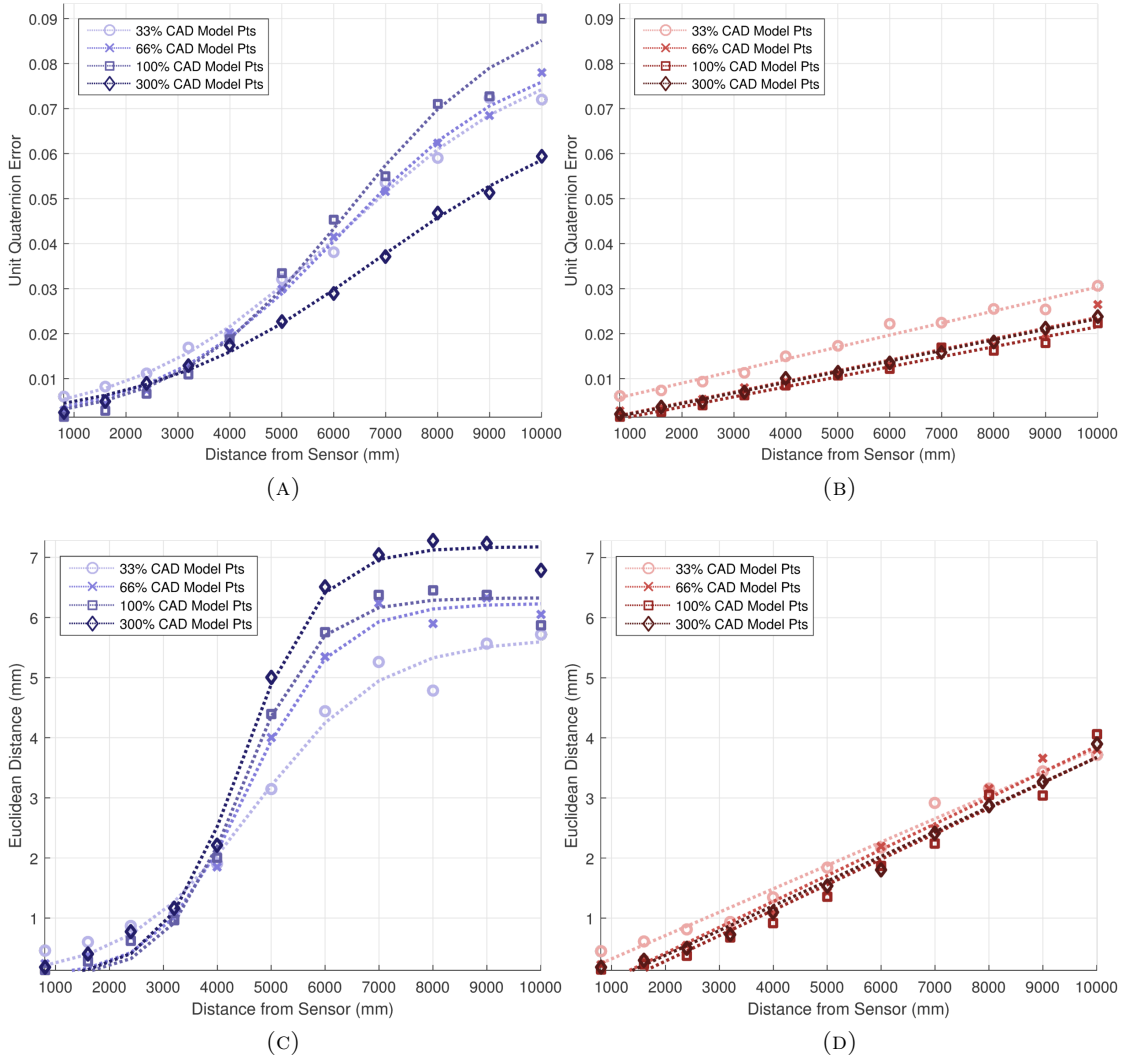


FIGURE 5.3: The complexity of the CAD model is varied and the discretized model is aligned to simulated full measurement point clouds to compare the relative rotation error for the (A) A-ICP and (B) PSR-MLE cost functions, as well as the total translation error for the (C) A-ICP and (D) PSR-MLE cost functions.

to generate noisy structured-light depth images, where different PSR pose estimation schemes were tested on the corresponding transformed pseudo-measurement point clouds. Here, the teapot was positioned in the center of the focal plane at a depth of 2400 mm (the midpoint of Kinect's operational range), and oriented as shown in Figure 4.5(A). The object size was varied to generate between 50 and 5000 pixels on target, and the surface material was assumed to be ideally diffuse, following the models presented in Sections 3.2 and 3.3. Accordingly, these IR intensity and noise models were utilized in the simulator to generate a set of 100 noisy depth images for each experimental setup. Note, the axial and lateral error models presented in Section 3.2 were utilized as the inhomogeneous depth error approximations for the PSR methods that account for anisotropic measurement errors.

The three priors presented in Section 5.3.1 were first tested in conjunction with the PSR-MLE method on the variously sized teapots, where the results are plotted in Figure 5.4. Here, the errors of the pose parameters correspond to the square root of the diagonal elements of the method’s sample MSE matrix. Perhaps not surprisingly, the fully filtered model point cloud methodology via ray casting performs only marginally better than the proposed fast facet visibility calculation from (5.4). As seen in Figure 5.1(B), the fast selection distribution methodology filters most self-occluded facets for the teapot under the tested orientation, and is therefore able to achieve comparable accuracy results. The unfiltered model point clouds, on the other hand, mostly underperform when compared to either computation of visibility method, especially for estimation of Y-axis rotation for all object sizes, X and Z-axis rotation for larger object sizes, and Z-axis translation for smaller object sizes. This is expected, because this prior algorithm assumes the object as being transparent, which is unrealistic for sensor measurements.

Lastly, the PSR-MLE method in conjunction with the fast facet visibility calculation was compared to performance achieved by the ICP and A-ICP cost functions, as well as the original CPD implementation¹, where the results are plotted in Figure 5.5. Note, in order to fairly compare the local and global optimization pose estimators, multiple searches were performed for each data set with varying initialized pose region sizes (between 2 – 40 degrees and 3 – 15 mm for each pose parameter), where the best estimate was saved. In this set of experiments, performance comparison is less definitive than in the ideal cases presented in Figures 5.2 and 5.3. The PSR-MLE mostly achieves the lowest MSE estimates, especially for estimation of Y-axis rotation for all object sizes. A modest improvement in estimation of X and Z-axis rotation, as well as X and Y-axis translation was also achieved for larger object sizes, though each PSR method seems to perform nearly equally for estimation of Z-axis translation. The most important take-away from the analyses of these simulated pseudo-measurement point cloud experiments, however, is that none of the tested PSR methods are able to achieve the estimated lower bounds on accuracy established in Section 4.1, which are depicted by the solid black line CRBs in Figures 5.4 and 5.5. This is due in large part to the inhomogeneous error sources introduced by processing structured-light IR images into depth images. Since PSR methods cannot account for object or scene-specific surface characteristics, the best they can do is limited to modeling errors that depend on the sensor only.

¹See: Matlab Central’s File Exchange resource center for [\(Robust Nonrigid Point Set Registration\)](#)

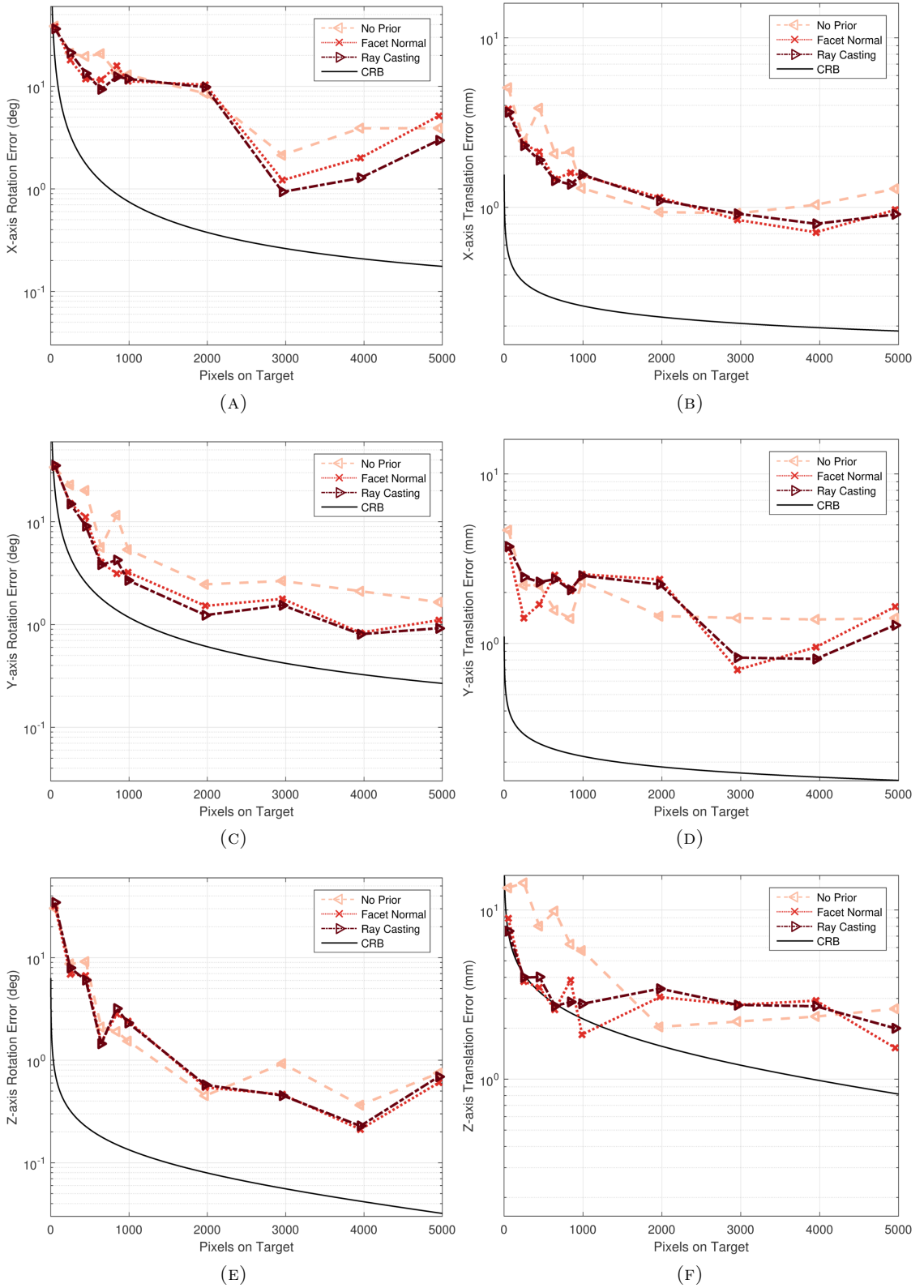


FIGURE 5.4: The log scale of pose parameter standard errors of the PSR-MLE method with three different point priors are compared for simulated pseudo-measurement point clouds of a teapot positioned at the center of the focal plane and at a depth of 2400 mm.

