3D Imaging for Outdoor Workspace Monitoring

by

Ali Haghighat-Kashani

B.A.Sc., The University of British Columbia, 2006

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

September 2011

Abstract

This research investigates the use of laser scanners to provide fast and accurate workspace information on excavators. The thesis first evaluates the use of laser scanners to track a mining shovel’s dipper position, in order to extract its arm geometry variables in real-time. The low spatial resolution of laser scanners and the need for accurate initialization challenge the reliability and accuracy of laser-based object tracking methods. The first contribution of this work is in introducing a dipper tracking algorithm that provides an accurate, reliable and real-time performance. To accomplish this task, the proposed algorithm uses the shovel dipper’s kinematic constraints and pose history in a Particle Filter to perform an efficient global search in the workspace. The result is then used to initialize an Iterative Closest Point algorithm that uses a geometrical model of the dipper to perform a local search and increase the accuracy of the positioning estimate. In the experiments performed on a mining shovel, the algorithm reliably tracked the dipper in real-time, and obtained a mean dipper positioning error of 6.7cm – 0.3% of the dipper range.

In contrast to arm geometry, most workspace information cannot be extracted using a single laser scanner since it cannot scan the entire scene at a real-time frame-rate. In addition to having a real-time frame-rate, 3D images have to be accurate and reliable to be useful for workspace variable extraction. Hence, this work investigates whether and how such images can be provided using a laser scanner and an intensity camera.

The slow and sparse results of depth sensors (e.g., laser scanner) make them ineffective when used alone. This work proposes a method for fusing depth sensors with an intensity camera to capture accurate and dense 3D images. The proposed fusion uses a super-resolution algorithm based on bilateral filtering to upsample small depth sensor images, with the help of high-resolution images from an intensity camera. The proposed method obtains more accurate results compared to the literature – an error reduction of up to 6.8x based on the Middlebury benchmark – with sharper and more realistic edge definition.

Preface

This thesis presents research performed by Ali Haghighat Kashani.

- The Introduction chapter was written by Ali Haghighat Kashani incorporating suggestions from his supervisors Peter Lawrence and Robert Hall. Parts of the Introduction chapter are excerpts from a conference proceeding co-authored by Ali Haghighat Kashani. The entire conference paper is available in *Appendix A: Rope Shovel Collision Avoidance System,* and it was published in the Proceedings of Applications of Computers and Operations Research in the Mineral Industry (APCOM) Conference, Vancouver, Canada: 2009.

  Ali was primarily responsible for the Arm Geometry section and the 3D Workspace Monitoring section, and collaborated in revising the paper. The version of the paper presented in Appendix A has been slightly modified to make it coherent with the thesis. The changes include the addition of the 3D Workspace Monitoring section.
- Chapter 2 was written by Ali Haghighat Kashani with feedback and suggestions by William Owen, Peter Lawrence and Robert Hall. The research, experiments and data analysis were performed by Ali Haghighat Kashani.

  A version of Chapter 2 has been published in: A.H. Kashani, W.S. Owen, P.D. Lawrence, and R.A. Hall, “Real-Time Robot Joint Variable Extraction from a Laser Scanner,” IEEE International Conference on Automation and Logistics, 2007, pp. 2994-2999. The version presented here has been modified to make the thesis coherent.
- Chapter 3 was written by Ali Haghighat Kashani with feedback and suggestions by Peter Lawrence and Robert Hall. The section on inverse kinematics (Section 3.4) was originally drafted by William Owen. The research, experiments and data analysis were performed by Ali Haghighat Kashani.

A version of Chapter 3 has been published in: A.H. Kashani, W.S. Owen, N. Himmelman, P.D. Lawrence, and R.A. Hall, "Laser Scanner-based End-effector Tracking and Joint Variable Extraction for Heavy Machinery," The International Journal of Robotics Research, 2010. The version presented here has been modified to make the thesis coherent. The changes include modifications to the introduction and conclusions sections, removing subsections that were already described in Chapter 2 (parts of Section 3.2 and Section 3.3.2), adding background information on Kalman filter-based tracking (Section 3.3.4), adding implementation details on Particle Filter (Section 3.3.4) and Distance transformation (Section 3.3.5.1), and adding the inverse kinematics section (Section 3.4) that were not included in the publication.

- Chapter 4 was written by Ali Haghighat Kashani with feedback and suggestions by Peter Lawrence and Robert Hall. The research, experiments and data analysis were performed by Ali Haghighat Kashani.

  A version of Chapter 4 has been submitted for publication: A.H. Kashani, P.D. Lawrence, R.A. Hall, and N.P. Himmelman, "Active and Passive Sensor Fusion Using Bilateral Filtering on Depth Probability," (submitted: Oct, 2010). The version presented here has been modified to make the thesis coherent. The changes include modifications to the introduction and conclusions sections, adding a literature review of laser-camera fusion (Section 4.1), and reporting additional experimental results (Section 4.3.2 and Section 4.4.4).

- The conclusions chapter was written by Ali Haghighat Kashani with feedback and suggestions by Peter Lawrence and Robert Hall.

The research program was identified and initiated by Peter Lawrence, Robert Hall and Tim Salcudean.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ICP	Iterative Closest Point
PF	Particle Filter
BPF	Bootstrap Particle Filter (a.k.a., Condensation Particle Filter)
DT	Distance Transformation
MTBF	Mean Time Between Failures
GPS	Global Positioning System
DH	Denavit-Hartenberg
SDP	Super-resolution on Depth Probability
MRF	Markov Random Field
TOF	Time-of-flight
JBU	Joint Bilateral Upsampling
NAFDU	Noise Aware Filter for Depth Upsampling
GPU	Graphical Processing Unit

Acknowledgements

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under the STPGP 321813-05 grant, and the CGS-D3 award. Special thanks are owed to the employees of Highland Valley Copper for their invaluable assistance.

I would like to thank my supervisors Dr. Peter Lawrence and Dr. Robert Hall for their guidance and encouragement throughout this work. I am extremely grateful for their mentorship, and continuous support of my ambitions. I also thank Dr. William Owen for his help in the first year of my studies, and Drs. John Meech, Shahram Tafazoli, David Lowe, and Septimiu Salcudean for their kind attention and insightful suggestions along the way.

I also like to recognize my friends and colleagues Nicholas Himmelman, Andrew Rutgers, James Borthwick, Mike Lin, and Nima Ziraknejad for the helpful discussions and insights, and for making the years spent in the lab memorable. My gratitude goes to my close friends Mohammadhossein Afrasiabi, Setareh Bateni, Janice Cheam, Pirooz Darabi, Jonathan Hallam, Masih Hanifzadegan, Salim Hassan, and Kiana and Ali Sadeghi Hariri for their unconditional support and encouragement during my years of studies. Thank you for always being there for me!

To my family, I am sincerely grateful for supporting me throughout this process. Eghbal Haghighat Kashani, Hossein, Roya, Maryam and Amir Haghighat Kashani, Dokhi and Ali Bateni, Simin Mashkour, Bita and Armin Aminbakhsh, Nazanin and Amir Bateni, and Moha Bateni, thank you for your care and devotion. You inspired and encouraged me to strive for my Ph.D.

I am indebted to my loyal champions, my grandparents Tayebeh and Abdullah Hashemi, Fakhrosadat Ashrafzadeh, and my deceased grandfather Gholamali Haghighat Kashani for being a source of inspiration and providing me with the loving environment to grow confident and ambitious.

Finally, my parents, Fahimeh and Ahmad, and my sister Salma, have been a constant tower of support – emotional, moral and more. This thesis is dedicated to you as a small token of my gratitude, for your love, your patience, your inspiration, and your guidance at every step of the way, without which none of this would have been possible. I owe my deepest gratitude to you.

To Ahmad, Fahimeh, and Salma

Chapter 1

Introduction[1]

This research investigates the use of laser scanners to provide excavator workspace information. Current algorithms in the literature are discussed, and new ones are proposed, to extract joint-space and workspace information from laser data, as well as to enhance laser scanner images in order to produce suitable images for extracting a broad range of workspace variables. The Introduction chapter presents the motivation and general background for the work.

1.1 Motivation for This Work

The mining industry, such as coal, oil sands, and diamonds, rely heavily on trucks and excavators for material transfer. Rope shovels (Figure 1-1) are used extensively in open pit mining to extract material from the earth and load it into haul trucks. The rate at which they are able to extract and load the material is often the limiting factor in a mine’s throughput, and as such, the shovels need to be run continuously in order to meet production targets. When the bucket is filled, the machine is rotated about a vertical “swing” axis to position the dipper bucket over the truck box and dump the load into it. Unfortunately, the truck loading process is not without risk: the bucket can collide with the haul truck during loading and self-collisions are possible between the dipper and the caterpillar tracks. To avoid collisions with trucks, the relative position of the bucket with respect to the truck must be known. To obtain this, at each point in time, the position and orientation (i.e., *pose*) of the truck with respect to the base coordinate system of the shovel is required, and the values of each of the joint variables of the machine are also required. The above assumes that the kinematic parameters of the machine have been determined through

[1] Parts of this chapter has been published in [1].

documentation or measurement in advance. Although collisions do not typically occur on a daily or even weekly basis, when they do occur they can result in serious injury to the truck driver and expensive downtime and machine repairs for the mine.

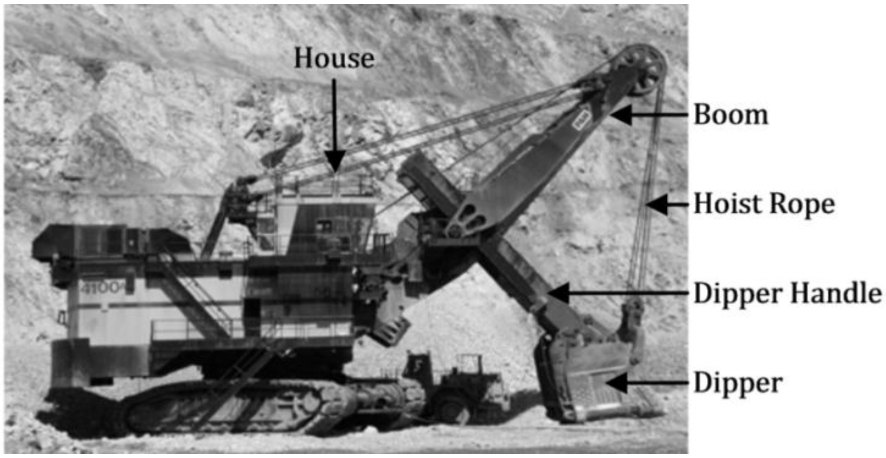


Figure 1-1: P&H Electric Rope Shovel

To maintain competitiveness, larger machines have been introduced to the industry. The introduction of larger trucks and shovels has led to many consequences on the overall mining operation:

- Larger trucks require larger shovels to allow them to be loaded in an economical fashion. Shovels currently exist that can load a 300 ton truck with 3 passes, each pass involving digging or loading the shovel, swinging the load to the truck, dumping the load and returning the empty bucket for the next cycle.
- Compared to smaller excavators, the operator of a large mining shovel is located further away from the end-effector, causing small absolute distances to be more easily misjudged. Moreover, larger machines have higher inertia due to their mass, and therefore their operation is more challenging since it takes them longer to slow down or stop. Thus, the increased size of the shovel introduces potential safety concerns during machine operations. This leads to more contact with the truck (hitting the truck while swinging), and in machines without a modern workspace limiter,

with the shovel itself (hitting and breaking track pads). Further, given the size of the shovel, the consequences of a truck-shovel or shovel-shovel collision are more severe.

The increased size of mining trucks and shovels has given rise to safety issues with the operation of the shovel [2]. Given the size of the shovel, collisions can cause truck driver injury, damage to infrastructure, and excessive wear and tear on the machines [3]. Our survey of the US Department of Labor mine accidents database [4] revealed over 500 recorded truck loading accidents in the past 25 years in the US. The survey reveals that up to 30% of all truck loading accidents can be addressed by an operator assist system that warns the shovel operator of potential collisions and untimely bucket dumping. Therefore, this work is motivated by the desire to introduce either impending collision warnings, or partial automation, by using image-based truck pose and machine geometry sensing in mining operations.

1.2 Background on Excavator Automation

Several studies have examined the automation of excavators of various sizes in the literature [5–10]. Stentz et al. [5] fully automated a hydraulic construction excavator using laser scanners to construct a full environment contour model. Winstanley et al. [6] instrumented a dragline with the goal of assisting operators in moving the bucket through free space with a suitable sensing and control system. Recently, the automation of excavators has turned towards electric rope shovels. Dunbabin and Corke [7] automated a 1:7 scale-model rope shovel and Wallis et al. [8] present a dynamic model of an electric rope shovel with the goal of automating the shovel. Furthermore, a laser scanner-based bucket tracking system for mining excavators was proposed by Duff [11]; however, implementation details are not publicly available. In addition to safety improvement, shovel automation can enhance productivity and lower maintenance cost [3].

Cameras, or intensity cameras (the terms used here to refer to a computer-interfaced video camera), laser scanners and radars have all been used as sensors towards the development of automation on mining excavators [5], [7], [12], [13]. The depth accuracy of a laser scanner is relatively constant with range and it has been shown ([6], [12]) to return useful data in the presence of environmental factors such as rain, dust, shadows and varying ambient light levels which can be more of a challenge for intensity cameras [6]. Moreover, radar can provide reliable long range measurements as well, but it is slow and expensive compared to laser scanners [14], [15]. Thus, this work focuses on the use of a cab-mounted laser scanner or a combination of a laser scanner and an intensity camera, for monitoring the machine geometry and the workspace.