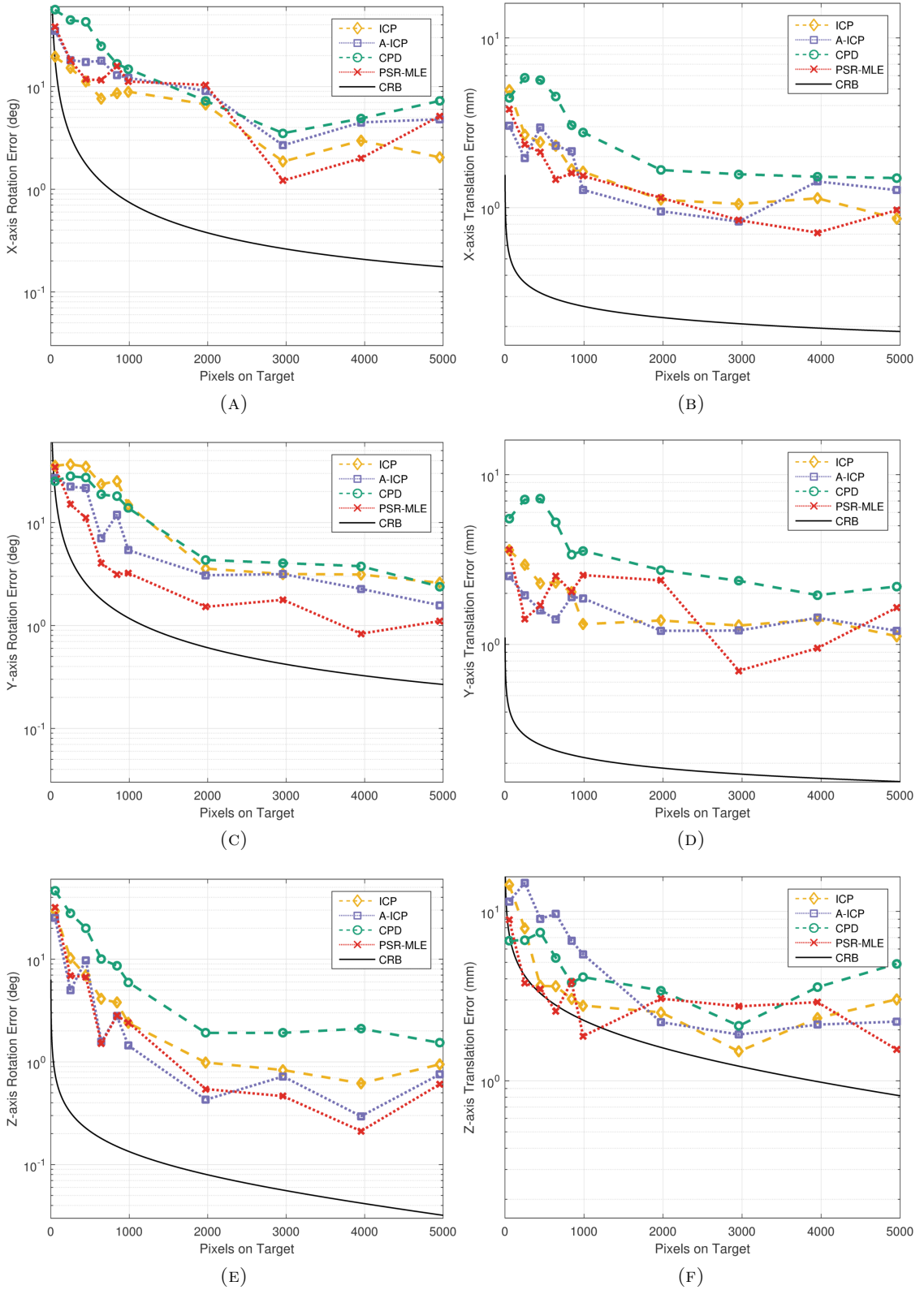


FIGURE 5.5: The log scale of pose parameter standard errors from four PSR methods are compared to the CRB of simulated structured-light data for a teapot positioned at the center of the focal plane and at a depth of 2400 mm.

6

IR IMAGE PREDICTION

Mathematical reasoning may be regarded rather schematically as the exercise of a combination of two facilities, which we may call intuition and ingenuity.

– Alan Turing (1938)

This chapter is structured as follows: Section 6.1 gives an overview of sensor-based image prediction methods for various range imaging devices. In Section 6.2, the proposed SLIR-MLE pose estimator is formulated that operates directly on the measured IR images. SLIR-MLE is finally shown to consistently outperform cutting edge 6D object pose estimators in Section 6.3, as well as nearly achieve the estimated error bounds, thereby demonstrating near optimality.

6.1 Background

Another approach for model-based shape matching utilizes the sensor physics and the statistics of object scintillation and sensor noise. Given a model of the object, a model for generating an image, and the statistical PDFs, likelihoods can be computed for the conditional density of pixel measurements given an object’s pose hypothesis. This class of methods takes full advantage of object-sensor interactions and has the potential for superior performance since they work closer to the raw, unprocessed measurement data, though they rely on accurate, calibrated sensor models. Accordingly, sensor-based image prediction has previously been applied to SAR images [27], lidar range data [62, 121], and ToF [99].

Estimating pose from information early in the structured-light sensing/processing chain has the potential to alleviate errors induced in subsequent nonlinear processing steps that are in fact locally scene dependent (Figure 6.1). Moreover, all 3D information is

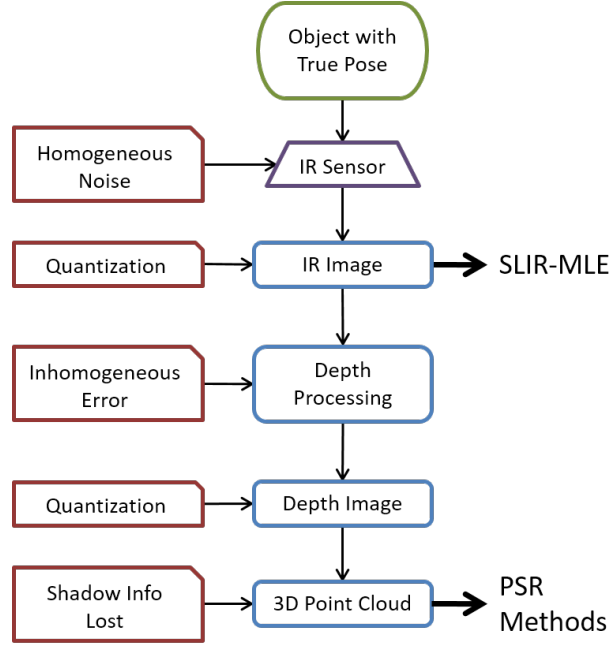


FIGURE 6.1: By operating on the more informative IR images, inhomogeneous error sources are avoided, providing the potential for optimal pose estimation.

contained in the IR image produced by the stereo system, and the speckle and thermal noise are more nearly homogeneous. Despite this, image matching has not yet been applied to the raw Kinect IR images, most likely due to the lack of a complete and reliable model for the input IR image statistics and light pattern. However, due to the extensive analysis provided in Chapter 2, as well as the high-fidelity structured-light IR and depth image simulator presented in Chapter 3, sensor-based image prediction is made possible. This motivates the method presented in this chapter, which excludes the extra, inhomogeneous error sources from both measurement and prediction, and nearly attains the lower bounds on estimation error formulated in Chapter 4.

6.2 SLIR-MLE Algorithm

Given the sensor model, a maximum likelihood method to estimate the pose of an object by operating on the raw structured-light IR images is devised, which is referred to as the structured-light IR image maximum likelihood estimation (SLIR-MLE) method. As mentioned previously, an IR image consists of the received pattern of dots, where each dot is shifted by a different disparity based on the localized depth of the object/scene that the corresponding transmitted laser beam intersects. Each pixel in the measured image then contains either the 10-bit intensity of the received noisy IR dot energy, or is empty and corrupted only by thermal noise. A noise-free image of the measured scene can be predicted given a CAD model and hypothetical pose of the object, where the intersecting transmitter rays and mean intensity of the pixels are determined by (4.5).

The distribution of dots in a predicted image $\mu(\boldsymbol{\theta})$, which is relegated by the tested pose of the CAD model, can then be compared to the noisy measured IR image \mathbf{Z} . As follows, a conditional expression can be written to represent the intensity distribution of a pixel with or without an IR dot

$$f_Z(z) = \begin{cases} f_{\Gamma\text{MG}}(z; k, \mu(\boldsymbol{\theta}), \sigma_n) & \text{if } \textit{pix} \text{ w/ dot} \\ f_{\tilde{n}}(z; \sigma_n) & \text{if } \textit{pix} \text{ w/o dot} \end{cases}, \quad (6.1)$$

where the distribution of a pixel without a predicted dot energy is simply the thermal noise distribution from (3.14).

In order to evaluate the likelihood (6.1) of a predicted image given the measurements, the function defined by (3.8) must be evaluable for pixels with IR dot energy. Unfortunately, a closed-form solution of (3.8) does not exist. The problem can instead be simplified by assuming that the intensity distribution of pixels with IR dots can be described solely by (3.12), which is generally true when $\sigma_n \ll \mu(\boldsymbol{\theta})$. This is demonstrated in Figure 4.3(A), when the IR laser intersects an object at a range of 2400 mm. Here, the intensity distribution of the IR dot corrupted by speckle and thermal noise is virtually indistinguishable from the distribution when thermal noise is neglected. Thus, for much of Kinect's operating range of 800 – 4000 mm, (6.1) can be simplified as

$$f_Z(z) = \begin{cases} f_{\tilde{\Gamma}}(z; k, \mu(\boldsymbol{\theta})) & \text{if } \textit{pix} \text{ w/ dot} \\ f_{\tilde{n}}(z; \sigma_n) & \text{if } \textit{pix} \text{ w/o dot} \end{cases}. \quad (6.2)$$

The maximum likelihood estimate for pose $\boldsymbol{\theta}$ can then be found by maximizing the log of the product of $f_Z(z)$ for all pixels so that the incremental log-likelihood is given by

$$L_{Z;\boldsymbol{\theta}}(z; \boldsymbol{\theta}) = \ln(f_Z(z)), \quad (6.3)$$

and the composite log-likelihood for all pixels z in the IR image \mathbf{Z} given the pose of the object is

$$\begin{aligned} \tilde{L}(\mathbf{Z}, \boldsymbol{\theta}) &= \sum_{p \in P} L_{Z;\boldsymbol{\theta}}(z_p; \boldsymbol{\theta}), \\ &= \sum_{p \in P_{\text{Dot}}} \ln(f_{\tilde{\Gamma}}) + \sum_{p \in P_{\text{NoDot}}} \ln(f_{\tilde{n}}), \end{aligned} \quad (6.4)$$

where P_{Dot} and P_{NoDot} denote the number of pixels with and without IR dot energy, hence $P = P_{\text{Dot}} + P_{\text{NoDot}}$. When the prior distribution for pose $p_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ is uniform, the

pose estimate $\hat{\theta}$ is chosen as the value that maximizes (6.4), where a global optimization routine is deployed since this equation cannot be solved in closed-form. Note that prior knowledge about more or less likely poses can easily be incorporated in the algorithm by multiplying $\tilde{L}(\mathcal{Z}, \theta)$ by a non-uniform prior $p_{\theta}(\theta)$ before maximization.

The large SNR assumption starts to diminish, however, when the object is positioned at further distances from the sensor, and thermal noise has a noticeable impact on the intensity distribution $f_Z(z)$. This is demonstrated in Figure 4.3(B) when the IR laser intersects an object near Kinect’s maximum operational range. In this case, noisy IR intensities produce different likelihoods for the distributions defined by $f_{\tilde{\Gamma}}$ (green curve) and $f_{\Gamma\text{MG}}$ (blue curve), especially at lower intensity values. The ΓMG convolution could, instead, be evaluated for pixels predicted to contain a low energy IR dot at each iteration of global optimization, though in practice this requires considerably more computational expense. It should also be noted that the shorter separation between the $f_{\tilde{n}}$ (red curve) and $f_{\Gamma\text{MG}}$ distributions highlights how the correct ‘classification’ between a pixel with or without an IR dot is more difficult under lower SNR conditions. Granted, these low SNR conditions would affect the accuracy of any pose estimation method, as depth image measurements contain more error.

6.3 Results

For all experiments in this section, the objects were positioned in the center of the focal plane at a depth of 2400 mm, and oriented as shown in Figures 4.5(A), 4.5(B), and 4.5(C). The object sizes were again varied to generate between 50 and 5000 pixels on target, and the surface material was assumed to be ideally diffuse, where the Chapter 3 simulator generated a set of 100 noisy IR and depth images for each experimental setup. Note, the Utah teapot, Stanford bunny, and Stanford dragon are each unique and asymmetrical about the axes of rotation, and the pose parameters are therefore identifiable.

The proposed SLIR-MLE method was first tested on the noisy IR image sets of the objects with varying sizes, where Matlab’s built-in genetic algorithm was implemented to find the global maximum likelihood on the continuous 6D pose domain. Figure 6.2 provides the sample histograms of the resulting pose errors in the form of box plots. This figure shows that the variance of the SLIR-MLE errors clearly shrink as the number of pixels on target increases. Moreover, the pose estimates converge to the true pose parameter, i.e. the bias tends to zero, which demonstrates the method’s consistency.

The error distributions also become symmetrical about the true pose parameter as the number of pixels on target increases in Figure 6.2, where the mean converges towards the median and falls directly between the first and third quartiles, demonstrating asymptotic normality. This property is further supported in Figure 6.3, where the SLIR-MLE sample MSE (blue curve) approximately achieves the CRB for each pose parameter across all

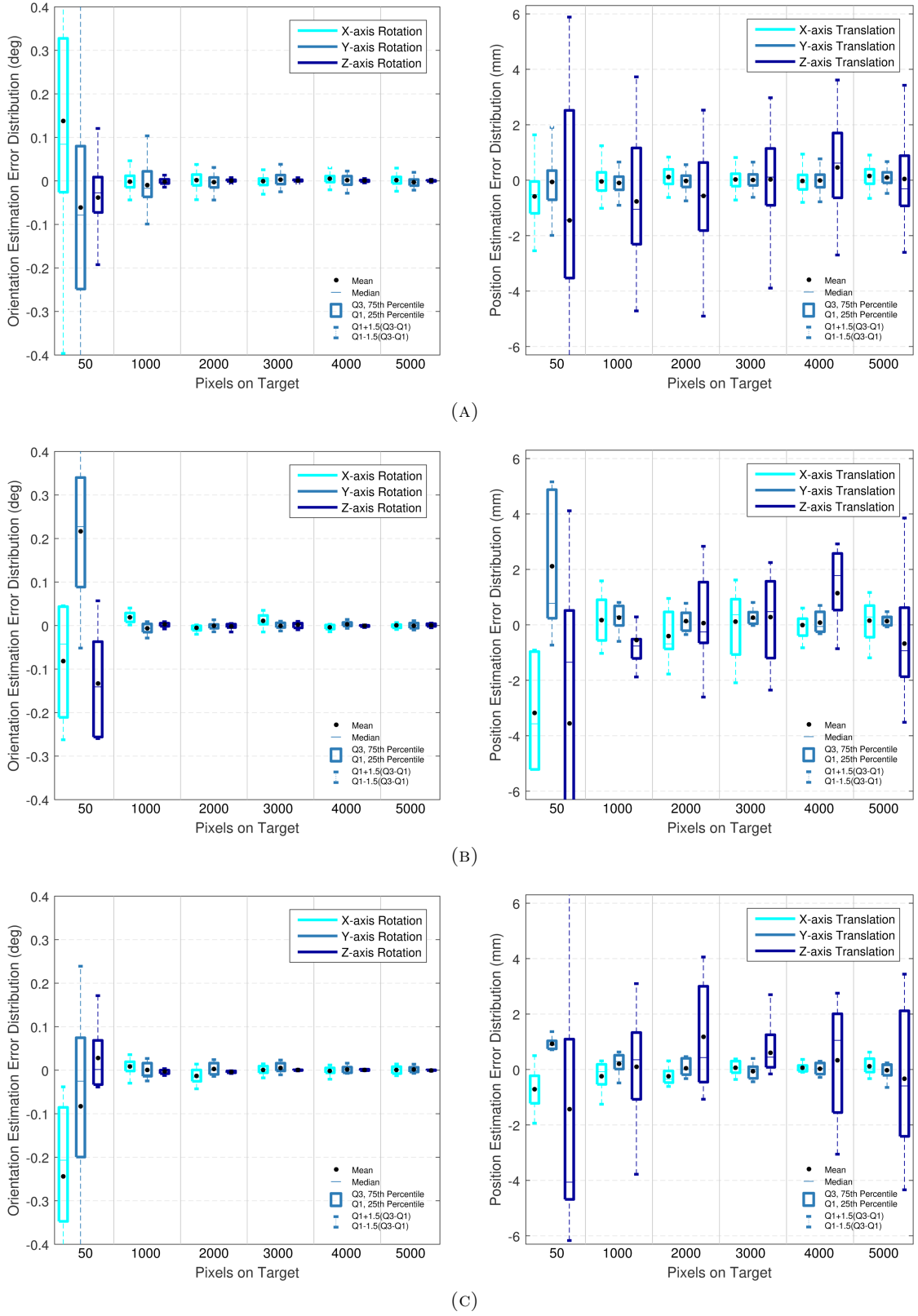


FIGURE 6.2: The error distributions of the SLIR-MLE method converge and become symmetrical about the true pose parameter when more measurements are available from larger (A) teapots, (B) bunnies, and (C) dragons at a depth of 2400 mm.

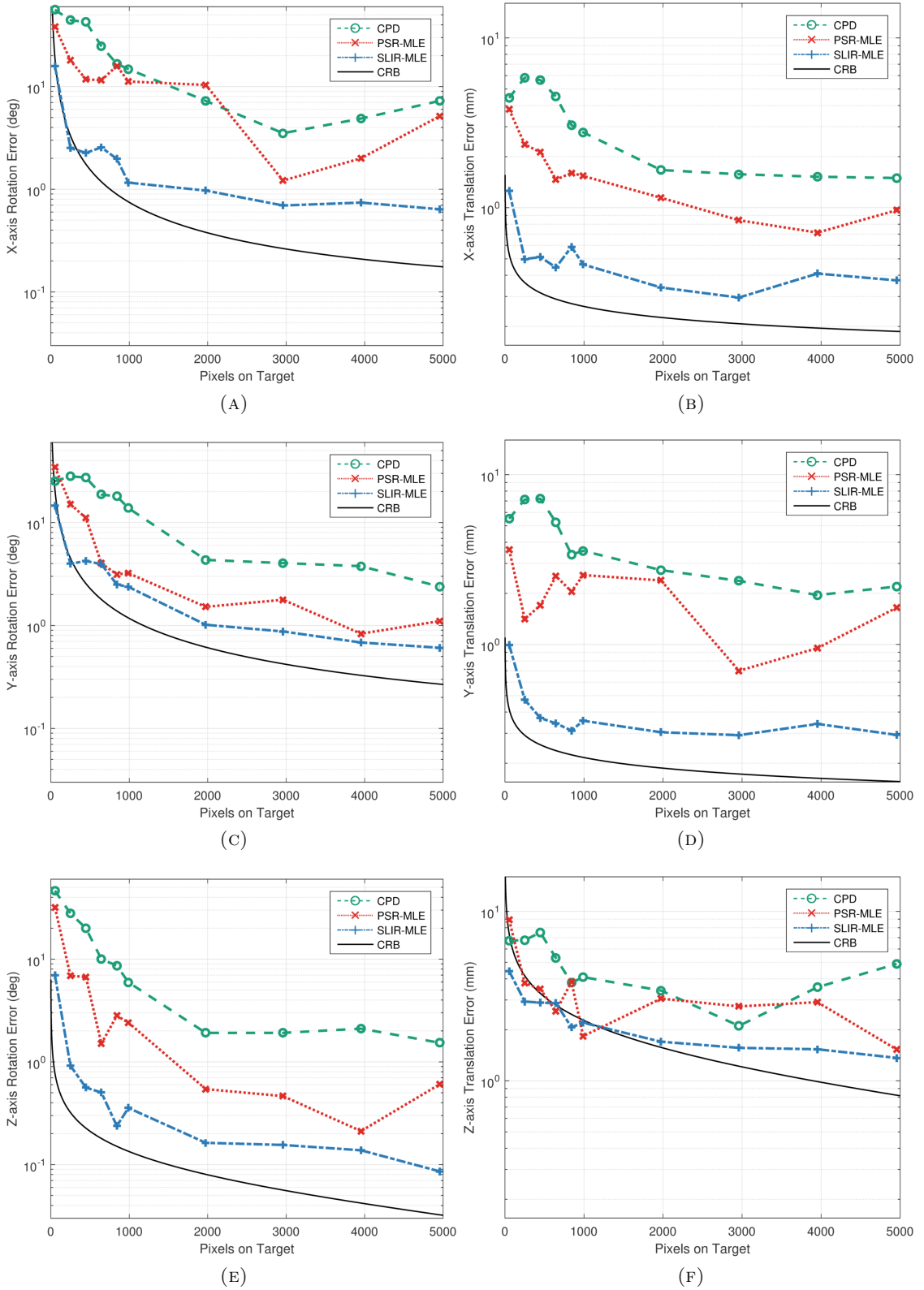


FIGURE 6.3: The log scale of pose parameter standard errors from two PSR methods and the proposed SLIR-MLE method are compared to the CRB of simulated structured-light data for a teapot positioned at the center of the focal plane and at a depth of 2400 mm.