1.3 Background on Laser Scanners

Laser scanners provide information in the form of “point-clouds” – a set of points that lie on the external surface of objects to represent geometry. Each data point represents a vector from the scanner origin (mounted in this case on the cab) to the intersection of the laser beam and the surrounding objects. Hence, the output can be described as a 2D slice of the environment [16]. An example point-cloud is shown in Figure 1-2 where each point in the set is sequentially obtained as the laser beam, scanned by a mirror rotating on a horizontal axis, reflects from points on the boom first, then the stick and then the bucket. The boom, stick and bucket are evident in the point-cloud.

Laser scanners have proven to be robust in outdoor applications [6–8], [13], [16], having a visibility range of up to 150m [13], and an accuracy of a few centimeters over their full operating range [17]. Boehler and Marbs [18] and Stone et al. [19] present detailed analyses of various parameters causing error in laser scanners. The parameters include angular accuracy, surface reflectivity, precipitation and sunlight. Laser scanners such as the one used in this work are shown to have a primarily constant error over their operating range [18]. Note that laser scanner measurement error cannot be represented by a unimodal Gaussian distribution. Gaussian distributions have been used to independently model laser measurement error caused by surface reflectivity, angle of beam incidence or other similar parameters. But the parameters influencing measurement accuracy are too numerous [18], [19] to be able to model laser error accurately using a Gaussian or a similar unimodal distribution. Thus the error distributions reported by laser manufacturers are based on statistical analysis of calibration testing, but they do not address a whole host of issues. For instance, when a laser beam is reflected off more than one object (i.e., the beam splitting problem on object edges), the generated distance value is often significantly erroneous [18], [19]. Such errors are not considered in manufacturer specifications, but can affect application performance.

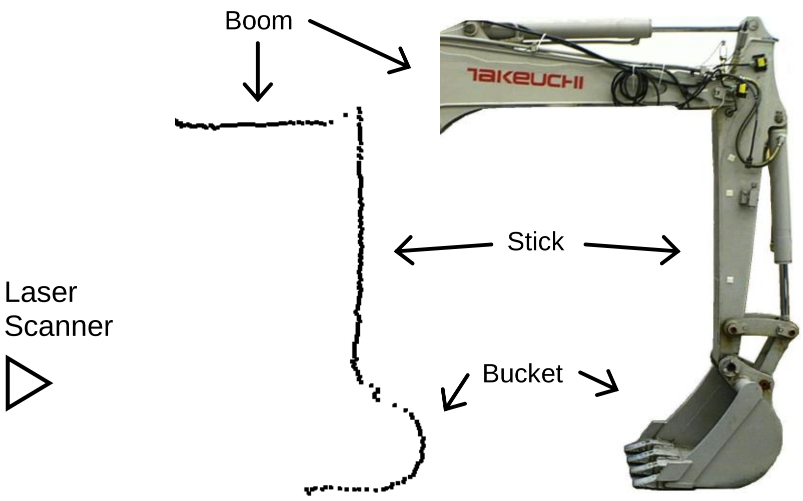


Figure 1-2: Laser Point-Cloud of an Excavator Arm

2D scan planes taken while rotating the plane by a series of incremental angles can be used to construct a full 3D point-cloud of the scene. This can be achieved by rotating or “swinging” the laser scanner about an axis (e.g., the vertical axis in Figure 1-2), to capture different planes of the scene, and then integrating the point-clouds into one complete point-cloud set. Many applications can benefit from 3D point-clouds generated by a swinging laser scanner. Some examples include object detection, 3D video, and robot navigation. To distinguish between 2D and 3D laser imaging in this work, we refer to non-rotating 2D laser scanners as a *planar laser scanner*, and to the rotating one as a *3D laser scanner*.

1.4 Objectives and Approach of This Thesis

Given the motivation of reducing the machine/truck collisions by warning the operator of impending collisions or temporarily automating control of the machine, it was deemed necessary to gain real-time truck pose variables and bucket position variables (which, through inverse kinematics, can be used to obtain arm joint variable estimates). In addition to a predictive warning system, the above information could be used to monitor the dig-face and assist the operator using optimal path planning and shovel base positioning systems. Based on previous work as outlined previously, it appeared that laser scanner could be used alone or in combination with an intensity camera to make measurements of the above variables. This would have the advantage of a measurement system mountable at a single point on the machine that can be retrofitted to a variety of machines (in contrast to fitting sensors at each machine joint).

Thesis Objective: Building upon, and extending existing computer vision techniques in the literature, the objective of this thesis is to evolve technologies suitable for cab-mounted, laser-only measurement of bucket position and joint variables, and for laser-enhanced image camera measurement of the location of workspace objects (e.g. trucks) with adequate accuracy for obstacle avoidance.

Context of the Overall Team Project: The overall design of a rope shovel collision avoidance system is described in Appendix A. It includes systems to measure the dipper pose and shovel arm geometry, the truck pose, and the shovel swing angle. But this thesis only focuses on the use of laser scanners to estimate dipper pose and arm geometry, and monitor the workspace. Figure 1-3 highlights the components of a rope shovel, and identifies the sensor placements for the collision avoidance system proposed in Appendix A. The laser scanner is mounted vertically underneath the boom (L) viewing the dipper. In this position, the laser scanner maintains a consistent view of the dipper and occlusions are minimal.

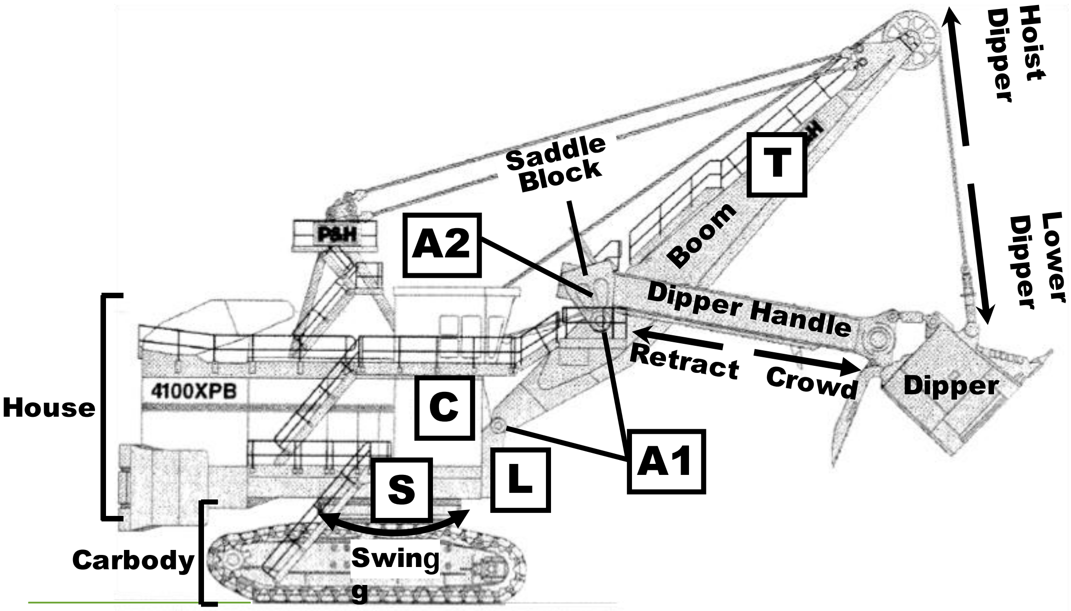


Figure 1-3: Shovel Movements and Hardware Locations

A1) accelerometers placed on the rotary joints, A2) rangefinder placed on the saddle block prismatic joint, C) central computing system located in house underneath operator cab, L) laser scanners placed below boom foot; one planar laser scanner for arm geometry extraction, and one 3D laser scanner along with camera for workspace monitoring, S) stereo camera placed underneath house looking inwards, and T) truck localization stereo camera mounted on boom.

1.4.1 Approach to Laser-Based Joint Variable Extraction

Arm geometry variables consist of the angle of the boom (a revolute joint), the extension of the dipper handle (d, a prismatic joint), and the angle of the dipper handle (θ_2, a revolute joint) as shown in Figure 1-4. The boom angle is set by adjusting the length of the boom suspension ropes whose length is kept constant during operation but can change slightly after the boom has been lowered for maintenance and later raised (joint B in Figure 1-4). The angle at which the dipper is attached to its handle is also adjustable, but stays constant during operation (joint T in Figure 1-4).

During typical operation the arm geometry is controlled by the crowd and hoist functions. The arm geometry can be represented in two ways. The hoist rope length and the dipper handle extension can be used to specify the arm geometry. One difficulty that arises with using the hoist rope length is estimating the stretch in the hoist rope as it depends on the load and arm geometry. Alternatively, the angle of the dipper handle with respect to the boom (θ_2 in Figure 1-4) can be used along with the dipper handle extension (*d* in Figure 1-4) to identify the dipper bucket (also known as the “dipper” or the “bucket”). By using the angle of the dipper handle with respect to the boom the uncertainty associated with stretch in the hoist rope can be avoided.

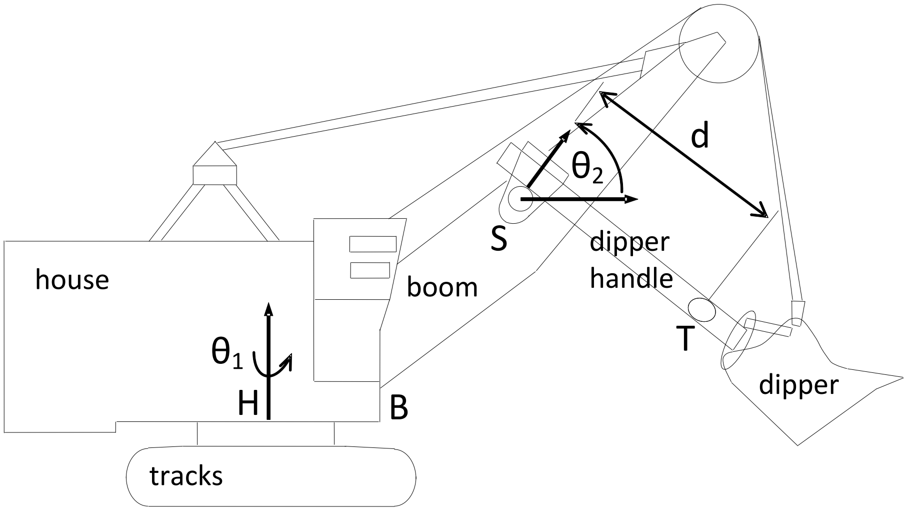


Figure 1-4: Rope Shovel Geometry Variables

The electric rope shovel's swing angle (θ_1), dipper handle angle (θ_2), and dipper handle extension (*d*) are outlined. Variables θ_2 and *d* are referred to in this work as the *arm geometry variables*.

Traditionally, joint angle or extension sensors are used to determine the arm geometry variables. *Forward kinematics* is subsequently applied to the joint variables, which computes the workspace location of the shovel end-effector (the bucket) [20]. This approach may not be practical for many excavators since:

- Certain machines measure some but not all joint variables (e.g., hydraulic excavators measuring actuator extensions, and not joint angles).
- In some excavators, the bucket is not rigidly connected to the body (e.g., Dragline excavators), making forward kinematics inapplicable [21].
- Linkage singularities can cause numerical instabilities in the inverse kinematics solution [20].

- Joint sensors may not be available on the machine, and retrofitting a series of joint sensors can be a difficult task with different challenges on each machine make and model.
- Joint sensors are vulnerable to mechanical damage [6], and maintenance of several joint sensors can be difficult and time consuming.

Overall, custom retrofitting of distal joint sensors to different machines is more costly than mounting a single proximal sensor at the base of the machine to measure all joint variables. As well, repair and maintenance of multiple distal sensors is arguably more challenging than that of a single proximally-mounted sensor, leading to additional on-going costs to the mine operation.

Thus, rather than using joint sensors in the research reported here, a non-contact cab-mounted laser scanner has been used to measure the position of and track the bucket. This is then used to calculate the joint variables. We estimate that to prevent a bucket from hitting the truck, a bucket-positioning safety zone of 30cm would be required (see Section 3.2.1). This is consistent with Brooker et al. [14] who used 1% of the maximum range as the required accuracy. A number of challenges are inherent in a laser scanner-based tracking algorithm. These include laser noise, occlusion of the object of interest, and aliasing of the object's shape appearance due to laser scanner's low spatial sampling. These cause a loss of geometric information (see Section 2.2.1), and since the tracking algorithm relies heavily on geometry to locate the bucket, they undermine the reliability and accuracy of the algorithm. For instance, Duff [11] reports unreliability in tracking an excavator bucket using a laser scanner, primarily due to loss of geometric features. Additionally, while real-time performance is an important requirement for many tracking applications, there is often a trade-off between accuracy and speed. In Chapter 2 and 3 a laser-based bucket tracking algorithm is proposed that addresses the above problems and extracts arm geometry variables useful for collision avoidance.

1.4.2 Approach to Laser-Based Workspace Monitoring

As discussed earlier, collision avoidance also requires the pose of the truck. While Chapter 2 and Chapter 3 provide a solution for extracting arm geometry information using a planar laser scanner, most workspace information including the truck pose cannot be reliably obtained from a planar laser point-cloud. Rather than a planar scanner, a 3D depth sensor is required to provide the necessary information for workspace monitoring. This work aims to propose a single cab-mounted 3D sensing solution that can monitor the workspace information such as truck pose, while also providing imagery for arm geometry variable extraction discussed earlier.

Depth sensors (e.g., laser scanner, stereo-camera, TOF camera) capture *depth images* – an image in which pixel values represent distance to the sensor. A depth image can be transformed into a 3D point-cloud – and vice versa. Accurate and efficient depth sensing can benefit many applications such as robot navigation, scene segmentation, object recognition/tracking, 3D video, and human-computer interaction. In recent years, significant progress has been made on depth sensing using stereo-matching [22], [23]. Stereo algorithms have been improved to better handle image ambiguity [24], and global optimization algorithms with smoothness constraints have been proposed that are computationally expensive but improve accuracy [25–27]. Nevertheless, image ambiguity caused by textureless regions, repeated patterns or occlusions degrades the stereo accuracy and limits its application. Moreover, stereo accuracy degrades quadratically with depth, practically limiting its use to short distances such as indoor environments [28].

Compared to stereo's passive depth sensing (i.e., no light source), *active sensors* that project their own light, such as laser scanners and Time-of-Flight (TOF) cameras, offer a more reliable and accurate alternative. Laser scanners provide accurate long-range results, but are slow for capturing dense real-time data and for handling dynamic scenes compared to stereo cameras (e.g., ~10-100K laser points/sec, versus ~10M for stereo camera). But unlike stereo cameras, laser scanners have a primarily constant depth error at larger distances from the laser origin. TOF cameras are also more accurate than stereo, but are low-resolution, poorly calibrated and noisy, due to the difficulty in modelling their error caused by the complexity of the underlying measuring principle [29]. The accuracy of TOF cameras degrades in dynamic scenes, since faster motion requires faster shutter speed which adds more noise to the data [30].

Towards the goal of shovel workspace monitoring, the final part of this thesis (Chapter 4), explores whether and how the sparse, slow but accurate measurements of a laser scanner (i.e., an active sensor) can be complemented with fast high-resolution images from a single passive intensity camera to generate high-resolution 3D images. This is accomplished through the development of an *active-passive fusion* algorithm to produce high quality images **that cannot be generated using each sensor individually**. These images can be used in future works to extract mining workspace variables, including the future possible estimation of truck pose for obstacle avoidance.

Basic interpolation can be used to upsample low-resolution laser or TOF images, but only smooth surfaces are recovered properly. It fails when dealing with fine details such as sharp edges because there is not enough information [28], [31]. Alternatively, a popular active-passive fusion method is super-resolution: low-resolution laser or TOF images are upsampled with the help of a high-resolution intensity camera image [30–32]. The advantage of super-resolution over basic interpolation is that it is capable of recovering high-frequency details (e.g., edges) using information from the higher-resolution camera image.

Chapter 4 investigates the use of super-resolution for integrating a depth sensor and an intensity camera. An algorithm is proposed that uses *bilateral filtering* to upsample small laser images with the help of a high-resolution camera [32], [33]. The proposed approach builds upon Yang et al. [30], but it applies the bilateral filter to a different formulation of a cost-volume. The novel formulation developed in Chapter 4 is designed to be suitable for sparse and unevenly-distributed depth measurements from active depth sensors such as laser scanners. The comparative evaluation of the proposed algorithm against the literature [30–32], [34–37] demonstrates that the proposed method improves upon the current state of the art. The quantitative comparisons to demonstrate this fact are presented in Chapter 4. Finally, our method is shown to require minimal parameter adjustment across different scenes with varying conditions (e.g., lighting, range, etc.). Chapter 4's focus is on generating high-resolution and accurate 3D images. The algorithms needed to extract workspace information from the 3D images have already been documented in the literature. Thus, they are not addressed in this work.

1.5 Outline and Contributions

1.5.1 Chapter 2: Basic Bucket Tracking on a Mini-Excavator

In the first part of the work, a planar laser scanner is used to develop an effective bucket-tracking method for extracting arm geometry variables. Chapter 2 describes a basic algorithm for bucket tracking of a hydraulic mini-excavator. The proposed algorithm is a variant of the Iterative Closest Point algorithm (ICP) [38]. The algorithm was tested on a hydraulic mini-excavator, which was used as a test-bed for preliminary performance evaluations.

Contributions: The algorithm proposed in Chapter 2 succeeds in tracking the excavator bucket and extracting its arm geometry information. Unlike joint sensors, the proposed non-contact arm geometry extraction system is easily retrofittable to existing machines, is less vulnerable to mechanical damage, requires minimal maintenance, is not prone to numerical instabilities in the kinematics solution, and is capable of handling buckets that are not rigidly connected to the body (see Section 1.4.1). Another practical impact of Chapter 2 is in addressing the laser scanner-based tracking challenges, and obtaining accurate results when faced with occlusion and aliasing.

1.5.2 Chapter 3: A Complete End-Effector Tracking Algorithm on a Rope Shovel

Chapter 3 expands upon the bucket tracking algorithm of Chapter 2, to make it suitable for a mining rope shovel with a larger workspace and a geometrically less distinctive end-effector. The ICP approach used in Chapter 2 is found to fail when used on the rope shovel due to the added complexity, and thus a refined algorithm is developed in Chapter 3 that provides reliable and accurate results when used on the rope shovel. The refined algorithm uses the shovel dipper's kinematic constraints and pose history, in conjunction with the dipper's geometrical model, to track the dipper accurately and reliably.

Contributions: Chapter 3 proposes a novel integration of Particle Filtering (PF) and the ICP algorithm. The PF is used to introduce the kinematic constraints and pose history of the end-effector into the tracking algorithm. The novelty of the proposed integration approach is in using PF to significantly reduce the search space before utilizing ICP to accurately locate the shovel's dipper. Moreover, this work proposes an optimization step in the particle filtering algorithm, by using Distance Transformation to reduce the computational time which achieves real-time performance. Experiments performed on a mining shovel demonstrate the reliability, accuracy, and low computational cost of the proposed algorithm. The practical impact of Chapter 3 is in improving upon the laser-based tracking algorithm of Chapter 2, to better handle aliasing, and reliably track geometrically indistinctive shapes in larger scenes.

1.5.3 Chapter 4: Fusion of Laser Scanner and Intensity Camera for 3D Imaging

While Chapter 3 provides a solution for extracting arm geometry information using a planar laser scanner, most workspace information cannot be obtained from a planar laser point-cloud (e.g., position of obstacles, pose of the truck, surface geometry, etc.). A 3D laser scanner can provide the necessary information for a more comprehensive workspace monitoring, but it is slow in capturing the entire 3D scene and it creates very sparse point-clouds. In recent years, fusion of laser scanners and cameras has shown potential for addressing this problem. Chapter 4 proposes a super-resolution algorithm using a bilateral filter to upsample small 3D laser scanner depth images, with the help of a high-resolution camera image.

Contributions: The proposed method is shown to obtain considerably more accurate 3D image results compared to the literature, with sharper and more realistic edge definition. For instance, the Yang algorithm [30] – among the most accurate upsampling algorithms reported in the literature – shows an average "bad-pixel" error of 7.7% on four benchmark scenes (see [22] for details on the error metric). On the other hand, the method proposed in Chapter 4 obtains an average error of 2.4% in the same experiment. The improved accuracy of the proposed method is shown quantitatively using the Middlebury benchmark, as well as a set of laser-scanned scenes that includes mine site images taken using sensors mounted on a rope shovel.

The potential impact of the Chapter 4 is in providing a 3D imaging system that can capture accurate images for applications such as workspace variable extraction on mining excavators.

Chapter 2

Laser Scanner-Based Bucket Tracking and Joint Variable Extraction on a Mini-Excavator

This chapter focuses on the non-contact arm geometry measurement system described in Chapter 1. This chapter describes a feasibility study of arm joint variables extraction using a laser scanner. For this purpose, a proof-of-concept is deployed on a small excavator, which uses a basic laser-based tracking algorithm to track the bucket, and applies inverse kinematics to the result to obtain arm joint variables. The calculated results are then compared against measurements from a set of joint sensors, to evaluate the accuracy of the system. The system is deemed suitable for further investigation on mining machines if its accuracy is comparable to joint sensors currently used in mines. Note that depending on the application, both the bucket pose and the arm joint variables may be needed. For example, machine dynamics modeling requires knowledge of the joint variables, whereas truck collision avoidance requires knowledge of the bucket pose variables. Therefore, this work estimates the bucket position and orientation using the laser-based system and calculates the joint variables from that.

A hydraulic mini-excavator has been used to validate the proposed algorithm. With a maximum reach of 3.5m, the mini-excavator has a significantly smaller workspace area than a rope shovel that has a reach of over 20m. The mini-excavator is used as a test-bed for the development of a proof-of-concept algorithm and a preliminary evaluation of its performance. In Chapter 3, the approach introduced here is expanded upon and deployed on a rope shovel.

This chapter is based on a paper published in the Proceedings of IEEE Conference on Automation and Logistics [10].

2.1 Introduction

This chapter describes a customized Iterative Closest Point (ICP) algorithm for bucket pose estimation and joint angle evaluation of a hydraulic mini-excavator using a laser scanner. The proposed ICP variant handles point-cloud tracking problems such as noise and occlusion. The novelty of this work is in demonstrating the feasibility of laser scanner-based arm geometry extraction, and the adaptation of the ICP algorithm to laser scanners to track the bucket.

The present algorithm identifies the bucket in the laser scanner reading (*laser point-cloud*) and estimates the pose of the bucket along with the corresponding joint angles. Real-time joint variable extraction of an excavator requires an accurate and time-efficient bucket pose-estimation algorithm.

The method has been tested on a Takeuchi mini-excavator and was demonstrated to be reliable with small workspace error – the Euclidean distance between the estimated bucket position and its actual position. A typical laser scanner point-cloud is provided in Figure 2-1 where the boom, stick and bucket are evident.

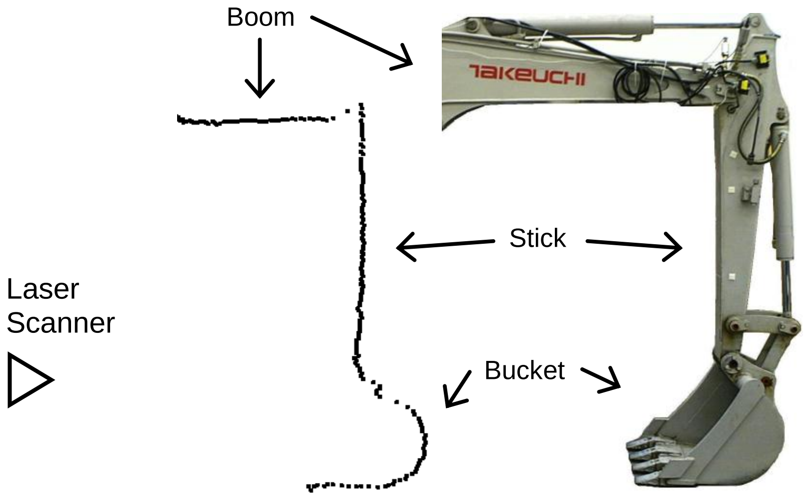


Figure 2-1: Laser Point-Cloud of an Excavator Arm

2.2 Methodology

2.2.1 Pose Estimation in Laser Scanner Point-Clouds

Given a laser point-cloud of an excavator during operation, such as the one shown in Figure 2-1, the task is to detect the bucket in the point-cloud and estimate its position and orientation (i.e., *pose*) with respect to a known point on the excavator body (e.g., the laser sensor location). In order to detect the bucket, a 2D contour model of the bucket is defined and used as a template to find its match within the excavator point-clouds. The contour model (*model point-cloud*) can be manually extracted from a laser point-cloud of the bucket. The defined model is used to find the bucket within the subsequent *laser point-clouds* in real-time. If the bucket is detected within a point-cloud, a transformation matrix can be constructed for describing the position and orientation of the bucket.

Point-cloud registration algorithms are proposed in the literature to determine the displacement between two matching point-clouds, and evaluate the matching quality. In this work, a registration algorithm can be used to find the correct subset of the laser point-cloud that most precisely matches the bucket model. To find the best alignment of the model and the laser point-cloud, a distance measure between the two is defined. Subsequently, the distance value is minimized using various rotations and translations to find the best pose [16], [38].

Various registration methods for identifying an object and its pose have been proposed for laser point-clouds. A few examples are a probabilistic approach [39], Fourier-based registration [40], plane curve matching using lower order derivatives [41] and Iterative Closest Point (ICP) algorithm [38]. Salvi et al. [42] offers a thorough review of point-cloud registration algorithms, and compares them using experimental results. Salvi et al. shows ICP and ICP-variants to have lower registration error in most conditions compared to alternative algorithms [42]. That is why ICP variants have been a dominant method for point-cloud registration [43], and have been considered by many as the benchmark algorithm for such applications in the literature [42–45]. Although there was no prior direct evidence that ICP was the most effective method for bucket localization in this work, due to the above evidence a variant of the ICP algorithm [38] is used for bucket tracking.

The novelty of the present work is in the adaptation of the ICP algorithm to laser scanners to produce robot arm geometry. Producing reliable and accurate tracking results is difficult due to:

- *Laser Noise*: The laser range error results in noisy and erroneous point-clouds (see Section 1.3). This can degrade the results of algorithms that employ tangent lines [46] and derivatives [47]. Thus, the use of noise-sensitive measures such as derivatives is avoided in this work. Figure 2-2 shows a laser point-cloud of a smooth planar surface that appears as a jagged line.