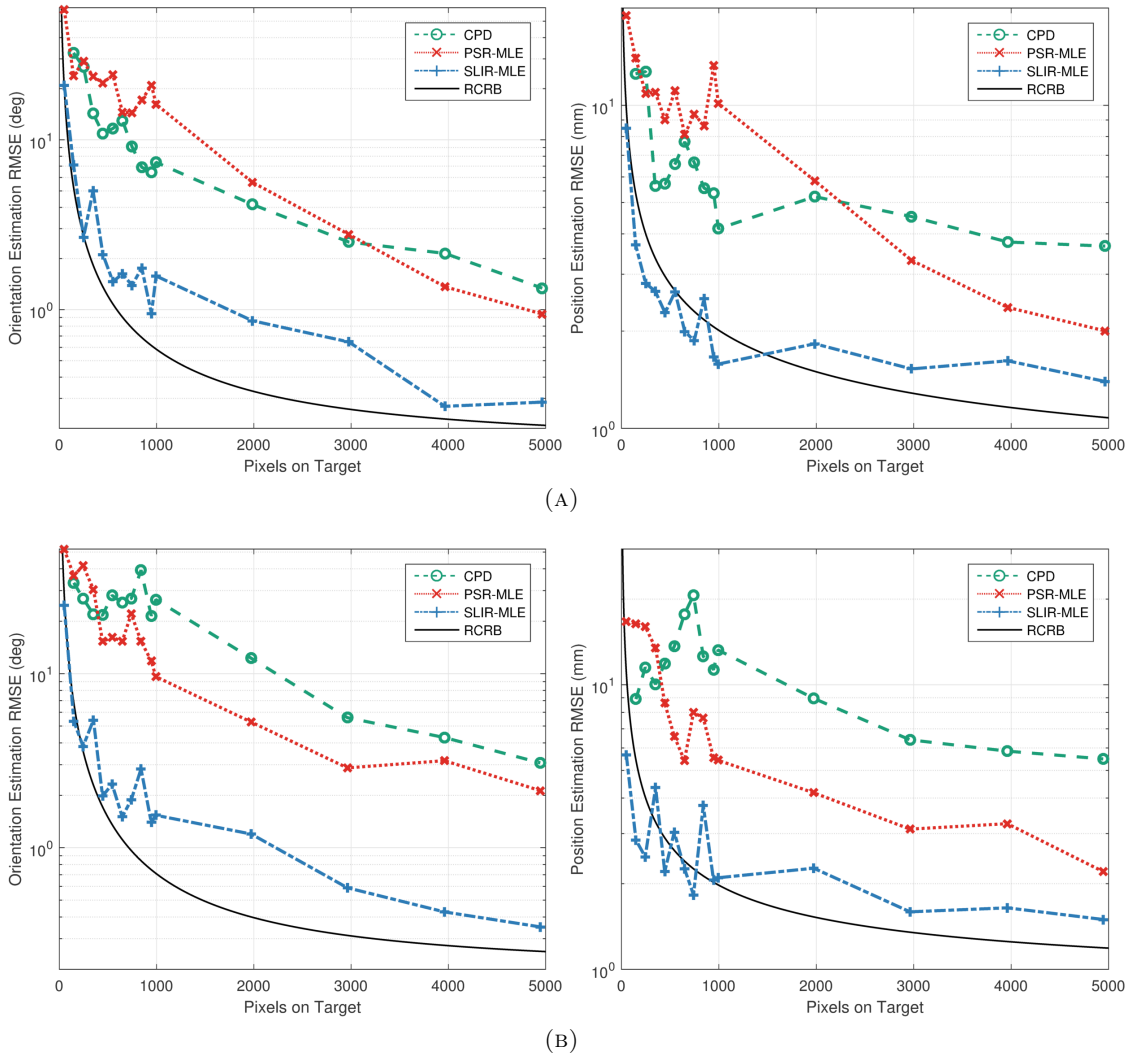


FIGURE 6.4: The log scale of RMSEs from two PSR methods and the proposed SLIR-MLE method are compared to the RCRBs of simulated structured-light data for a (A) bunny and (B) dragon positioned at the center of the focal plane and at a depth of 2400 mm.

teapot experimental data sets. Here, the errors of the pose parameters again correspond to the square root of the diagonal elements of the method's sample MSE matrix. The proposed method also approximately achieves the RCRB for the bunny and dragon data sets in Figures 6.4(A) and 6.4(B), where the root MSE (RMSE) for orientation and position estimation are computed similarly to (4.15). Note, minimizing the RMSE is a commonly implemented criterion, often associated with the 'A-optimality' criteria for least squares estimators. Occasionally, the SLIR-MLE marginally underperforms (and in some cases, overperforms) the lower error bound when viewed on an amplifying log scale, which can be attributed to the assumptions made for the estimator and CRB calculation, as well as the finite sample sizes. Despite this, the method straddles the CRB border even when there are fewer pixels on target; it is therefore considered a nearly optimal and asymptotically efficient estimator for Kinect data. The results from the teapot

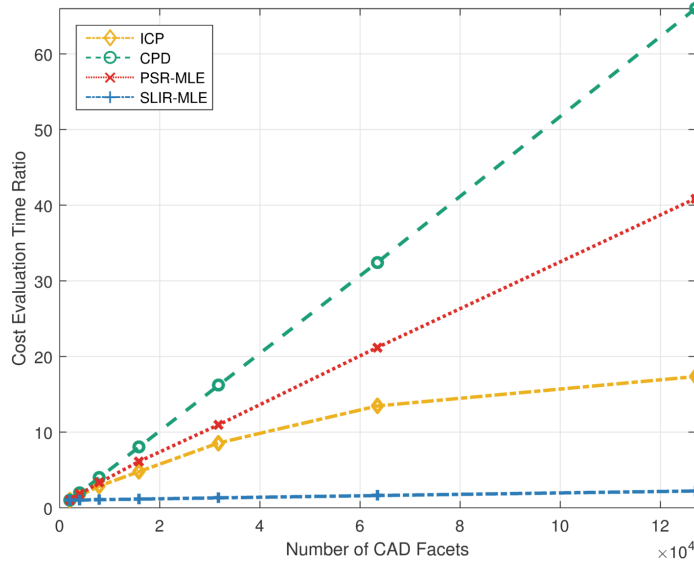


FIGURE 6.5: The time complexity of cost function evaluations for the proposed SLIR-MLE method does not significantly increase with an increase in CAD model complexity for IR images of a teapot with 1000 pixels on target.

experiments achieved by the CPD and PSR-MLE methods obtained in Section 5.4, as well as the results achieved by the two PSR methods from the simulated bunny and dragon pseudo-measurement point clouds, are also compared to the SLIR-MLE results in Figures 6.3 and 6.4. In each experimental setup for all three objects, the two methods consistently underperformed, and achieved MSEs at nearly an order of magnitude worse than the SLIR-MLE method.

The proposed method also has computational advantages when compared to PSR methods. Firstly, the time complexity of SLIR-MLE cost evaluation depends linearly on the number of sub-rays representing a single simulated IR dot. This translates to nearly identical cost evaluation times as a function of CAD model complexity, because ray casting algorithms that build collision structures with fast kd-Trees [109] yield $\mathcal{O}(\log n)$ intersection calculations per ray [42], where n represents the number of facets. On the other hand, the time complexity of PSR cost evaluations can grow much faster than linearly with CAD model complexity, depending on the point pair correspondence algorithm. This is shown in Figure 6.5, which demonstrates how each method scales with increased CAD complexity. Here, CAD complexity was increased by again interpolating between existing facets to double the total number of facets for each complexity experiment. The data samples correspond to the ratio of the complexity experiment’s mean cost evaluation processing time to the initial experiment’s mean time, where the mean is derived from the clock time for each method’s cost function evaluated at 100 random pose trials. Note, the mean times for CPD, PSR-MLE, and SLIR-MLE are respectively 101.0 ms, 34.7 ms, and 307.9 ms for the initial experiment of a teapot with 992 facets, which are relegated by the computer hardware and optimized code architectures. It is also clear that the negative log-likelihood surface of the SLIR-MLE plotted over all X and Y-axis

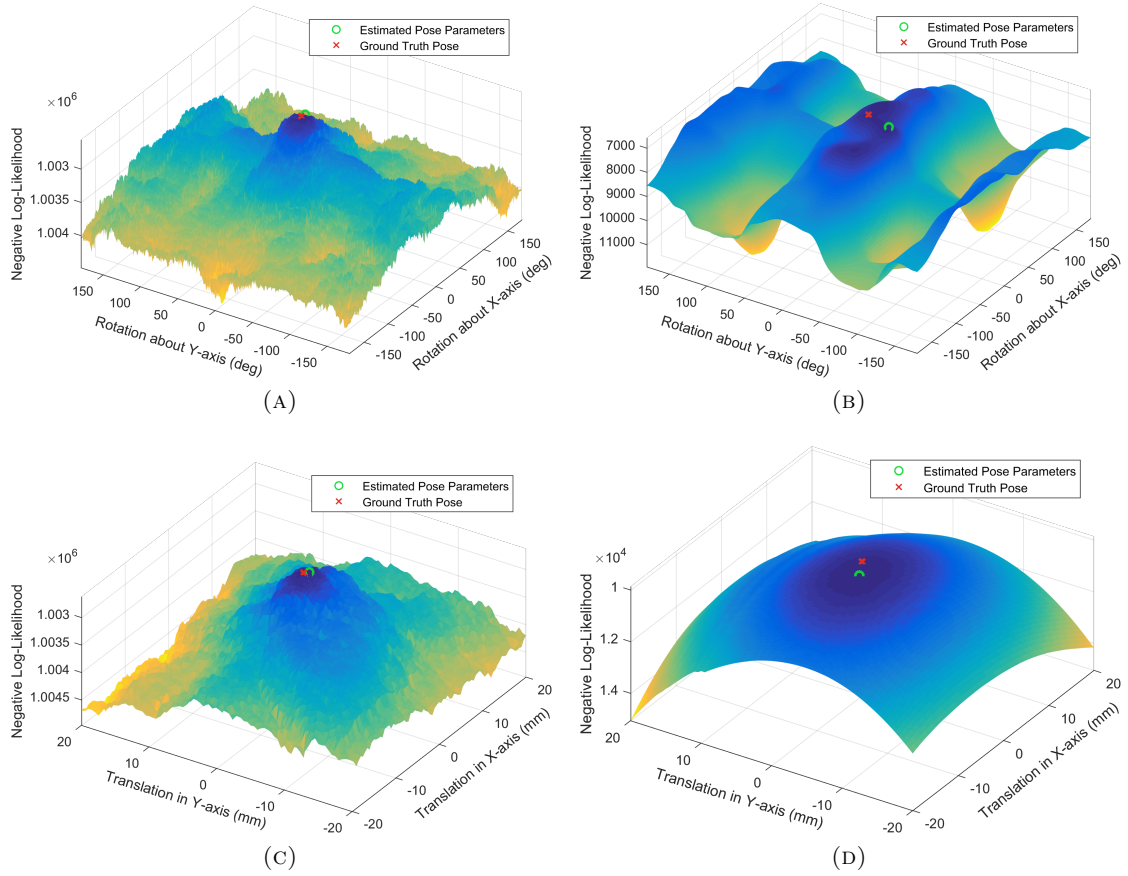


FIGURE 6.6: The negative log-likelihood surface of the proposed SLIR-MLE is plotted over all (A) X and Y-axis rotations and (C) X and Y-axis translations for a teapot positioned at a depth of 2400 mm with 550 pixels on target, which demonstrates a clear single global optimum near the true pose parameter. Conversely, the PSR-MLE method generates a negative log-likelihood surface with several local wells for rotation estimation as shown in (B), and a broad well near the true pose parameter for rotation and position estimation as shown in (B) and (D). Note, in each plot, the negative log-likelihood axis is inverted.

rotations contains a single global optimum near the true pose parameter for a noisy IR image of a teapot with 550 pixels on target, as seen in Figure 6.6(A). Note, these observations are true for the other pose parameters such as translations in the X and Y-axes as shown in Figure 6.6(C), and when the number of pixels on target are varied as shown in Figures 6.7(A) and 6.7(C). This allows SLIR-MLE to consistently converge in the neighborhood of the true pose parameter when the initialized error bounds were set to 5 degrees and 4.5 mm, 20 degrees and 9 mm, and 40 degrees and 15 mm for the global optimization routine, as shown in Figure 6.8. In contrast, the CPD and PSR-MLE methods would often converge in an incorrect pose neighborhood when the search was initialized with a modest error. This is due to the existence of many local wells in the PSR negative log-likelihood surfaces, as well as a broad well near the true pose parameter from the teapot with 550 pixels on target [Figures 6.6(B) and 6.6(D)], and a global optimum well that deviates from the true pose parameter for rotation estimation from

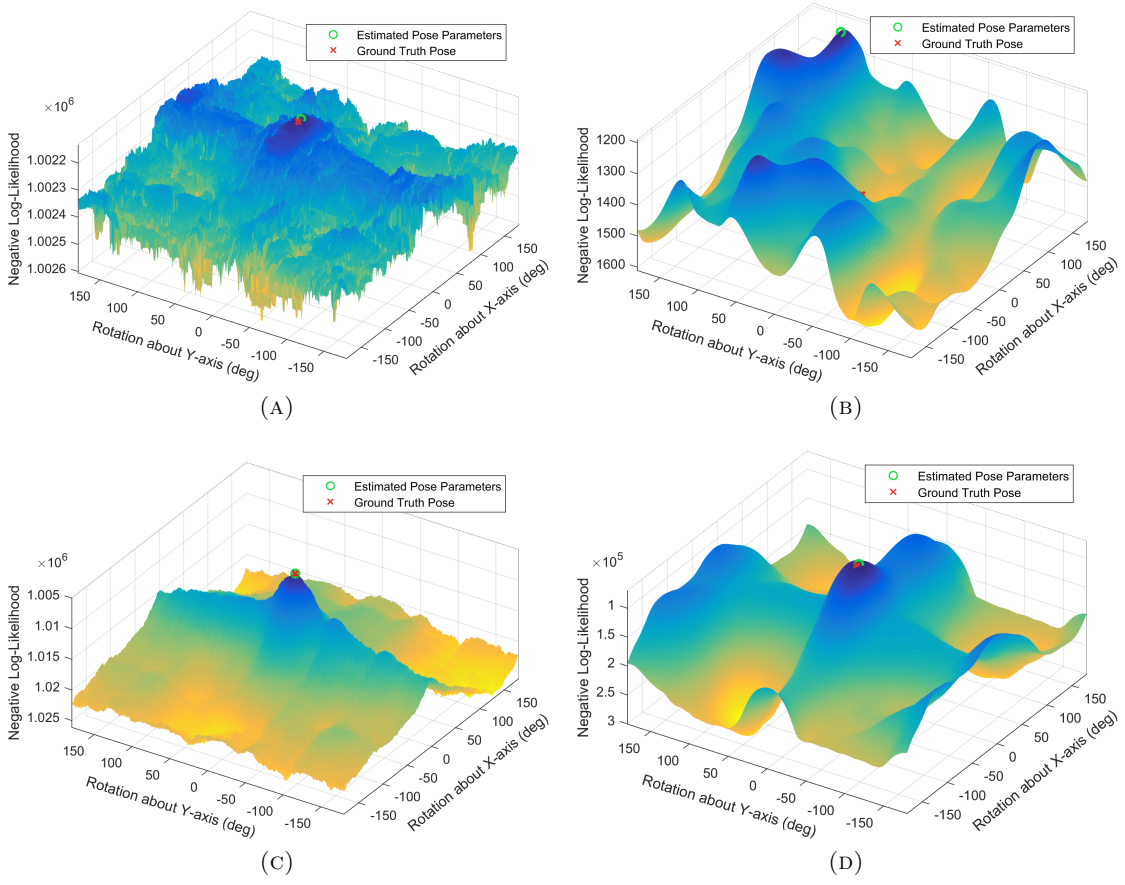


FIGURE 6.7: The negative log-likelihood surfaces are plotted over all X and Y-axis rotations for a teapot at a depth of 2400 mm with (A) 150 and (C) 5000 pixels on target for the proposed SLIR-MLE method, and (B) 150 and (D) 5000 pixels on target for the PSR-MLE method. Note, in each plot, the negative log-likelihood axis is inverted.

the teapot with 150 pixels on target [Figure 6.7(B)]. Finally, since SLIR-MLE uses a model to render and predict images, the need to store a library of hundreds or thousands of object viewpoints is alleviated.

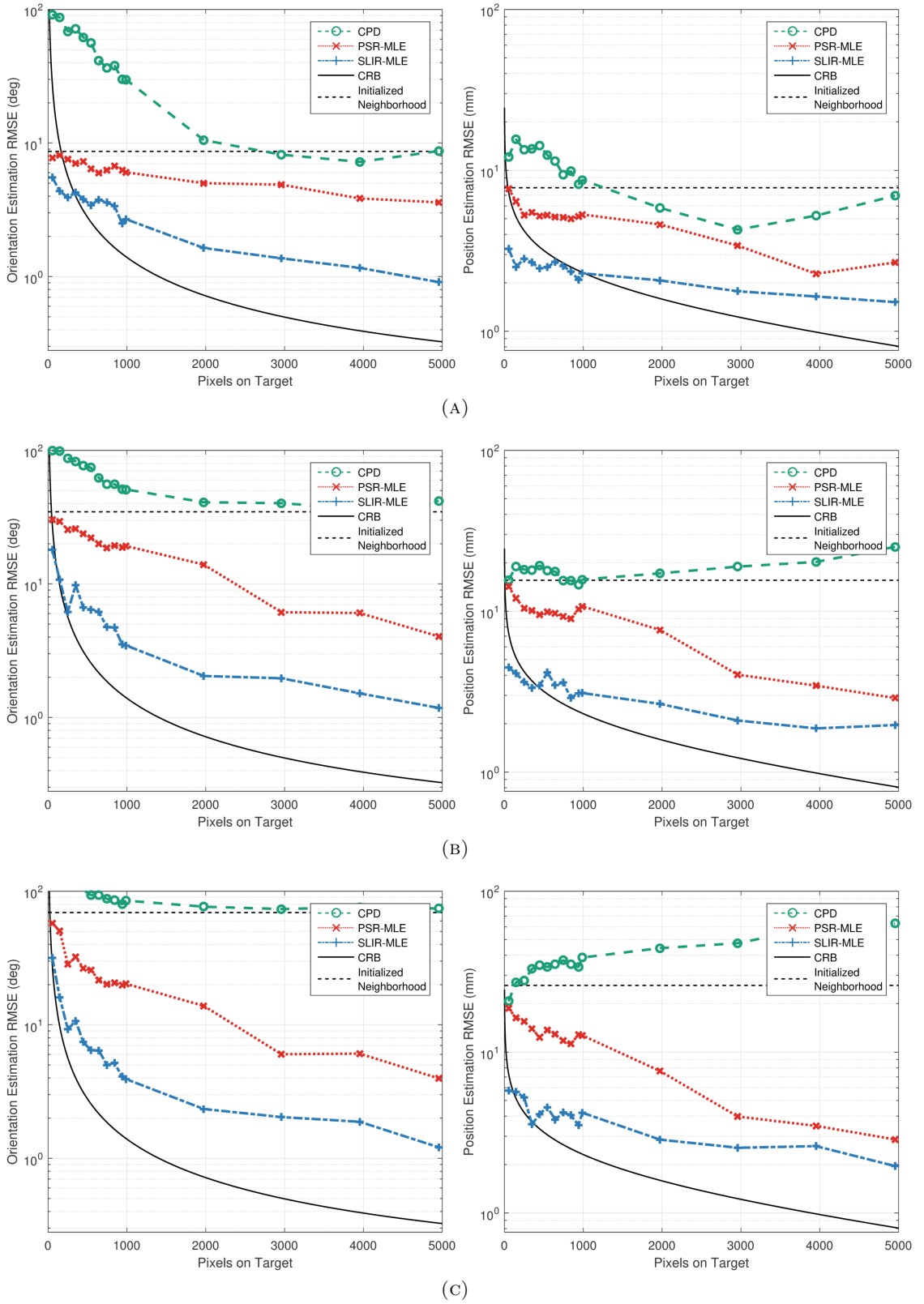


FIGURE 6.8: The proposed SLIR-MLE method consistently converges in the neighborhood of the true pose parameter when the initialized error bounds were set to (A) 5 degrees and 4.5 mm, (B) 20 degrees and 9 mm, and (C) 40 degrees and 15 mm for a teapot at a depth of 2400 mm.

7

CONCLUSION AND FUTURE WORK

The road goes ever on and on, down from the door where it began. Now far ahead the road has gone, and I must follow, if I can.

– J. R. R. Tolkien (1937)

7.1 Summary of Dissertation

Structured-light depth sensors have emerged as a widely used class of commodity-priced sensors for many industrial, robotic, and household applications that require accurate 6D pose estimates of various objects with known CADs, which is due in part to their simplicity in design and ability to process depth data with minimal computational expense. Certain structured-light light coding methods can, however, lead to a loss of information, as well as inhomogeneous depth errors that depend on the composition and properties of the object and scene. This results in a reduction of potential accuracy for model-based pose estimation methods that operate on the depth images or subsequently transformed 3D point clouds, such as the popular class of PSR methods.

In order to fully understand and exploit the underlying sensing mechanism and information content, this dissertation makes four principle contributions. The first contribution provides a detailed model to produce mean (i.e. expected) or noisy realizations of structured-light IR and depth images, which was motivated by an extensive study of the Kinect sensor’s underlying mechanisms and performance characteristics, as well as newly constructed empirical models for the intensity, speckle, and detector noise of the received IR dot pattern. Since the simulator accurately recreates salient image artifacts and has validated error statistics, researchers may use it as a tool to provide ground truthed data sets of any object/scene with accompanying CAD models. The proposed model can also be applied to a wide set of other applications that include constructing richer axial and lateral error models, improved object detection, classification, shape

inspection, and pose estimation, and developing a simulator for other depth sensors that employ a structured-light system.

The second contribution is the derivation of an information theoretic framework that is utilized to establish the sufficiency, identifiability, and completeness of structured-light IR image data sets. This framework includes the formulation of an expression for the CRB, which is computed by exploiting the uncovered properties of the Fisher information of IR images given an object pose. The significance of the framework is that it provides insight and an explanation as to why operating on the raw, unprocessed IR images of structured-light depth sensors is the only medium that offers the promise of optimal pose estimation. A corollary observation is that the common approach of using heavily processed depth images or transformed pseudo-measurement point clouds as the basis for model-based shape matching is decidedly suboptimal.

The third contribution presents a PSR method to align the CAD model of an object to a noisy measurement point cloud, which is robust for CAD model complexity, anisotropic measurement noise, and sparse and incomplete data sets. This method, referred to as the PSR-MLE method, adapts a Bayesian object classification model, which performs a many-to-many soft assignment between the measurement and model point clouds. PSR-MLE is shown to outperform a variant of the widely utilized ICP method for simulated point clouds, as well as modestly improve performance for pseudo-measurement point clouds transformed from simulated noisy depth images when compared to other PSR architectures.

The fourth contribution is the construction of an asymptotically optimal and efficient 6D pose estimator with respect to the CRB that operates within the raw IR image medium, which is designed for rigid objects sensed by structured-light depth sensors. This pose estimator, referred to as the SLIR-MLE method, is adapted from the simulator to predict IR images based on the known projected light pattern, transmitter/receiver physics, and calibrated IR intensity and noise distributions. The proposed method is shown to perform an order of magnitude better than current PSR methods, and to nearly achieve the calculated CRB with varying amounts of pixels on target. Moreover, SLIR-MLE consistently converges in the neighborhood of the true pose parameter, regardless of the number of pixels on target or initialized global optimization error bound, and the computational expense of the cost function evaluation is invariant to CAD model complexity.

In order to apply SLIR-MLE concepts to other model-based shape matching applications, the resulting likelihoods generated from comparing predicted IR images to the noisy measurements could be utilized. For instance, a unique threshold could be applied to each *composite* likelihood from an object recorded at different viewpoints in order to detect a coarse anomaly for shape inspection analysis, where the thresholds could be precomputed for each pose (Figure 7.1). If a likelihood falls below the selected threshold, an ‘out-of-tolerance’ notification could be reported indicating a deviation detection.