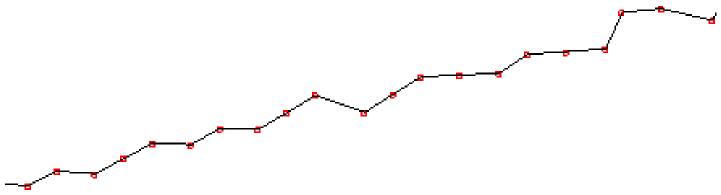
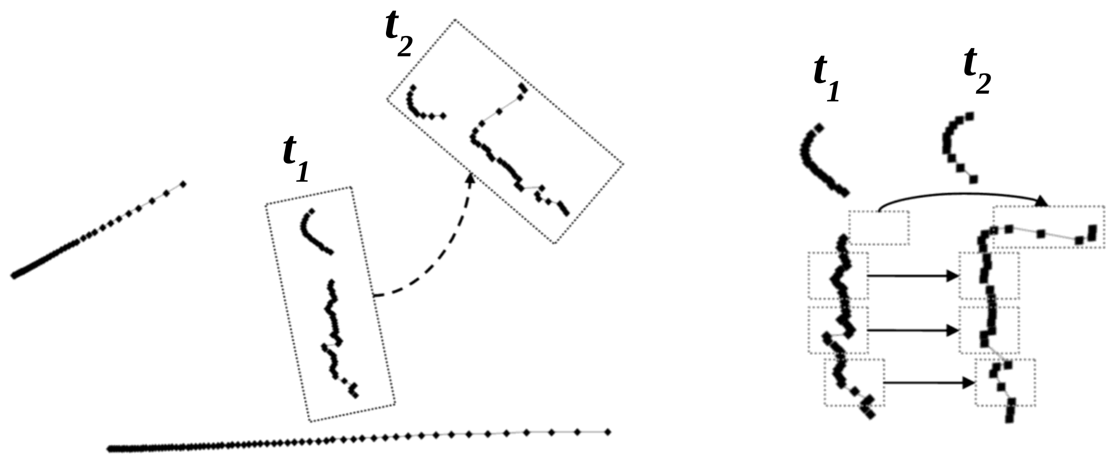


Figure 2-2: 2D Laser Point-Cloud of a Smooth Planar Surface

Captured by a laser scanner, this point-cloud represents a cross-section of a smooth plane. The laser scanner used had a mean systematic error (i.e., measurement bias) of 5cm, with a standard deviation of 1cm.

- *Aliasing*: Due to the laser scanner's low angular resolution causing loss of high-frequency spatial information (i.e., aliasing), a small displacement of an object can cause significant shape variation in its point-clouds (see Figure 2-3). Shape variations introduce challenges to many registration algorithms [48]. This problem is more severe when the object does not have distinctive geometric features to differentiate it from an environment containing similar geometries.

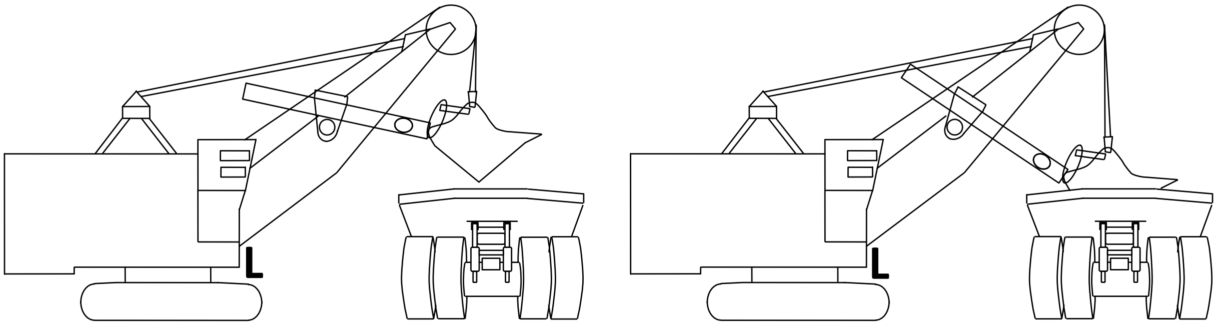


a) b)

Figure 2-3: Variation of Object Shape Appearance in Laser Point-Clouds Due to Aliasing

a) The laser point-cloud of a moving object at times t_1 and t_2. b) The movement of the object causes shape variation in its point-cloud appearance due to aliasing. The changed segments are highlighted.

- *Occlusion*: Surrounding objects can occlude parts of the bucket during operation. For instance, the bucket gets partially occluded by the truck during the dumping stage (see Figure 2-4). Moreover, self-occlusion is also possible when rotation causes a part of the bucket to block other parts. Tracking algorithms relying heavily on geometry are susceptible to error when faced with occlusion.



a) b)

Figure 2-4: Bucket Occlusion During Excavator Operation

a) An electric rope shovel is shown with the bucket positioned above and outside the mining truck. b) The bucket is lowered inside the truck canopy. The laser scanner is located underneath the boom (marked by letter L). The lowering of the bucket causes the laser scanner to lose sight of a portion of the bucket.

- *Partial-overlapping:* If all the points in the model point-cloud have a corresponding point in the laser point-cloud, the two point-clouds fully overlap. However, if the laser point-cloud contains more than just the target object points, the laser point-cloud would only partially overlap with the model point-cloud (see Figure 2-6). Algorithms that are designed for fully-overlapping problems cannot be utilized for partially-overlapping problems [16], [49], [50] as the non-overlapping segments can cause confusion.

2.2.2 Iterative Closest Point Algorithm

Proposed by Besl and McKay [38], the Iterative Closest Point (ICP) algorithm is the benchmark registration algorithm for point-clouds [42], [44]. ICP is a general-purpose, accurate and efficient method that is capable of point-cloud registration with unknown correspondence between each point in one set and a point in the other. ICP is an iterative approach that starts with two point-clouds and an initial estimate of their relative rigid-body transformation [50]. It then refines the transformation by finding point-to-point correspondence between the laser point-cloud and the model point-cloud, and minimizes the distance between the two. A priori correspondence between the laser points and the model points is unknown; hence, depending on the current relative position of the two point-clouds, varying correspondences might be found at each iteration step. Thus, the advantage of using an iterative approach is that the point-to-point correspondence is refined at each step, which improves registration and subsequently generates a better pose estimation [51].

Many variations of ICP have been developed in the past decade to improve its accuracy and computation time complexity. Due to the sparseness of planar laser point-clouds, the computation time complexity is not a concern for bucket pose estimation; thus, in this work, the methods proposed for computation time complexity enhancement are not considered.

The ICP algorithm has been divided into six steps, and ICP variants affect one or multiple of these steps to achieve better results [50]: selection, matching, weighting, rejecting, assigning an error metric, and minimizing the metric, as seen in Figure 2-5. The following sections describe each ICP step.

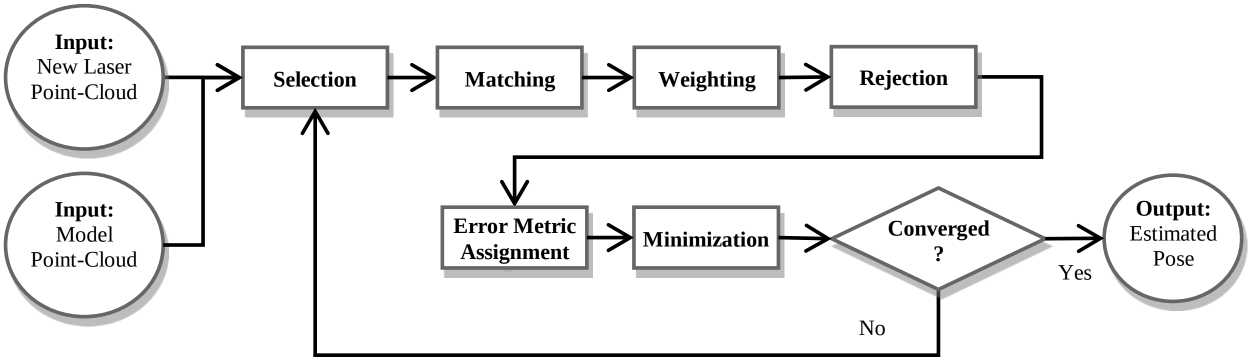


Figure 2-5: ICP Flowchart Diagram

After each ICP iteration, the reduction in the error metric is measured, which represents the magnitude of pose estimation refinement. ICP stops when the refinement is smaller than the desired pose estimation precision, meaning that the required precision has been obtained [38].

2.2.2.1 Selection

The selection step has been introduced in ICP variants in order to obtain better computational cost by reducing the size of the point-clouds. The original ICP [38] does not propose a Selection step. The use of a Selection step is limited to applications with large point-clouds, and therefore it is unnecessary for this work.

2.2.2.2 Matching

Matching is the process of seeking point-to-point correspondence between the point-clouds. For each point in the model point-cloud, the corresponding point within the laser point-cloud is found through the use of closest point method [38]. This is accomplished by matching each point in the model point-cloud with the closest point in the laser point-cloud. The distance between a model point, $\mathbf{m} = (x, y, z)$, and point-cloud P is:

$$d(\boldsymbol{m}, P) = \min_{\forall i} \ d(\boldsymbol{m}, \boldsymbol{p}_i) \qquad (2\text{-}1)$$

where $\boldsymbol{p}_i = (x_i, y_i, z_i)$ is the i^{th} point within *P*, and $d(\boldsymbol{m}, \boldsymbol{p}_i)$ is the Euclidean distance between the two points. The corresponding point within *P* for ***m*** is defined to be the closest point $\boldsymbol{p}_j$ that satisfies the following equality [38]:

$$d(\boldsymbol{m}, P) = d(\boldsymbol{m}, \boldsymbol{p}_j) \qquad (2\text{-}2)$$

Another matching approach uses normal lines [47]; however, Rusinkiewicz and Levoy [50] show that such methods cause instability and do not always converge. Hence, the original closest point matching technique by Besl and McKay [38] is utilized for the bucket pose estimation.

The matching function, C, finds the set of closest points in *P* for every point in the model point-cloud *M*, using Equation (2-2):

$$Y = \mathrm{C}(M, P) \qquad (2\text{-}3)$$

where *Y* is the set of matching points for the model point-cloud.

2.2.2.3 Weighting

Some ICP variants have proposed a Weighting step to model varying uncertainty in the point-clouds. For instance, if a depth sensor’s measurement error increases with distance, the distant points in the point-cloud can be assigned smaller weights. Laser scanners such as the one used in this work have a primarily constant error over their measuring range [18]. Therefore, in the absence of uncertainty variation within the laser scanner point-cloud, the Weighting step was deemed unnecessary for this work. Similarly, the original ICP [38] did not perform Weighting.

2.2.2.4 Rejecting

A pair of point-clouds may not have a one-to-one correspondence, and it is possible to have multiple points in M matched to a single point in P. This is likely to occur in the laser point-clouds due to the sampling variation caused when the laser discretizes continuous object surfaces, with the object moving to different poses. However, the original ICP algorithm assumes that for each point in the model point-cloud, there is a corresponding point in the laser point-cloud. This is an invalid assumption in the majority of applications. Incorrect correspondence degrades the original ICP's performance.

A rejection step is introduced to discard corresponding points that do not actually match. In order to enhance ICP's performance when dealing with laser point-clouds, this work studied various rejection approaches. Corresponding points that do not actually match can be rejected when the distance between them is greater than a threshold [38], [47], [49], [50], by rejecting boundary points [49], or by rejecting 'invisible' points [16]. Based on the performed experiments in this work, the distance-based rejection was found to be unsuitable as it degraded the performance of the bucket-tracking algorithm by rejecting valuable corresponding points.

Turk and Levoy [49] suggest that their boundary correspondence rejection poses advantages by removing points with high risk of matching-error. Boundary points are the laser measurement at or near edges of an object. Boundary points have a high chance of being incorrectly associated with points that have no valid correspondence in the point-cloud due to partial overlapping or occlusion. By excluding all model points that are assigned correspondence with a boundary point in the laser point-cloud, erroneous pairings that cause a systematic bias in ICP's pose estimation can be avoided.

For example, Figure 2-6.a demonstrates the bucket model on the left with its correspondences to the occluded bucket point-cloud on the right. The correspondences for the occluded areas have all been assigned to the boundary points of the occluded point-cloud. In Figure 2-6.b, however, excluding the boundary associations has eliminated the majority of such flawed correspondences.

Lu and Milios [16] offer another useful rejection methodology. After applying the estimated rigid-body transformation on the model point-cloud, the visibility of each point to the laser scanner can be verified. As the object moves, parts of the object become invisible to the laser scanner due to self-occlusion. Thus, the model points associated with the invisible segments will have no correspondence in the laser point-cloud [16]. Taking up the suggestion of Lu et al., it is possible to find out which points in the model point-cloud become self-occluded when the model is located at various poses during the estimation process.