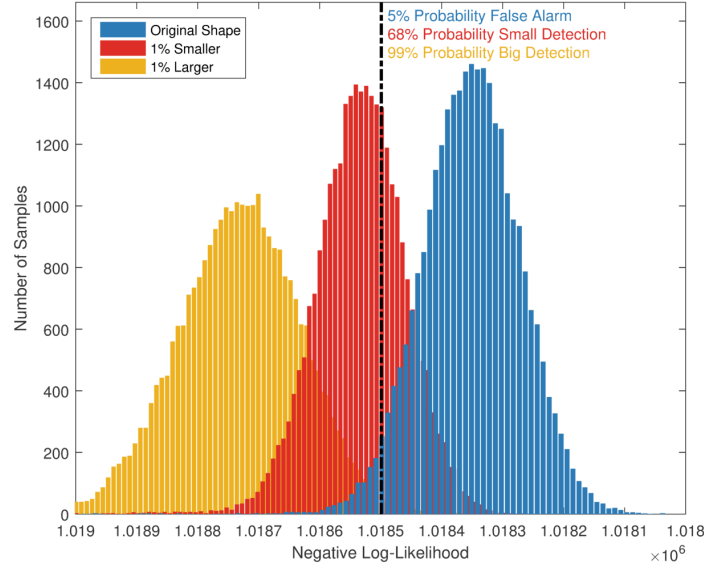


FIGURE 7.1: A threshold is selected to generate a 5% probability of false alarm from a distribution of composite likelihoods generated from noisy IR images of a perfectly constructed and aligned teapot with 5000 pixels on target. This threshold would produce a 68% probability of detection for a teapot that is 1% smaller than the original size, and a 99% probability of detection for a teapot that is 1% larger.

Several frames could then be collected to compute an average IR image at the alerting viewpoint to increase SNR, and if the anomaly is confirmed via a subsequent composite likelihood threshold, the *incremental* likelihoods could then be compared to another set of predetermined thresholds in order to localize the aberration. Note, depending on the user's tolerance for false positives or false negatives, the discrimination thresholds could be varied based on where the corresponding performances fall on a ROC curve. Similar to shape inspection, a detection threshold could also be deployed to determine if an object exists within a scene. Additionally, pose estimation could be extended to classification, where the maximum composite likelihood generated from a set of aligned CAD models would be selected as the correct object.

7.2 Calibrating the Sensor Model

Since the simulation presented in Chapter 3 corresponds to a representative but nevertheless ideal structured-light system, a fully automated and comprehensive calibration architecture needs to be devised. This architecture would acknowledge imperfect physical discrepancies among structured-light devices, as well as the variation in IR intensity and noise distributions. Calibration could be performed in three different areas that are dependent on an imperfect projection of the light pattern or detector response, as well as the variability of object and scene material properties:

IR Pattern Dependent Calibration

As mentioned in the preamble, the proposed system is constructed to easily adopt other light projection systems with a known, spatially fixed pattern. The simulation may therefore be extended to other depth cameras (e.g. the structured-light sensors listed in Table 2.1), provided that the system parameters can either be obtained, or estimated by following the calibration methods presented in Sections 3.3 and 3.4.

While manufacturers commonly implement the same IR pattern within each sensor replica, small discrepancies in the factory installed diffuser unit result in differences between the projected patterns of each sensor when placed at the same position and orientation. For instance, as shown in Figure 2.3, the projected patterns from each Kinect diverge by a different 3D rotation. Thus, the best match between a measured flat wall projection and a 3D rotated idealized IR projection can be obtained to estimate the orientation of the diffuser installation within a given sensor replica.

Imperfect pattern propagation may also occur from different depth sensors. For instance, since the Kinect diffuser unit contains two DOEs arranged in series to generate and propagate the distribution of bright and dark spots into a 3×3 grid of sub-patterns, significant pincushion and moderate barrel distortion occurs. Lens distortion can potentially be accounted for by using Brown's model [20] to estimate radial parameters, which has been previously adopted in [55] and [25]. Additionally, a treatment of the bright center dots as anchor points could be incorporated to determine the 3D rotation required to align the 3×3 grids. The lens distortion and rotation parameters can then all be jointly estimated by optimizing the alignment of the center dots.

Detector Dependent Calibration

Each photoconductor in the CMOS sensing array receives photons from the detected IR pattern, which are interpreted as a current or voltage drop, independently for each detector [103]. Accordingly, the responsivity of each detector may differ; therefore it may be possible to estimate and store each detector's actual responsivity relative to unity. It should be noted though that calibrating each detector, while technically feasible, may prove to be tedious and ultimately unnecessary. This is in accordance with the trend that current digital circuitry designs (e.g. CCD and CMOS) are fairly resistant to noise, and have a strong, nearly symmetric response.

Surface Property Dependent Calibration

In this dissertation, the effects that the SNR, IR dot pattern, object complexity, and number of pixels on target have on the tightness of the CRB and estimator efficiency were explored. However, as Betke *et al.* [15] observe, the surface texture is also an important factor to consider. In particular, surfaces with little texture, as well as reflective and absorptive surfaces, provide more of a challenge when estimating the unknown parameters. For instance, specular surfaces can reflect most of the energy from an incoming IR beam

onto a patch of detectors that deviates from what is expected from diffuse scattering. This results in poorly estimated and missing depth data for many active 3D scanners, as shown in Figures 1.2 and 3.3. These properties can however be managed in a modified sensor model that adopts ray tracing [86], provided that the albedo information for each CAD facet is calibrated. A model for objects that exhibit a mixture of diffuse and specular properties can therefore be included, and the SLIR-MLE method’s resilience to a weakly detected IR pattern, as well as its ability to account for an imperfect calibration, can be tested. Commodity structured-light depth sensors can then potentially be used for accurate model-based shape matching on a much wider array of objects with surfaces that are metallic, highly polished, glossy, etc.

Similarly, the empirical models for the intensity and noise distributions presented in Section 3.3 were constructed with an assumed global brightness and ambient intensity offset. External thermal radiation sources can however affect a scene’s recorded brightness and ambient intensity [71]. The sensor model can therefore be extended to account for different external environmental properties and constant ambient offset by following the presented calibration methods.

7.3 Constructing a Real-Time System

With the introduction of the proposed model-based shape matching method, there may be a need to reduce the computational resources required to produce an optimal result in order to maintain real-time processing speeds. Though the time complexity of SLIR-MLE is linear with the number of sub-rays deployed in the ray casting routine, and has a diminished $\mathcal{O}(\log n)$ dependency to CAD complexity (as shown in Figure 6.5), each cost function evaluation does require several computational steps [i.e. floating-point operations per second (FLOPs)]. There are a number of directions that could be taken in order to reduce the computational expense of the pose estimator while still maintaining optimality:

Global Optimization

The common problem in local optimization of the nonlinear 6D pose domain arises, where local optima exist in the likelihood probability distribution of any pose estimation cost function, especially for sparse measurement data sets as shown in Figure 6.6(A), 6.6(C), and 6.7(A). These local optima can occur for various reasons, such as object symmetry or noise introduced from the sensor, and would cause basic local optimization methods such as gradient descent to fail. While most global optimization routines may arrive at the pose estimate that produces the global optimum likelihood, certain methods that require multiple cost function evaluations at each iteration of optimization can take advantage of parallelization. For example, evolutionary algorithms like the genetic

algorithm, and swarm-based optimization algorithms like particle swarm optimization, are valid candidates that can make use of parallel processing computer architectures.

Anytime Algorithm Framework

In a real-time system with limited computational resources, an answer is required after each time step with new measured data to process, regardless of changing environmental factors that could vary the amount of input data at different time steps. If an anytime process were used, a reliable answer could be provided after each time step while avoiding a considerable loss in accuracy. Anytime algorithms, also referred to as interruptible algorithms, are defined as algorithms that are able to provide a valid approximation to an estimate of a problem at any time before a preset stopping condition is reached. These processes are useful for real-time systems that require estimates in short successions of time, and are flexible depending on the size of the data set and available resources at each time step. An anytime structure can initially be applied to any of the global optimization methods mentioned above (e.g. [115] and [102]), which iteratively converge towards the global optimum or true pose on the continuous domain. A new anytime pose estimation scheme may also be devised, similar to Redard's Single-Pass Tree algorithm for automatic target recognition (ATR) [87], or the hierarchical tree for joint object recognition and pose estimation proposed in [122]. Thus, when a new frame needs to be processed, global optimization on the current frame can be interrupted and still end up with a nearly optimal pose estimate.

Pose Tracking

In the current SLIR-MLE framework, pose estimation is performed instantaneously and independently for each frame of a recorded sequence of IR images. Since there is no prior information about the pose of the object given by the preceding time samples, an initial guess cannot be provided to an optimization routine, and the entire 6D pose domain needs to be searched. In regard to real-time processing, the optimization routines could benefit from a 'hot start', or an initialized pose estimate from a predicted state. A track could then take advantage of prior knowledge about the object's pose, where the track state consists of the 6D pose augmented with one or more pose rate states. Depending on the object's pose rate with respect to the sensor's frame capture rate, the object pose should not change a considerable amount between each frame, and a finer search centered on the predicted pose could therefore be employed. Additionally, the size of the search window may be determined from the confidence in the current estimate via a modified decision model inherent to the proposed method. Note, in addition to reduced computational cost, an improved pose estimate can also be obtained by fusing the track's prediction with the current frame estimate.

BIBLIOGRAPHY

- [1] “ATOS Triple Scan - Industrial Optical 3d Digitizer,” 2010. [Online]. Available: http://www.henindo.co.id/home/ATOS-Triple-Scan_EN_RevA.pdf
- [2] “Teardown of the Microsoft Kinect,” Dec. 2010. [Online]. Available: www.chipworks.com/en/technical-competitive-analysis/resources/blog/teardown-of-the-microsoft-kinect-focused-on-motion-capture
- [3] “Hardware info - OpenKinect,” Feb. 2011. [Online]. Available: http://openkinect.org/wiki/Hardware_info
- [4] “Kinect for Windows SDK 1.8,” Feb. 2012. [Online]. Available: <http://msdn.microsoft.com/en-us/library/hh855347.aspx>
- [5] “Image Sensors - MT9m001c12stm Data Sheet - Aptina Imaging,” Jan. 2013. [Online]. Available: http://www.apina.com/products/image_sensors/mt9m001c12stm/
- [6] “Specs about OpenNI compliant 3d sensor Carmine 1.09 (Short range),” Mar. 2013. [Online]. Available: <http://openni.ru/rd1-09-specifications/index.html>
- [7] “Products | Orbbec,” 2016. [Online]. Available: <https://orbbec3d.com/products/>
- [8] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. Rusu, and G. Bradski, “CAD-model recognition and 6dof pose estimation using 3d cues,” in *13th IEEE International Conference on Computer Vision (ICCV) Workshops*, Nov. 2011, pp. 585–592.
- [9] U. Asif, M. Bennamoun, and F. Sohel, “Real-time pose estimation of rigid objects using RGB-D imagery,” in *8th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Jun. 2013, pp. 1692–1699.
- [10] K.-H. Bae and D. D. Lichti, “A method for automated registration of unorganised point clouds,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 63, no. 1, pp. 36–54, Jan. 2008.
- [11] M. Baeg, H. Hashimoto, F. Harashima, and J. Moore, “Pose estimation of quadratic surface using surface fitting technique,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, Aug. 1995, pp. 204–209.

- [12] J. R. Barry, E. A. Lee, and D. G. Messerschmitt, *Digital Communication*. Springer Science & Business Media, Sep. 2003.
- [13] P. J. Besl and N. D. McKay, "A Method for Registration of 3-D Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, Feb. 1992.
- [14] M. Betke and N. Makris, "Information-conserving object recognition," in *6th International Conference on Computer Vision (ICCV)*, Jan. 1998, pp. 145–152.
- [15] M. Betke, E. Naftali, and N. Makris, "Necessary conditions to attain performance bounds on structure and motion estimates of rigid objects," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, Dec. 2001, pp. 448–455.
- [16] F. Blais, "Review of 20 years of range sensor development," *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 231–243, Jan. 2004.
- [17] M. Blum, J. T. Springenberg, J. Wülfing, and M. Riedmiller, "A learned feature descriptor for object recognition in RGB-D data," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2012, pp. 1298–1303.
- [18] L. Bo, X. Ren, and D. Fox, "Unsupervised Feature Learning for RGB-D Based Object Recognition," in *Experimental Robotics*. Springer International Publishing, 2013, no. 88, pp. 387–402.
- [19] B. J. Brown and S. Rusinkiewicz, "Global Non-rigid Alignment of 3-D Scans," in *ACM Transactions on Graphics (TOG)*. ACM, 2007.
- [20] D. C. Brown, "Close-range camera calibration," *Photogrammetric Engineering*, vol. 37, no. 8, pp. 855–866, Jan. 1971.
- [21] J. Byne and J. Anderson, "A CAD-based computer vision system," *Image and Vision Computing*, vol. 16, no. 8, pp. 533–539, Jun. 1998.
- [22] G. Choe, J. Park, Y.-W. Tai, and I. S. Kweon, "Exploiting shading cues in Kinect IR images for geometry refinement," in *27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2014, pp. 3922–3929.
- [23] B. Choo, M. J. Landau, M. DeVore, and P. A. Beling, "Statistical analysis-based error models for the Microsoft Kinect™ depth sensor," *Sensors*, vol. 14, no. 9, pp. 17 430–17 450, Sep. 2014.
- [24] J. C. Chow, K. D. Ang, D. D. Lichti, and W. F. Teskey, "Performance analysis of a low-cost triangulation-based 3d camera: Microsoft Kinect system," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 39B5, pp. 175–180, Jul. 2012.

- [25] J. Chow and D. Lichti, "Photogrammetric bundle adjustment with self-calibration of the PrimeSense 3d camera technology: Microsoft Kinect," *IEEE Access*, vol. 1, pp. 465–474, Jul. 2013.
- [26] D. F. Dementhon and L. S. Davis, "Model-based object pose in 25 lines of code," *International Journal of Computer Vision*, vol. 15, no. 1-2, pp. 123–141, Jun. 1995.
- [27] M. DeVore and J. O'Sullivan, "Quantitative statistical assessment of conditional models for synthetic aperture radar," *IEEE Transactions on Image Processing*, vol. 13, no. 2, pp. 113–125, Feb. 2004.
- [28] M. D. DeVore and X. Zhou, "Minimum probability of error recognition of three-dimensional laser-scanned targets," *Society of Photo-Optical Instrumentation Engineers (SPIE)*, vol. 6234, pp. 623 407–623 407, May 2006.
- [29] T. Deyle, "Low-Cost Depth Cameras (aka Ranging Cameras or RGB-D Cameras) to Emerge in 2010?" Mar. 2010. [Online]. Available: <http://www.hizook.com/blog/2010/03/28/low-cost-depth-cameras-aka-ranging-cameras-or-rgb-d-cameras-emerge-2010>
- [30] R. M. Fewster and P. E. Jupp, "Information on parameters of interest decreases under transformations," *Journal of Multivariate Analysis*, vol. 120, pp. 34–39, Sep. 2013.
- [31] D. Fiedler and H. Müller, "Impact of Thermal and Environmental Conditions on the Kinect Sensor," in *Advances in Depth Image Analysis and Applications*. Springer Berlin Heidelberg, 2013, no. 7854, pp. 21–31.
- [32] D. Fofi, T. Sliwa, and Y. Voisin, "A comparative survey on invisible structured light," in *Machine Vision Applications in Industrial Inspection XII*, vol. 5303, May 2004, pp. 90–98.
- [33] C. S. Fox, *The Infrared & Electro-Optical Systems Handbook. Active Electro-Optical Systems*. Infrared Information and Analysis Center, 1993, vol. 6.
- [34] G. Frankowski and R. Hainich, "DLP/DSP-based optical 3d sensors for the mass market in industrial metrology and life sciences," in *Society of Photo-Optical Instrumentation Engineers (SPIE)*, vol. 7932, Feb. 2011, pp. 79 320D–79 320D–11.
- [35] B. Freedman, A. Shpunt, M. Machline, and Y. Arieli, "Depth mapping using projected patterns," U.S. Patent US20 100 118 123 A1, May, 2010.
- [36] P. Fürsattel, S. Placht, M. Balda, C. Schaller, H. Hofmann, A. Maier, and C. Riess, "A Comparative Error Analysis of Current Time-of-Flight Sensors," *IEEE Transactions on Computational Imaging*, vol. 2, no. 1, pp. 27–41, Mar. 2016.
- [37] M. Galor, J. Pokrass, A. Hoffnung, and O. Or, "Sessionless pointing user interface," U.S. Patent US20 160 041 623 A1, Feb., 2016.

- [38] J. Garcia and Z. Zalevsky, “Range mapping using speckle decorrelation,” U.S. Patent US7 433 024 B2, Oct., 2008.
- [39] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann, “Robust Global Registration,” in *Eurographics Symposium on Geometry Processing*. Eurographics Association, Jul. 2005.
- [40] J. Geng, “Structured-light 3d surface imaging: a tutorial,” *Advances in Optics and Photonics*, vol. 3, no. 2, p. 128, Jun. 2011.
- [41] J. Glover, G. Bradski, and R. Rusu, “Monte Carlo Pose Estimation with Quaternion Kernels and the Bingham Distribution,” in *Proceedings of Robotics: Science and Systems*, 2012, vol. 7.
- [42] J. Goldsmith and J. Salmon, “Automatic Creation of Object Hierarchies for Ray Tracing,” *IEEE Computer Graphics and Applications*, vol. 7, no. 5, pp. 14–20, May 1987.
- [43] J. W. Goodman, *Speckle Phenomena in Optics: Theory and Applications*. Roberts and Company Publishers, Nov. 2010.
- [44] V. M. Govindu and A. Pooja, “On Averaging Multiview Relations for 3d Scan Registration,” *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1289–1302, Mar. 2014.
- [45] P. Graczyk and S. Mamane, “Fisher Information and Exponential Families Parametrized by a Segment of Means,” *arXiv:1402.1305*, Feb. 2014.
- [46] J. Han, L. Shao, D. Xu, and J. Shotton, “Enhanced Computer Vision With Microsoft Kinect Sensor: A Review,” *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1318–1334, Oct. 2013.
- [47] Y. Hara, S. Bando, T. Tsubouchi, A. Oshima, I. Kitahara, and Y. Kameda, “6dof iterative closest point matching considering a priori with maximum a posteriori estimation,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2013, pp. 4172–4179.
- [48] K. Harding, “Industrial metrology: Engineering precision,” *Nature Photonics*, vol. 2, no. 11, pp. 667–669, Nov. 2008.
- [49] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, “Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes,” in *2011 IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 858–865.
- [50] D. Q. Huynh, “Metrics for 3d Rotations: Comparison and Analysis,” *Journal of Mathematical Imaging and Vision*, vol. 35, no. 2, pp. 155–164, Oct. 2009.

- [51] A. Janoch, S. Karayev, Y. Jia, J. Barron, M. Fritz, K. Saenko, and T. Darrell, "A category-level 3-D object dataset: Putting the Kinect to work," in *13th IEEE International Conference on Computer Vision (ICCV) Workshops*, Nov. 2011, pp. 1168–1174.
- [52] B. Jian and B. Vemuri, "Robust Point Set Registration Using Gaussian Mixture Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1633–1645, Aug. 2011.
- [53] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, 1993, vol. 1.
- [54] E. Kayahan, O. Gundogdu, F. Hacizade, and H. Nasibov, "Autocorrelation analysis of spectral dependency of surface roughness speckle patterns," in *International Symposium on Optomechatronic Technologies (ISOT)*, Sep. 2009, pp. 235–240.
- [55] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of Kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, Feb. 2012.
- [56] K. Konolige, P. Mihelich, and A. Tsuda, "Technical description of kinect calibration," Dec. 2012. [Online]. Available: http://wiki.ros.org/kinect_calibration/technical
- [57] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 1817–1824.
- [58] M. J. Landau and P. A. Beling, "Optimal Model-Based 6d Object Pose Estimation with Structured-Light Depth Sensors," *IEEE Transactions on Computational Imaging*, pp. 1–16, 2016, (under revision).
- [59] M. J. Landau, P. A. Beling, and M. D. DeVore, "Efficacy of statistical model-based pose estimation of rigid objects with corresponding CAD models using commodity depth sensors," in *40th IEEE Conference of the Industrial Electronics Society (IECON)*, Oct. 2014, pp. 3445–3451.
- [60] M. J. Landau, B. Choo, and P. Beling, "Simulating Kinect Infrared and Depth Images," *IEEE Transactions on Cybernetics*, vol. PP, no. 99, pp. 1–14, Nov. 2015.
- [61] M. J. Landau, E. Koltsova, K. Ley, and S. Acton, "Multi-cell 3d tracking with adaptive acceptance gates," in *IEEE Southwest Symposium on Image Analysis Interpretation (SSIAI)*, May 2010, pp. 49–52.
- [62] A. D. Lanterman, "Jump-diffusion algorithm for multiple target recognition using laser radar range data," *Optical Engineering*, vol. 40, no. 8, pp. 1724–1728, Aug. 2001.

- [63] N. Le, “RealSense - A Comparison of Intel RealSense Front-Facing Camera SR300 and F200,” Feb. 2016. [Online]. Available: <https://software.intel.com/en-us/articles/a-comparison-of-intel-realsensetm-front-facing-camera-sr300-and-f200>
- [64] P. Y. Lee and J. Moore, “Geometric optimization for 3d pose estimation of quadratic surfaces,” in *38th Asilomar Conference on Signals, Systems and Computers*, vol. 1, Nov. 2004, pp. 131–135 Vol.1.
- [65] E. L. Lehmann and G. Casella, *Theory of Point Estimation*, 2nd ed. Springer New York, Sep. 2003.
- [66] D. Liu and T. Chen, “Soft shape context for iterative closest point registration,” in *International Conference on Image Processing (ICIP)*, vol. 2, Oct. 2004, pp. 1081–1084 Vol.2.
- [67] O. Lopes, M. Reyes, S. Escalera, and J. Gonzalez, “Spherical blurred shape model for 3-D object and pose recognition: Quantitative analysis and HCI applications in smart environments,” *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2379–2390, Dec. 2014.
- [68] I. Lysenkov, V. Eruhimov, and G. Bradski, “Recognition and pose estimation of rigid transparent objects with a kinect sensor,” in *Proceedings of Robotics: Science and Systems*, Jul. 2012.
- [69] J. Ma, J. Zhao, and A. L. Yuille, “Non-Rigid Point Set Registration by Preserving Global and Local Structures,” *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 53–64, Jan. 2016.
- [70] L. Maier-Hein, A. M. Franz, T. R. dos Santos, M. Schmidt, M. Fangerau, H.-P. Meinzer, and J. M. Fitzpatrick, “Convergent iterative closest-point algorithm to accommodate anisotropic and inhomogeneous localization error,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 8, pp. 1520–1532, Aug. 2012.
- [71] T. Mallick, P. Das, and A. Majumdar, “Characterizations of noise in Kinect depth images: A review,” *IEEE Sensors Journal*, vol. 14, no. 6, pp. 1731–1740, Jun. 2014.
- [72] F. Menna, F. Remondino, R. Battisti, and E. Nocerino, “Geometric investigation of a gaming active device,” in *Society of Photo-Optical Instrumentation Engineers (SPIE)*, vol. 8085, Jun. 2011, pp. 80 850G–80 850G–15.
- [73] K. Mühlmann, D. Maier, J. Hesser, and R. Männer, “Calculating Dense Disparity Maps from Color Stereo Images, an Efficient Implementation,” *International Journal of Computer Vision*, vol. 47, no. 1-3, pp. 79–88, Apr. 2002.
- [74] R. Mojtahedzadeh, T. Stoyanov, and A. Lilienthal, “Application based 3d sensor evaluation: A case study in 3d object pose estimation for automated unloading