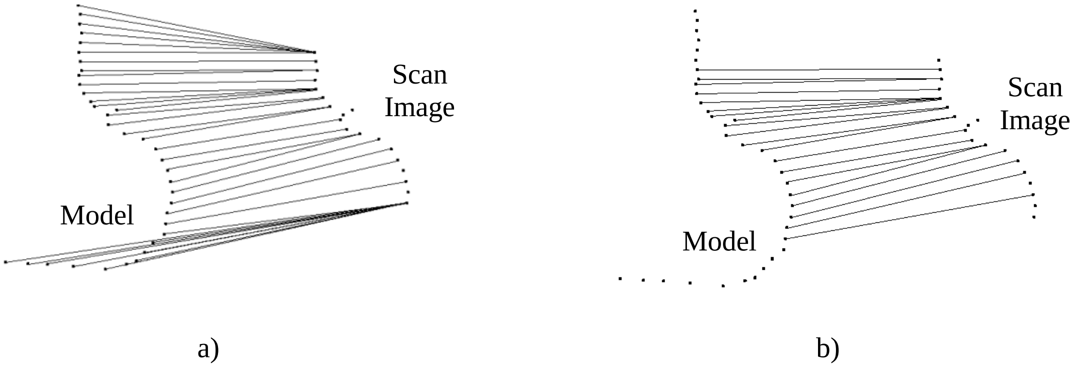


Figure 2-6: Boundary Rejection

a) Partial overlapping between the two bucket point-clouds has caused invalid point-to-point assignments. b) By eliminating the boundary points, the majority of the invalid correspondences are rejected.

The present work proposes an algorithm termed *projection-based rejection* for detecting and rejecting the occluded points, as suggested by Lu and Milios [16]. An ordering constraint is used for identifying the occluded points. The following pseudo-code demonstrates the proposed method:

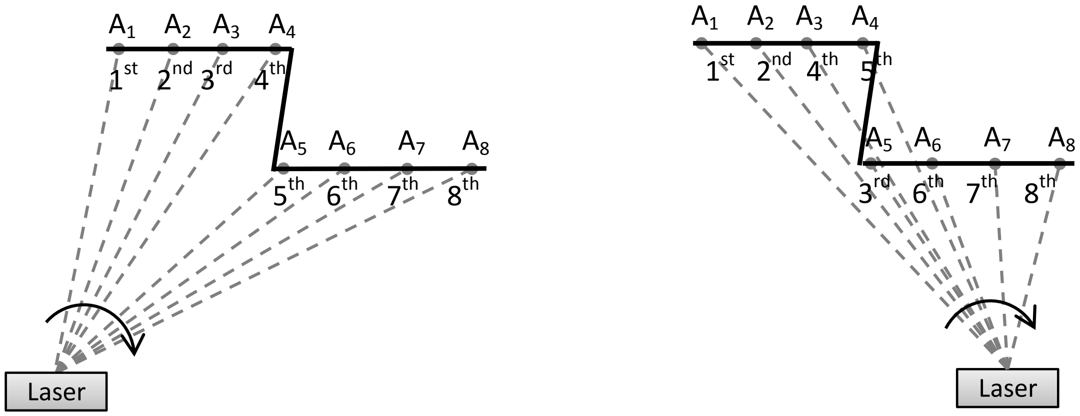
1. Project the model point-cloud M onto a given pose $\boldsymbol{v}$, $M^T = \mathrm{T}(M,\boldsymbol{v})$, using a transformation function T that rotates and translates each point in M to its new position in M^T.

2. For every point $\boldsymbol{m}^T_i$ *in* M^T, calculate θ_i = atan(y_i/x_i) with y_i and x_i being the point's coordinates. This is the angle of the ray from the laser scanner origin, measuring point $\boldsymbol{m}^T_i$.
3. Sort the point set M^T based on θ's: M^P = sort(M^T, θ).
4. Do the following until set O is empty:
    a. Compare M and M^P and find laser points that are "out of order": $O = \{\boldsymbol{m}_i \notin \boldsymbol{m}^P{}_i\}$

    b. In set O, find the farthest laser point – the point with the highest $\sqrt{x_i^2 + y_i^2}$ value.

    c. Remove it from M and M^P.

The iterations stop when the order of the points is identical in both M and M^P. Figure 2-7 and Table 2-1 illustrate this approach. In the example, points A_3 and A_4 would be removed before the iterations stop. The removed points are the ones that are *occluded* to the laser, in the right hand image.



a) **b)**

Figure 2-7: Projection-Based Rejection for Occlusion Detection

An ordering constraint can be used for detecting and rejecting occluded points within a point-cloud. a) Laser point-cloud of object A is shown. The order of appearance of the points is written underneath them (from left the right). b) The object's point-cloud is translated to a new position in the workspace. The laser-measured points appear in a different order (A_3, A_4 and A_5). Note that A_3 and A_4 are occluded.

Table 2-1: The Order of Appearance of Points in Figure 2-7

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
M	1st	2nd	3rd	4th	5th	6th	7th	8th
M^P	1st	2nd	4th	5th	3rd	6th	7th	8th

The order of appearance, from left to right, of the points in Figure 2-7. Points A_3, A_4, and A_5, are highlighted as "out of order," given that their order of appearance changes.

The use of projection-based rejection poses an advantage because it removes laser points that have become invisible and hence are causing invalid correspondence. Figure 2-8 demonstrates projection-based rejection by adjusting the model point-cloud into a point-cloud that is very similar to the laser point-cloud. In the top row, the laser point-clouds are presented and it demonstrates how a segment of the bucket point-cloud becomes invisible after the bucket is displaced. The bottom row shows the original bucket model point-cloud (Figure 2-8.c) and the model after displacement and projection-based rejection (Figure 2-8.d). As shown, the modified model (Figure 2-8.d) looks very similar to the actual laser point-cloud (Figure 2-8.b). The enhanced similarity improves the ICP registration output.

The rejection step in this work includes both the boundary-rejection and the projection-based rejection. The boundary-rejection is an $O(n)$ operation, and the algorithm proposed here for projection-based rejection is an $O(n\ log(n))$ operation. Overall, the rejection step takes a pair of matched point-clouds M and Y, and produces new point-clouds $\overline{M}$ and $\overline{Y}$ in which the rejected points are excluded. The rejection operation in this work is denoted as:

$$\left(\overline{M}, \overline{Y}\right) = \mathrm{R}\left(M, Y\right) \qquad (2\text{-}4)$$

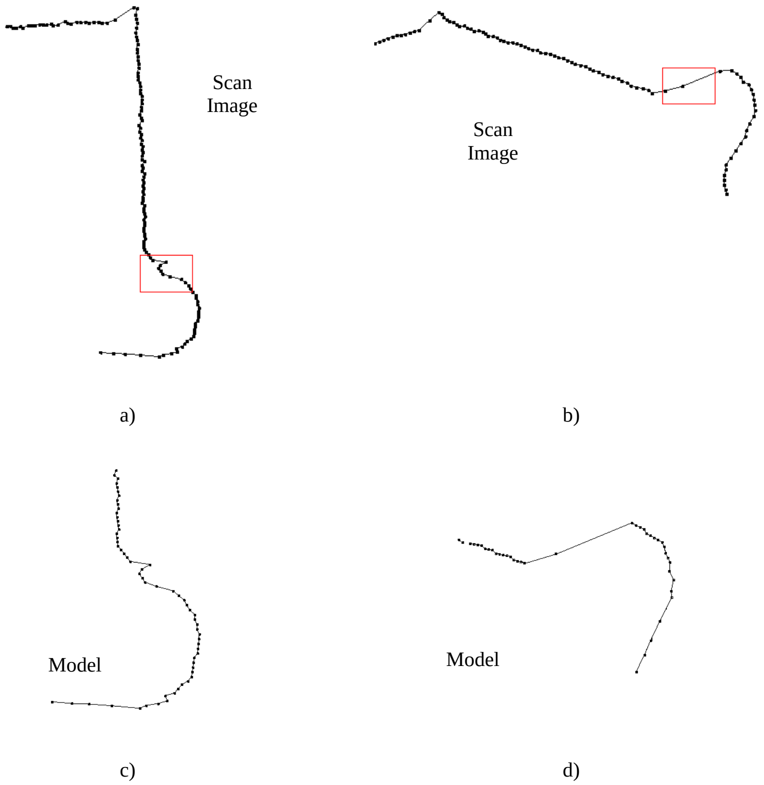


Figure 2-8: Projection-Based Rejection Results on Excavator Point-Clouds

a, b) The laser point-cloud for the excavator arm in two different positions. The geometry of the highlighted area is distorted due to occlusion caused by displacement. c) The original bucket model at its original orientation. d) The bucket model at a different pose, with the invisible points rejected using the projection-based rejection step.

2.2.2.5 Assigning an Error Metric

Besl and McKay's original ICP algorithm calculates the *ICP registration error, e*, by a sum of squared distances between corresponding points in point-clouds $\overline{M}$ and $\overline{Y}$ [38]:

$$e(\overline{M},\overline{Y}) = \sum \| m_i - y_i \|^2 \tag{2-5}$$

where $\| m_i - y_i \|$ is the Euclidean distance between a point within the model point-cloud, m_i, and its corresponding point in the laser point-cloud, p_i. Note that the error metric calculation is performed on the resulting point-clouds from the previous ICP steps, $\overline{M}$ and $\overline{Y}$.

Other error metric assignment methods have been suggested by Zhang [47], and Chen and Medioni [46]. Both methods use derivatives and tangent lines that are unreliable for noisy point-clouds. The *Hausdorff distance metric* can also be used, but its sensitivity to outliers limits its applicability [52] here due to large number of potential outliers. Robust variations of Hausdorff distance have also been proposed [52] which require parameter adjustment in varying scenes. Thus, for this application, ICP's original error metric evaluation was used for the bucket tracking algorithm.

While our ICP implementation uses Equation (2-5) for error metric reporting, for reporting purposes in this chapter we define a more intuitive registration error metric named the average matching distance, e_M:

$$e_M\left(\overline{M}, \overline{Y}\right) = average\ \left(\left\| m_i - y_i \right\|\right) \tag{2-6}$$

We use the above equation to report ICP registration accuracy in the results section, because the average matching distance is more intuitive than a total sum of errors squared.

2.2.2.6 Minimizing

Besl and McKay's [38] quaternion-based algorithm is employed for the minimization step, as described below. Using the distance between corresponding points, a transformation is calculated that guarantees a lower error metric. Given the matching point-clouds, $\overline{M}$ and $\overline{Y}$, the objective function to be minimized is defined as [38]:

$$f = \frac{1}{n}\sum_{i=1}^{n}\left\| y_i - R\left(\overrightarrow{q_R}\right)m_i - \overrightarrow{q_T} \right\|^2 \tag{2-7}$$

where n is the number of points in $\overline{M}$, m_i is the i-th point in $\overline{M}$, y_i is its corresponding point in $\overline{Y}$, and $\overrightarrow{q_R} = (q_0, q_1, q_2, q_3)$ and $\overrightarrow{q_T} = (q_4, q_5, q_6)$ are the vectors for rotation and translation respectively. Function $R\left(\overrightarrow{q_R}\right)$ calculates the following 3x3 rotation matrix based on the quaternion rotation vector:

$$\vec{q}_R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (2\text{-}8)$$

The cross-variance matrix of *Y* and *M* are obtained using:

$$\Sigma_{\mathrm{ym}} = \frac{1}{n}\sum_{i=1}^{n} (y_i - \mu_y)(m_i - \mu_m)^T \quad (2\text{-}9)$$

where *μ* is the “center-of-mass” of the point-cloud:

$$\mu_y = \frac{1}{n}\sum_{i=1}^{n} y_i, \qquad \mu_m = \frac{1}{n}\sum_{i=1}^{n} m_i \quad (2\text{-}10)$$

The eigenvectors of the matrix *Q* are needed for finding the optimal *q* to minimize the cost function *f* [38]:

$$Q = \begin{bmatrix} \Sigma_{\mathrm{ym}}^{\;T} & \Delta^T \\ \Delta & \left(\Sigma_{\mathrm{ym}} + \Sigma_{\mathrm{ym}}^{\;T} - \Sigma_{\mathrm{ym}}^{\;T}\,\boldsymbol{I}\right) \end{bmatrix} \quad (2\text{-}11)$$

where $\Delta = [A_{2,3} \quad A_{3,1} \quad A_{1,2}]^T$, $A_{i,j} = \left(\Sigma_{\mathrm{ym}} - \Sigma_{\mathrm{ym}}^{\;T}\right)_{i,j}$, with $A_{i,j}$ being a matrix element in A, and $\boldsymbol{I}$ being a 3x3 identity matrix.

The eigenvector of *Q* corresponding to its maximum eigenvalue is the optimal quaternion rotation vector $\vec{q}_R$. The optimal translation matrix is then calculated using [38]:

$$\vec{q}_T = \mu_m - R\left(\vec{q}_R\right)\mu_y \quad (2\text{-}12)$$

Based on the $\vec{q}_T$ and $\vec{q}_R$, the registration pose variables $\boldsymbol{v} = (x, y, \theta)$ are $\theta = q_1$, $x = q_4$, $y = q_5$. The above formulation is for the generic three-dimensional search-space (three translation and three rotation variables), whereas due to the planar nature of the laser scanner used in this work, a two-dimensional search-space (two translation and one rotation variable) is sufficient.