- of containers,” in *European Conference on Mobile Robots (ECMR)*, Sep. 2013, pp. 313–318.
- [75] V. Montazerhodjat, “Photon-limited time of flight depth acquisition: new parametric model and its analysis,” Master of Science, Massachusetts Institute of Technology, Apr. 2013.
- [76] Z. Mor, “Integrated structured-light projector,” U.S. Patent US20 140 376 092 A1, Dec., 2014.
- [77] C. D. Mutto, P. Zanuttigh, and G. M. Cortelazzo, *Time-of-Flight Cameras and Microsoft Kinect™*. Springer Science & Business Media, Mar. 2012.
- [78] A. Myronenko and X. Song, “Point Set Registration: Coherent Point Drift,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2262–2275, Dec. 2010.
- [79] 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 *10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Oct. 2011, pp. 127–136.
- [80] C. Nguyen, S. Izadi, and D. Lovell, “Modeling Kinect sensor noise for improved 3d reconstruction and tracking,” in *2nd International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, Oct. 2012, pp. 524–530.
- [81] M. Oleynik, “Methods and Systems for Food Preparation in a Robotic Cooking Kitchen,” U.S. Patent 20 150 290 795, Oct., 2015.
- [82] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, “Automatic Targetless Extrinsic Calibration of a 3d Lidar and Camera by Maximizing Mutual Information,” in *26th AAAI Conference on Artificial Intelligence*, Jul. 2012, pp. 2053–2059.
- [83] Y. Park, V. Lepetit, and W. Woo, “Texture-less object tracking with online training using an RGB-D camera,” in *10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Oct. 2011, pp. 121–126.
- [84] G. N. Peggs, P. G. Maropoulos, E. B. Hughes, A. B. Forbes, S. Robson, M. Ziebart, and B. Muralikrishnan, “Recent developments in large-scale dimensional metrology,” *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 223, no. 6, pp. 571–595, Jun. 2009.
- [85] B. Pesach, Z. Mor, S. Yalov, and A. Shpunt, “Projectors of Structured Light,” U.S. Patent US20 130 038 881 A1, Feb., 2013.
- [86] B. T. Phong, “Illumination for Computer Generated Pictures,” *Communications of the ACM*, vol. 18, no. 6, pp. 311–317, Jun. 1975.

- [87] D. Redard, “A Performance-estimation-based Tree Algorithm for Reducing Computation in Automatic Target Recognition,” Master of Science, University of Virginia, Charlottesville, VA, USA, Aug. 2008.
- [88] A. Reichinger, “Kinect Pattern Uncovered,” Mar. 2011. [Online]. Available: <https://azttm.wordpress.com/2011/04/03/kinect-pattern-uncovered/>
- [89] S. Rusinkiewicz and M. Levoy, “Efficient variants of the ICP algorithm,” in *3rd International Conference on 3-D Digital Imaging and Modeling*, May 2001, pp. 145–152.
- [90] Y. Salih, A. S. Malik, N. Walter, D. Sidibé, N. Saad, and F. Meriaudeau, “Noise Robustness Analysis of Point Cloud Descriptors,” in *Advanced Concepts for Intelligent Vision Systems*. Springer International Publishing, Oct. 2013, no. 8192, pp. 68–79.
- [91] J. Salvi, S. Fernandez, T. Pribanic, and X. Llado, “A state of the art in structured light patterns for surface profilometry,” *Pattern Recognition*, vol. 43, no. 8, pp. 2666–2680, Aug. 2010.
- [92] H. Sarbolandi, D. Lefloch, and A. Kolb, “Kinect range sensing: Structured-light versus Time-of-Flight Kinect,” *Computer Vision and Image Understanding*, vol. 139, pp. 1–20, Oct. 2015.
- [93] D. Scharstein and R. Szeliski, “A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms,” *International Journal of Computer Vision*, vol. 47, no. 1-3, pp. 7–42, Apr. 2002.
- [94] O. H. Shemesh, R. Quax, B. Miñano, A. G. Hoekstra, and P. M. A. Sloot, “Non-parametric estimation of Fisher information from real data,” *Physical Review E*, vol. 93, no. 2, p. 023301, Feb. 2016.
- [95] A. Shpunt, “Optical designs for zero order reduction,” U.S. Patent US20 090 185 274 A1, Jul., 2009.
- [96] —, “Depth mapping using multi-beam illumination,” U.S. Patent US20 100 020 078 A1, Jan., 2010.
- [97] A. Shpunt and B. Pesach, “Optical pattern projection,” U.S. Patent US20 100 284 082 A1, Nov., 2010.
- [98] J. Smisek, M. Jancosek, and T. Pajdla, “3d with Kinect,” in *13th IEEE International Conference on Computer Vision (ICCV) Workshops*, Nov. 2011, pp. 1154–1160.
- [99] S. Song and J. Xiao, “Sliding Shapes for 3d Object Detection in Depth Images,” in *13th European Conference on Computer Vision (ECCV)*. Springer International Publishing, Sep. 2014, no. 8694, pp. 634–651.

- [100] J. C. Spall, "Monte Carlo Computation of the Fisher Information Matrix in Nonstandard Settings," *Journal of Computational and Graphical Statistics*, vol. 14, no. 4, pp. 889–909, Dec. 2005.
- [101] M. Stein, A. Mezghani, and J. A. Nossek, "A Lower Bound for the Fisher Information Measure," *IEEE Signal Processing Letters*, vol. 21, no. 7, pp. 796–799, Jul. 2014.
- [102] P. B. Sujit and R. Beard, "Multiple UAV path planning using anytime algorithms," in *American Control Conference (ACC)*, Jun. 2009, pp. 2978–2983.
- [103] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer Science & Business Media, Sep. 2010.
- [104] P. Terdiman, "OPCODE," Aug. 2002. [Online]. Available: <http://www.codercorner.com/Opcode.htm>
- [105] I. Tosic and S. Drewes, "Learning joint intensity-depth sparse representations," *IEEE Transactions on Image Processing*, vol. 23, no. 5, pp. 2122–2132, May 2014.
- [106] H. L. V. Trees and K. L. Bell, *Detection Estimation and Modulation Theory, Part I*, 2nd ed. Hoboken, N.J: Wiley, Apr. 2013.
- [107] Y. Tsin and T. Kanade, "A Correlation-Based Approach to Robust Point Set Registration," in *8th European Conference on Computer Vision (ECCV)*. Springer Berlin Heidelberg, May 2004, no. 3023, pp. 558–569.
- [108] T. Varvadoukas, E. Giannakidou, J. V. Gomez, and N. Mavridis, "Indoor Furniture and Room Recognition for a Robot Using Internet-Derived Models and Object Context," in *10th International Conference on Frontiers of Information Technology (FIT)*, Dec. 2012, pp. 122–128.
- [109] I. Wald and V. Havran, "On building fast kd-Trees for Ray Tracing, and on doing that in $O(N \log N)$," in *IEEE Symposium on Interactive Ray Tracing*, Sep. 2006, pp. 61–69.
- [110] C. Wang and H.-W. Shen, "Information Theory in Scientific Visualization," *Entropy*, vol. 13, no. 1, pp. 254–273, Jan. 2011.
- [111] W. Wang and S. S. Iyengar, "Efficient Data Structures for Model-Based 3-D Object Recognition and Localization from Range Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 10, pp. 1035–1045, Oct. 1992.
- [112] M. A. Ward, "Calibrating Lateral-Effect Photodiodes for Use as Measuring Devices in Manufacturing," North Carolina University at Chapel Hill, Department of Computer Science, Tech. Rep., 1990.

- [113] E. Weigl, S. Zambal, M. Stöger, and C. Eitzinger, “Photometric stereo sensor for robot-assisted industrial quality inspection of coated composite material surfaces,” in *International Conference on Quality Control by Artificial Vision*, vol. 9534, Apr. 2015, pp. 95 341D–95 341D–8.
- [114] W. Wohlkinger, A. Aldoma, R. B. Rusu, and M. Vincze, “3dnet: Large-scale object class recognition from CAD models,” in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2012, pp. 5384–5391.
- [115] L. Wu, H.-y. Wang, F.-x. Lu, and P. Jia, “An anytime algorithm based on modified GA for dynamic weapon-target allocation problem,” in *IEEE Congress on Evolutionary Computation (CEC)*, Jun. 2008, pp. 2020–2025.
- [116] J. Xu, N. Xi, C. Zhang, Q. Shi, and J. Gregory, “Real-time 3d shape inspection system of automotive parts based on structured light pattern,” *Optics & Laser Technology*, vol. 43, no. 1, pp. 1–8, Feb. 2011.
- [117] J. Yang, X. Ye, K. Li, C. Hou, and Y. Wang, “Color-guided depth recovery from RGB-D data using an adaptive autoregressive model,” *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3443–3458, Aug. 2014.
- [118] Y. Yu, Y. Song, Y. Zhang, and S. Wen, “A shadow repair approach for Kinect depth maps,” in *11th Asian Conference on Computer Vision (ACCV)*, vol. 4. Springer-Verlag, 2013, pp. 615–626.
- [119] Z. Yun-feng, S. Gan-lin, and J. Bing, “A dynamic-template-library based method to measure the pose of maneuvering target,” in *4th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, May 2009, pp. 2080–2084.
- [120] Z. Zalevsky, A. Shpunt, A. Maizels, and J. Garcia, “Method and system for object reconstruction,” Patent WO2007043036 A1, Apr., 2007.
- [121] Q. Zheng, S. Z. Der, and H. I. Mahmoud, “Model-based target recognition in pulsed ladar imagery,” *IEEE Transactions on Image Processing*, vol. 10, no. 4, pp. 565–572, Apr. 2001.
- [122] X. Zhou, “Statistical Model-based Object Recognition from Three-dimensional Point-cloud Data,” Doctor of Philosophy, University of Virginia, Charlottesville, VA, USA, May 2008.
- [123] X. Zhou and M. DeVore, “Analysis of data and model accuracy requirements for target classification using ladar data,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 4, pp. 1416–1432, Oct. 2008.
- [124] X. Zhou and M. D. DeVore, “Shape recognition from three-dimensional point measurements with range and direction uncertainty,” *Optical Engineering*, vol. 44, no. 12, pp. 127 202–127 202, Dec. 2005.

-
- [125] M. Zia, M. Stark, B. Schiele, and K. Schindler, “Revisiting 3d geometric models for accurate object shape and pose,” in *13th IEEE International Conference on Computer Vision (ICCV) Workshops*, Nov. 2011, pp. 569–576.