The point-cloud $\overline{M}$ is translated and rotated using the resulting pose ***v***, and the ICP registration error is calculated using Equation (2-5) as follows:

$$e\left(\overline{M},\ \overline{Y}\right) = \sum \left\| m_i - y_i \right\|^2 \qquad (2\text{-}13)$$

The quaternion operation discussed in this section is denoted as:

$$\left(\boldsymbol{v}, e\right) = \mathrm{Q}\left(\overline{M}, \overline{Y}\right) \qquad (2\text{-}14)$$

To summarize, the ICP algorithm is performed as follows:

- Initialize the iterations:
    - Set pose variables, $\boldsymbol{v}_0$, to the last known bucket position,
    - Move the model point-cloud by $\boldsymbol{v}_0$ using the transformation function T which performs the rotation and translation operations: $M_0 = \mathrm{T}(M, \boldsymbol{v}_0)$.
- Apply the following steps iteratively – *k* being the iteration variable – until convergence:
    - *Matching Step:* Using the matching function, C, find the set of matching points $Y_k = \mathrm{C}\left(M_{k-1}, P\right)$ between *P* and $M_{\mathrm{k-1}}$.
    - *Rejection Step:* Perform the projection-based rejection and the boundary rejection, $\left(\overline{M}, \overline{Y}\right) = \mathrm{R}\left(M_{k-1}, Y_k\right)$.
    - *Error Metric Assignment, and Minimization Steps:* Compute registration by calculating the pose variables that minimize the error metric: $\left(\boldsymbol{v}_k, e_k\right) = \mathrm{Q}\left(\overline{M}, \overline{Y}\right)$.
    - Move the model point-cloud by $\boldsymbol{v}_{\mathrm{k}}$: $M_{\mathrm{k}} = \mathrm{T}(M, \boldsymbol{v}_{\mathrm{k}})$.
    - Terminate iterations when the change in error metric between consecutive iterations is less than the desired registration precision threshold τ: $|e_{\mathrm{k}} - e_{\mathrm{k-1}}| < \tau$.

The ICP iterations are proven to converge [38], though it is possible, and often likely, for ICP to converge to local minima. The implications of ICP's local minima convergence are discussed in Chapter 3.

2.2.3 Joint Angle Extraction

Once the bucket position and orientation are estimated, inverse kinematics provides the joint angles. Based on the laser data, the bucket position is given by (see Figure 2-9):

$$\boldsymbol{v}_B = \boldsymbol{v}_L + \boldsymbol{R}_L \boldsymbol{v}_B^L \qquad \text{(2-15)}$$

where $\boldsymbol{v}_B$ is the position of the bucket with respect to the excavator (origin O), $\boldsymbol{v}_L$ is the position of the laser scanner based on excavator coordinates, $\boldsymbol{v}_B^L$ is the estimate of the bucket position relative to the laser scanner in the laser's coordinates, and $\boldsymbol{R}_L$ is a rotation matrix from the laser scanner coordinates into the excavator coordinates.

The inverse kinematics for the excavator can be found in [53]. The boom angle is calculated by:

$$\theta_1 = \arccos\left(\frac{a_1^{\,2} + \|\boldsymbol{v}_B\|^2 - a_2^{\,2}}{2 \cdot a_1 \cdot \|\boldsymbol{v}_B\|}\right) - \xi \qquad \text{(2-16)}$$

where θ_1 is the boom angle, a_1 is the boom length, a_2 is the stick length, and ξ is the angle of the vector from the excavator frame to the bucket frame. Additionally, the stick angle is found using:

$$\theta_2 = \pi - \arccos\left(\frac{a_1^{\,2} + a_2^{\,2} - \|\boldsymbol{v}_B\|^2}{2 \cdot a_1 \cdot a_2}\right) \qquad \text{(2-17)}$$

It should be noted that while the range of θ_1 and θ_2 are limited by the mechanics of the machine, the range limitations were not taken into consideration in the present work.

Finally, using the orientation of the model point-cloud at the estimated pose, the excavator's bucket angle can also be determined. Since the primary objective of this work is arm geometry extraction of rope shovels, and given that a rope shovel bucket cannot pivot during operation, this work does not report and evaluate the bucket angle.

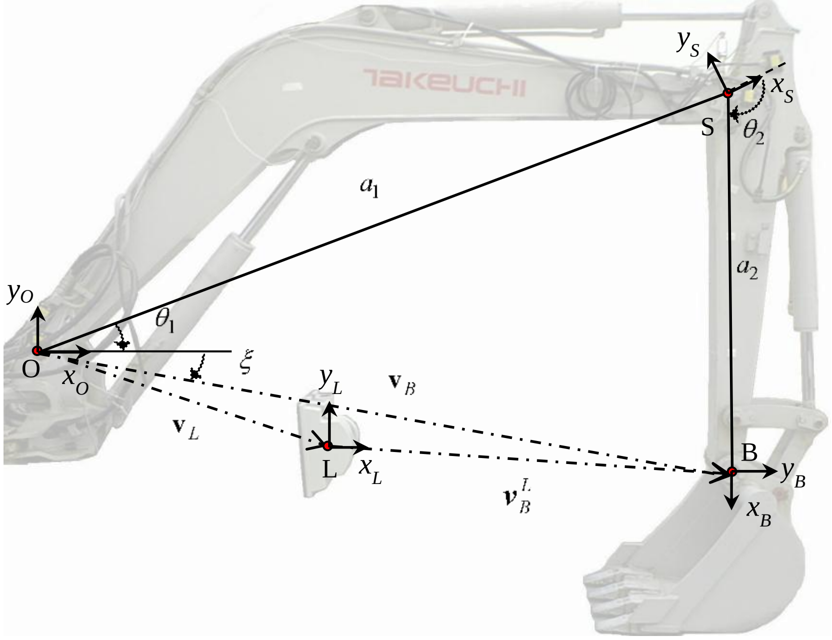


Figure 2-9: Joint Angle Calculations

2.3 Experimental Results

2.3.1 Setup

A SICK LMS221 laser scanner was used for the experiments. The laser's scanning plane is vertical, and it outputs a contour of the bucket using point-clouds consisting of 361 points. The LMS221 laser has a 180-degree field-of-view, a resolution of 1cm, and a mean (μ) accuracy of 5.0cm with a standard deviation (σ) of 1.0cm as reported by the manufacturer. As discussed earlier in Section 1.3, the reported laser accuracy is based on statistical analysis of calibration testing, and while it reports a Gaussian distribution, a unimodal distribution cannot represent all parameters influencing measurement accuracy [18], [19].

Experimental readings were taken on a Takeuchi TB034 mini-excavator. Figure 2-9 shows a side view of the excavator arm and the laser scanner. Over 400 laser point-clouds were captured to examine the reliability of the bucket-pose estimation. Joint-angle measurements were obtained with accelerometer-based sensors [54] in order to compare the laser-based joint angle calculations. The joint angles obtained with the accelerometer-based sensors have a worst-case error of 1.6°.

2.3.2 Obtained ICP Results

The customized ICP algorithm was reliable in all 400 test cases on the mini-excavator, with no reported registration failure – matching the model point-cloud to incorrect points in the laser point-cloud that do not actually belong to the bucket. In case of poor bucket model selection (e.g., very few points, or few distinctive features), it is possible to observe registration misalignment – matching most but not all of the model point-cloud to the points representing the bucket in the laser point-cloud (see Figure 2-11.d). Figure 2-10 plots the error metric of the performed experiments.

Figure 2-11 demonstrates three successful registrations and one case of misalignment due to an indistinctive choice of bucket model. The images on the left show the raw data with the laser point-cloud as small dots and the bucket model as larger dots, at its original position. Right side images show the result of ICP registration in which the model point-cloud has been transformed to the estimated position. The rejected points of the model point-cloud are not displayed; hence, it is possible to observe the strength of the rejection step in making the model point-cloud closely portray the laser observation (i.e., the laser point-cloud).

ICP succeeds in finding the correct bucket position in Figure 2-11.a through Figure 2-11.c with an average matching error, e_M , of 0.67, 1.61 and 1.05 cm, respectively -- see Equation (2-6). Figure 2-11.d shows a misalignment with an error of 4.89cm, caused by a poor choice of model point-cloud.

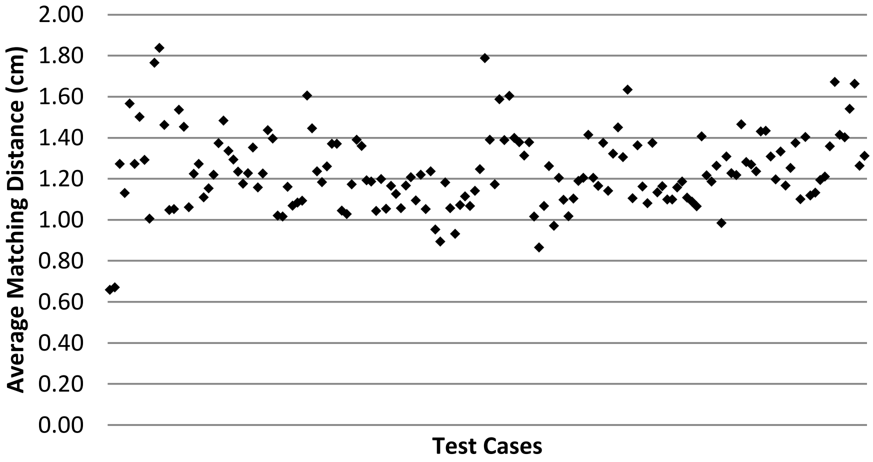


Figure 2-10: The Error Graph for Test Point-Clouds

The average matching error – Equation (2-6) – for the laser point-clouds obtained on a hydraulic mini-excavator.

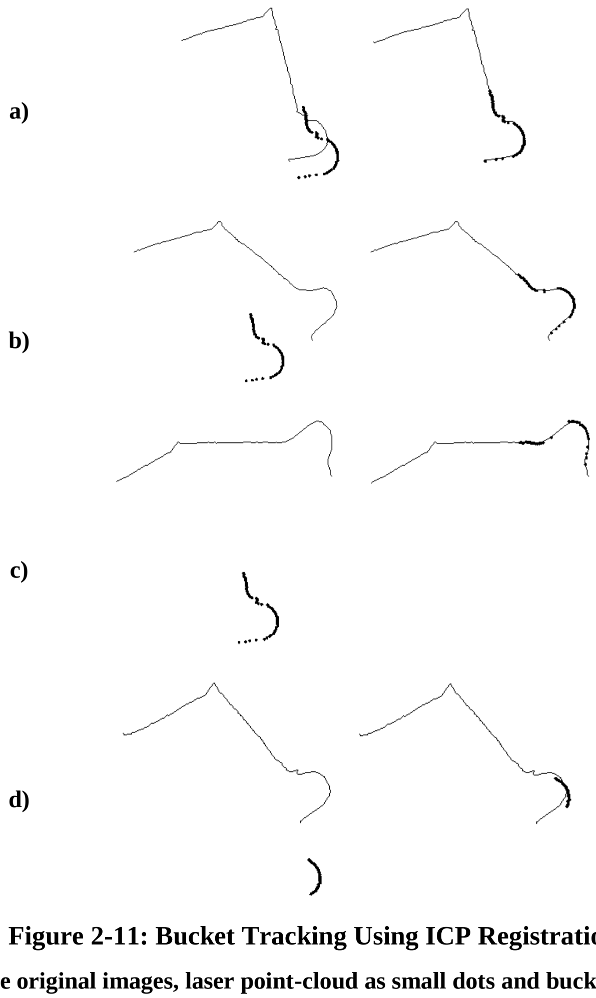


Figure 2-11: Bucket Tracking Using ICP Registration

The images on the left show the original images, laser point-cloud as small dots and bucket model point-cloud as larger dots. Right side images show the ICP outcome. a) through c) show successful registrations and d) shows a misalignment.

In terms of processing requirement, the ICP implementation in this work was performed in real-time and was able to locate the excavator bucket at the laser scanner's frame rate of 40Hz. This is not surprising since all the steps of the ICP algorithm, including matching and rejection, have a computational complexity less than or equal to $O(n \log (n))$, and the size of the laser point-clouds in this work are rather small at n=351 points. On average, ICP required less than 8 iterations to converge within $\tau = 0.1$cm, while the maximum iterations ever needed was 30.

2.3.3 Joint Variable Extraction

After verifying the ICP algorithm, the excavator joint variables are extracted from the pose estimation. The results are compared with the measured angle readings from the joint angle sensors. Figure 2-12 shows the boom angle (θ_1) and stick angle (θ_2) measurement errors. The set of test cases reported here includes both joint angles being positioned at their boundary conditions, as well as the bucket being positioned at the limits of its workspace.

The mean absolute error (μ) is 0.493˚ and 0.608˚ for the boom and stick respectively, while the standard deviation (σ) is 0.295˚ and 0.388˚. The maximum absolute error is 0.926˚ for the boom angle and 1.416˚ for the stick angle. Given that the joint angle sensors used for evaluation have up to 1.6˚ of error, it is not possible to evaluate the contributing factors to the obtained error results

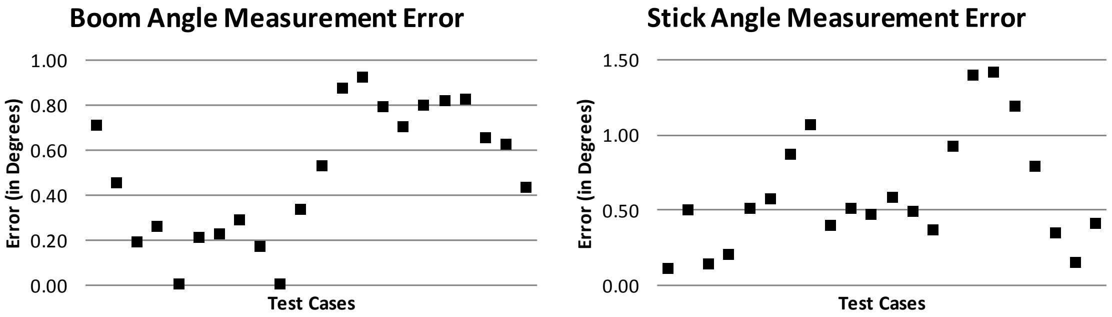


Figure 2-12: Joint Angle Measurement Errors

Estimated errors are reported for the laser-based arm geometry variable estimations. The errors are calculated by taking the absolute difference between the joint sensor value and the estimated angle by the laser-based bucket tracking system.

2.4 Conclusions

A system for hydraulic excavator bucket localization and arm geometry has been developed using a laser scanner. A new variant of the ICP algorithm was applied to determine the location of the bucket in the workspace and the corresponding joint angles were determined. On average the extracted joint angles agreed with the measured joint angles within 0.5° with the worst-case being 1.4°. Since the joint angle sensors were accurate to only 1.6°, one can only say that the worst case error of the proposed system is no greater than 3°. If the average joint angle sensor error[2] is 0 (i.e., no systematic bias), then the average error for our laser-based joint angle estimation would be 0.5°. Thus, the arm geometry extraction system proposed in this work would be a viable replacement for the present joint variables.

In this work, the extrinsic calibration of the laser scanner's coordinate frame – the exact distance vector between L and O in Figure 2-9 – was obtained using manual measurements and with the help of the manufacturer drawings of the laser scanner and the hydraulic excavator. Future works can investigate a general approach for estimating the extrinsic calibration parameters of the laser scanner (e.g., through a series of pre-designed calibration experiments).

Since the inside of the bucket faces the operator, when the bucket is filled, it changes its shape in the laser point-cloud. Therefore, the ICP algorithm proposed here was not capable of tracking the bucket as accurately when it is filled. A possible solution for this is mounting a distinct geometrical "marker" on the excavator arm and using that instead for laser-based end-effector tracking. The mini-excavator experiment in this chapter is mainly a preliminary study on non-contact arm geometry extraction. This work demonstrates some potential for the proposed approach; however, the system must be verified on and adapted to more complex scenes (e.g., larger workspace area, less distinctive end-effector shape).

[2] The actual average error in joint angle sensors is not specified by the manufacturer.

Chapter 3

Laser Scanner-Based Dipper Tracking and Joint Variable Extraction on a Rope Shovel

In this chapter, a novel approach is presented for estimating a mining shovel's dipper pose to obtain its arm geometry in real-time utilizing a planar laser scanner. The low spatial resolution of laser scanners and the need for accurate initialization challenge the reliability and accuracy of laser scanner-based object tracking methods. Experimentation in this work showed the ICP algorithm introduced in Chapter 2 fails to perform reliably when applied to a rope shovel. This is because the rope shovel's end-effector has few geometrically distinctive features, and it is often farther away from the laser scanner thus having relatively fewer laser scanner measurements than the mini-excavator bucket. This chapter addresses these issues by using the shovel dipper's kinematic constraints and position history, in conjunction with the dipper geometrical model, to track the dipper in space. The proposed method uses a Bootstrap Particle Filter (a.k.a., Condensation) with a Distance Transformation in order to perform a *global search* in the workspace. The particle filter's result is then used as the initial pose for a *local search* using an ICP algorithm that increases the accuracy of the pose estimate.

Experiments performed on a mining shovel demonstrate the reliability, accuracy, and low computational cost of the proposed approach. Moreover, using a single cab-mounted non-contact sensor can simplify mounting, reduce maintenance costs and machine down time – which will make it adaptable to various machines – and enhance tracking reliability.

This chapter is based on a published paper[3] [55].

3.1 Introduction

This chapter presents a novel integration of Particle Filtering (PF) and an Iterative Closest Point (ICP) algorithm. To address the reliability challenges of point-cloud tracking, ICP and PF are integrated to limit the tracking search space using complementary information to the geometrical data. The proposed method first utilizes the dipper's kinematic constraints and temporal data (i.e., history of its previous poses) in a particle filter to perform a *global* search to significantly limit the search space. Then, using the dipper's geometrical shape, the ICP algorithm performs a regionally constrained (i.e., *local*) search to locate the dipper accurately. The use of a PF-based global search followed by an ICP-based local search represents the primary contribution of this chapter. In practice, the significance of this approach is that it can consistently track an object with only a small number of range samples and few geometrical features.

ICP provides more accurate estimation results than PF [56], though it is found to occasionally lose track of the dipper due to the local nature of its search. Thus, when used in conjunction with ICP, the PF's global search compensates for ICP's limitations and makes tracking more reliable.

Methods have been proposed to integrate ICP and PF for improved reliability. Some integrations of ICP and PF require parameter adjustments for best performance [44], [57], but the parameters are hard to estimate and vary for different configurations. Instead of integrating ICP and PF, some methods switch between the two based on each algorithm's performance [57]. However, the tracking accuracy of such methods can be inconsistent due to the alternating behaviour. Finally, Sandhu et al. [44] used ICP to weight and update PF particles, which is a different integration method than the work presented here.

[3] A. H. Kashani, W. S. Owen, N. Himmelman, P. D. Lawrence, and R. A. Hall, "Laser Scanner-based End-effector Tracking and Joint Variable Extraction for Heavy Machinery," The Int'l Journal of Robotics Research, Sep. 2010.

Their method results in greater computational cost. The comparison of previous literature to the presented PF and ICP integration method is further discussed in Section 3.3.6.

The second contribution of this chapter is the use of a distance transform to weight the particles during PF, instead of carrying out an exhaustive point correspondence search or a more optimized approach such as kd-trees. This approach enables the operation to be carried out in real-time since the weight of each particle is evaluated in $O(n)$.

The third contribution of this chapter is a comparative evaluation of ICP, PF and an integrated ICP and PF method. The advantages and drawbacks of laser-based tracking using each of the three methods are presented and numerically compared in terms of accuracy and reliability using laser point-clouds. A discussion of the comparisons of the three methods is provided in Section 3.6.1 which demonstrates ICP's lack of reliability in shovel dipper tracking, PF's lack of accuracy, and our integrated method's ability to improve in both of those areas.

The final contribution of this chapter is that a full implementation of an end-effector tracking algorithm has been installed and quantitatively validated on a full-sized rope shovel by comparing it with the direct measurements of joint variables using joint sensors. This work provides a generic solution for retrofitting a single-sensor non-contact arm geometry extraction system that can be utilized on different excavators. The performance results for experiments on a mining shovel demonstrate the ability of the system to meet the speed, accuracy, and tracking-reliability criteria.

3.2 Background

3.2.1 The Use of Laser Scanner on Rope Shovel

A house-mounted laser scanner sensor is proposed here to track the shovel's dipper. To be operational on a rope shovel, the sensor must be extremely reliable with a Mean Time Between Failure (MTBF) greater than 1 year [14]. Moreover, the sensor must be capable of range measurements up to 30m to monitor the dipper in its entire workspace. Finally, the necessary accuracy of the sensor depends on the accuracy needed by the dipper tracking system. In turn, the required dipper tracking accuracy depends on the application at hand and the control requirements. We estimate that to prevent a bucket from hitting the truck, a bucket-positioning safety zone of 30cm would be required. This is similar to Brooker et al. [14] who used 1% of the maximum range as the required accuracy, which in a rope shovel workspace translates into 30cm.

Laser scanners have proven to be robust in outdoor applications [6–8], [13], [16], having a visibility range of up to 150m [13], and an accuracy of a few centimeters over their full operating range [17].

A typical laser scan of the rope shovel is provided in Figure 3-1.b where the shovel boom, dipper, ground, and truck are evident in the point-cloud. To obtain this point-cloud, the laser scanner is mounted underneath the shovel boom and has a vertical scan plane.

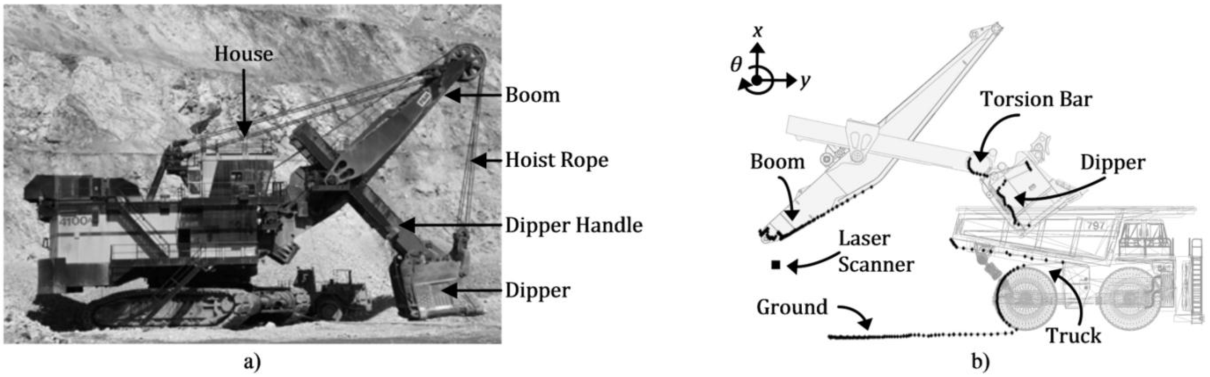


Figure 3-1: P&H Electric Rope Shovel

a) A P&H 2800 Electric Rope Shovel. b) A sample laser point-cloud superimposed on equipment diagrams. The laser scanner is mounted vertically underneath the shovel boom. The laser scanner's point-cloud, shown by a series of dots, represents a contour of the environment. Note that the above point-cloud is not taken during a normal digging cycle. Here, the laser sees the back of the truck, whereas during a normal loading sequence, the truck is located sideways and the laser only sees its side.

The base of the boom is an easy location to access for installation and maintenance of the laser sensor. Dunbabin and Corke [7] installed their laser scanner on the upper portion of the boom. While that enabled them to see inside the bucket, for this application the base of the boom is more appropriate since we are not interested in the bucket content, and the base of the boom is an easier location to access. Moreover, a laser sensor at the top of the boom can find the dipper mostly occluded by the boom when the dipper comes close to the shovel tracks. When the laser sensor is installed underneath the boom, however, a significant portion of the dipper stays constantly visible to the sensor. Constant visibility of the dipper is necessary for a reliable dipper tracking solution. Due to the ease of installation and constant dipper visibility, the base of the boom was used to mount the laser scanner for this work. The risk of installing the laser scanner at the base of the boom, however, is in its vulnerability to falling rocks.

3.2.2 Dipper Pose Estimation in Laser Scanner Point-Clouds

Similar to Chapter 2, given a laser point-cloud of a rope shovel (see Figure 3-1.b), the task is to detect the dipper in the point-cloud and estimate its pose.

While the ICP algorithm performed well in the experiments on a hydraulic mini-excavator (see Chapter 2), dipper tracking on a rope shovel is a more difficult problem. Firstly, the maximum reach of the mini-excavator is 3.5m, whereas a rope shovel can reach over 22.5m, making the dipper tracking search space larger. The larger shovel workspace also means that at farther distances, the shovel dipper is sampled by fewer laser points compared to the excavator bucket in Chapter 2. Finally, the rope shovel dipper is less geometrically distinctive than the excavator bucket; the dipper's discontinuities and edges (i.e. its geometric *features*) often disappear or change form due to the aliasing caused by the laser's low angular resolution. Therefore, the less distinctive dipper in a larger workspace area is more difficult to identify and track with the ICP algorithm.

For a rope shovel, a suitable dipper tracking algorithm must first approximately find which segment of the laser point-cloud corresponds to the dipper – a *global search* – before it can accurately estimate the dipper's pose. This adds complexity to the registration problem. A *coarse* registration algorithm is needed to perform a global search over the entire workspace [42]. Once coarse registration provides a rough estimate of the object pose, *fine* registration can be used to obtain an accurate pose estimate [42]. Fine registration performs a *local search* (similar to ICP) by limiting the search space using an initial estimate of the object pose -- provided by the coarse registration algorithm. If the initial estimate is poor, fine registration can result in erroneous registration. By using coarse registration to estimate the dipper pose within the entire scene, and then using fine registration to improve the accuracy of the estimate, the proposed approach addresses the partial-overlapping problem (see Section 2.2.1).

The dipper tracking algorithm presented here is designed to maintain accuracy and reliability in the dynamic environment of the mine and in the presence of noise, aliasing and occlusion.

3.3 Proposed Method for End-effector Tracking

Dipper tracking requires an accurate pose estimate of the shovel’s dipper from the real-time laser point-clouds. Taken from underneath the shovel boom, a vertical laser point-cloud contains a cross-section of the dipper at its current position in space (see Figure 3-1.b). The task is to detect the dipper within the point-cloud.

As shown in Figure 3-2, cross-sectional point-cloud of the shovel dipper consists of two segments: the upper segment, which represents the *torsion bar* that connects the left and right portions of the dipper handle, and the lower segment which represents the back of the dipper itself. The dipper is attached to the dipper handle at a fixed angle, and thus the torsion bar and the dipper move as a rigid body.

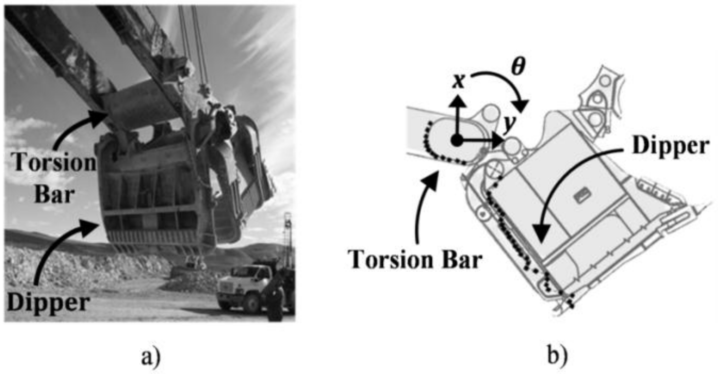


Figure 3-2: Laser Point-Cloud of a Shovel Dipper

a) Shovel's dipper and torsion bar. b) A laser point-cloud of the dipper and torsion bar superimposed on a graphical model.

Dipper pose estimation on 2D laser scanner point-clouds of the torsion bar and dipper is a three-dimensional search for *x*, *y*, and *θ* variables – as illustrated in Figure 3-1.b. This 3D search space correlates to three arm geometry variables: the extension length of the dipper handle (i.e., the crowd length), the length of the hoist rope (i.e., hoist length), as well as the angle at which the dipper is mounted on its handle since the angle varies on different mining shovels.

3.3.1 Point-Cloud Distance Metric

During the course of a search, a metric is required for comparing the poses and finding the best one. A common evaluation metric is the point-cloud distance, $d(P, M)$, between the current laser point-cloud, $P = [\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_m]^T$, and the model point-cloud, $M = [\boldsymbol{m}_1, \boldsymbol{m}_2, ..., \boldsymbol{m}_n]^T$, where $\boldsymbol{m_i}$ and $\boldsymbol{p_j}$ represent points in the laser scanner's 2D coordinates (*x, y*). A common way to find the distance between two point-clouds, $d(P, M)$, is to calculate the average of the Euclidean distances between every point in the model ($\boldsymbol{m_i}$) with a point in the laser point-cloud ($\boldsymbol{p_j}$) closest to $\boldsymbol{m_i}$:

$$d(P, M) = \frac{1}{n}\sum_{i=1}^{n} \min_{\forall j \in P} \left\| \boldsymbol{p}_j - \boldsymbol{m}_i \right\| \tag{3-1}$$

where *n* is the number of points in *M*, and $\left\| \boldsymbol{p}_j - \boldsymbol{m}_i \right\|$ is the Euclidean distance between points $\boldsymbol{p_j}$ and $\boldsymbol{m_i}$.

Once one places the model point-cloud at a pose $s_i = (x_i, y_i, \theta_i)$, its distance to the laser point-cloud is calculated. Among all possible poses, the pose with the closest distance is considered as the dipper pose, s_d:

$$\left\{ s_d \middle| d(P, M^{s_d}) = \min_{\forall k}\big(d(P, M^{s_k})\big) \right\} \qquad (3\text{-}2)$$

where M^{s_k} represents the model point-cloud placed at pose s_k.

Before the point-cloud distance evaluation can be performed in (3-1), a *matching step* is required in which for every point in *M*, the closest point in *P* is found. Using a kd-tree, the matching step is an $O(n \cdot \log\ n)$ operation on average. Since searching requires $d(P, M^{s_k})$ to be calculated for different s_k, matching is performed numerous times for every laser point-cloud. Therefore, calculating $d(P, M)$ can be the most time-consuming component of a registration algorithm. In this work, over a thousand such operations were found to be needed every second. Thus, each $d(P, M)$ calculation must be done in less than 1ms in order to achieve a real-time performance. Section 3.3.5 discusses the computational times achieved by different implementations of $d(P, M)$.

3.3.2 Dipper Tracking using Iterative Closest Point

As shown in Chapter 2, the ICP algorithm can be used for end-effector tracking. Section 2.2.2 describes the ICP variant introduced in this work for end-effector tracking. In Figure 3-3 and Figure 3-5, the proposed ICP method is applied to the laser point-cloud of the rope shovel.

Boundary Rejection [49]***:*** Figure 3-3.a shows a laser point-cloud without the dipper being occluded. Figure 3-3.b is a laser point-cloud obtained after the dipper is lowered into the truck bed, causing a part of the dipper to be occluded by the truck. By comparing consecutive frames, the occlusions can be identified. As shown in Figure 3-3.b, the correspondences for the occluded areas in the laser point-cloud have all been assigned to a *boundary* point – the point adjacent to the discontinuity. Applying the boundary rejection method, introduced in Section 2.2.2, eliminates the incorrect assignments (see Figure 3-3.c).

Projection-based Rejection [16]***:*** Figure 3-4 shows the apparent change in the shape of the dipper as it moves in the workspace. Figure 3-5 shows the result of the projection-based rejection method, introduced in Section 2.2.2, on the rope shovel point-clouds. The modified model point-cloud (B′) looks more similar to the laser point-cloud at time A. The enhanced similarity improves the ICP registration [16].

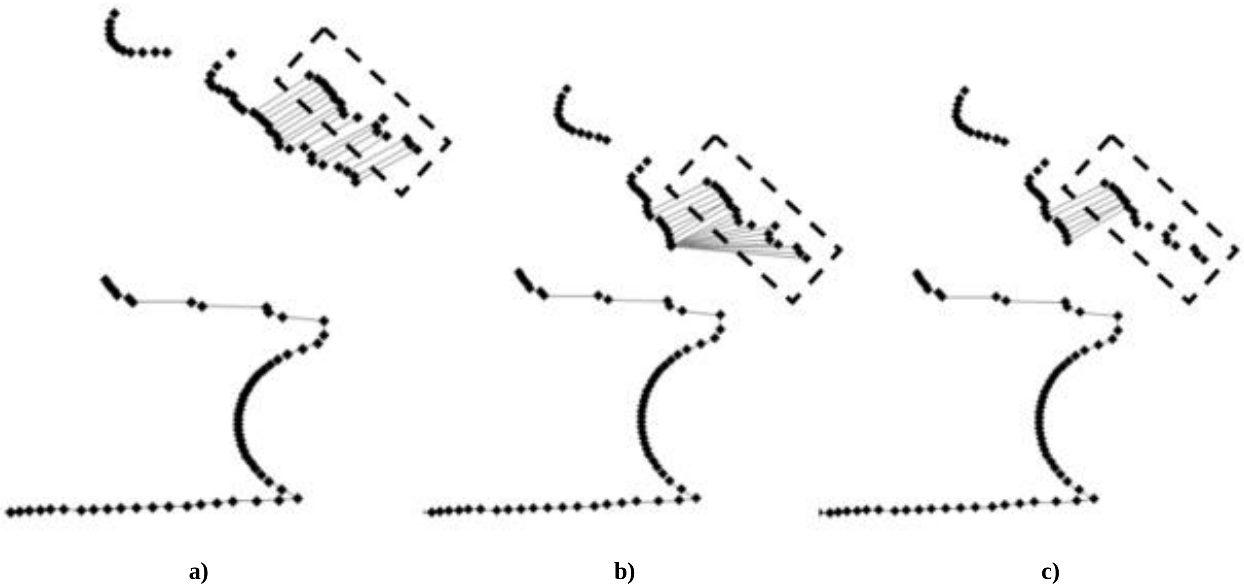


Figure 3-3: Boundary-Rejection on Dipper Point-Clouds for Occlusion Handling

a) Points in the model point-cloud (enclosed in a box) are connected to the points in the laser point-cloud with no occlusion. b) The truck has partially occluded the dipper in the laser point-cloud, which causes incorrect point matching. c) Using Turk's boundary rejection [49], incorrect matches are rejected.

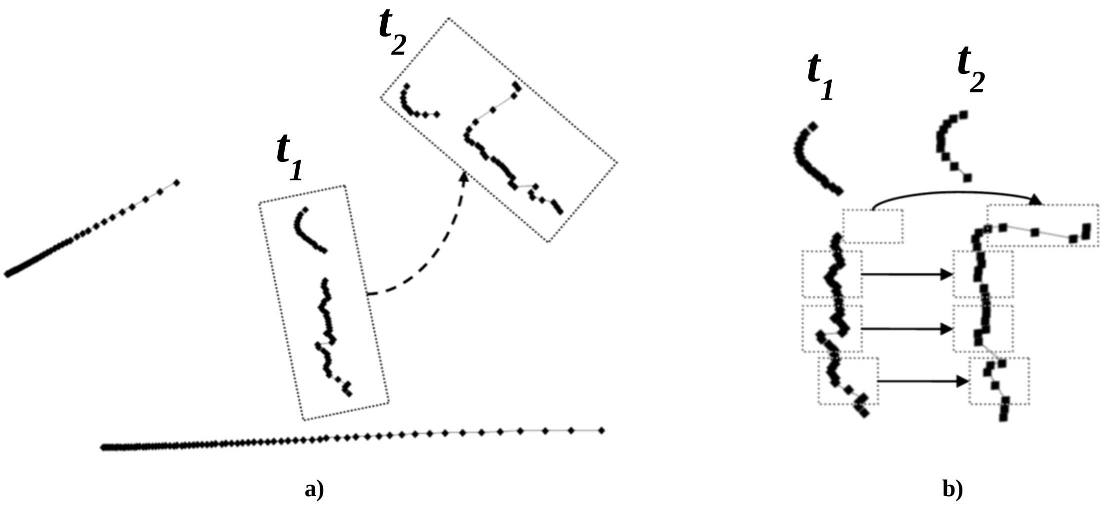


Figure 3-4: Shape Variation of the Shovel Dipper

a) The laser point-cloud of the shovel at times t_1 and t_2. b) The movement of the dipper causes shape variation in its point-cloud appearance. The changed segments are highlighted.

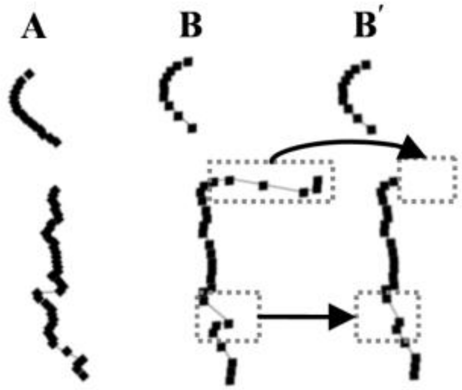


Figure 3-5: Projection-Based Rejection on Dipper Point-Clouds

The above dipper shape variation is due to the dipper movement, as shown in Figure 3-4. Using the projection-based rejection on the model point-cloud (B), the B' point-cloud has been obtained which is considerably more similar to the laser point-cloud of the object (A).

One drawback of the ICP algorithm is that as a fine registration algorithm, it requires a reasonably accurate initial estimate of the pose before calculating a more accurate result. Moreover, ICP is highly susceptible to local minima [44]. In our experiments, the ICP algorithm introduced in Chapter 2 was found not suitable for dipper pose estimation on a mining rope shovel, though it is a critical component of the proposed approach in this chapter.

3.3.3 The Need for a Global Search

A major strength of the ICP algorithm is its low computational cost [57]. ICP can perform point-cloud registration in real-time for sparse point-clouds generated by the laser scanner. Moreover, it normally takes only a few iterations (on the order of 5 to 10) before ICP converges. Using the described rejection step increases the reliability of ICP when dealing with partially overlapping point-clouds.

Due to ICP’s local search nature and its susceptibility to local minima, this work found that ICP’s performance when tracking a rope shovel dipper degrades in two ways: first, tracking can fail if the initial pose estimate given to ICP is not accurate enough, which causes ICP to match the model point-cloud to a wrong section of the laser point-cloud. Second, even when initialized accurately, as the dipper moves in space, the gradual accumulation of error in ICP estimations causes the ICP's estimated dipper pose to gradually ‘slide’ off the actual dipper and start tracking parts of the laser point-cloud that do not actually correspond to the dipper. This happens more often if the model does not contain highly distinctive geometric details to differentiate it from the surroundings, which is the case for the dipper point-clouds. The above two problems are frequently observed when using the ICP algorithm to track rope shovel dipper, which make ICP unreliable when used alone.

Due to the lack of distinctive features in dipper point-clouds and their apparent shape-varying nature illustrated in Figure 3-4, geometric information alone (i.e., the dipper model) is insufficient for reliably locating the dipper in laser point-clouds. Two other types of information can also be used for tracking purposes: the kinematic constraints of the moving object and end-point temporal data. The kinematic constraints are a set of constraints on the velocity and acceleration of the object of interest (i.e., the dipper), and temporal data is pose as a function of time (i.e., the pose at previous instances of time).

To elaborate, a dipper pose estimation s_t at time *t* is valid only when the velocity and acceleration are within the ranges defined by the dipper's kinematic constraints. Calculating the velocity and acceleration requires the dipper pose at previous instances of time, $\{s_0, ..., s_{t-1}\}$ – the temporal data. Utilizing the kinematic constraints and the above temporal data significantly limits the pose estimation search space, leading to better search performance.

The ICP algorithm is not capable of incorporating kinematics and temporal data. Thus in this work, a coarse registration algorithm is used that utilizes the kinematic constraints and the temporal data to address the two problems and make ICP more reliable. Due to its global search nature, the coarse registration method does not gradually lose track of the object due to shape variations, and it does not require an initial pose estimate to locate the dipper.

3.3.4 Global Search using Particle Filtering

Tracking of shape and motion is commonly achieved using Kalman filters [58–63]. While for many applications Kalman filters are efficient and reliable, they are based on linear and unimodal Gaussian processes, which makes them inadequate for applications with non-linear, non-Gaussian or multimodal probability densities [63–67]. For example, laser measurement error cannot be represented by a unimodal Gaussian distribution. Gaussian distributions have been used to independently model laser measurement error caused by surface reflectivity, angle of beam incidence or similar parameters. But the parameters influencing measurement accuracy are too numerous [18], [19] to be able to model laser error accurately using Gaussians or similar unimodal distributions. Error distributions reported by laser manufacturers are based on statistical analysis of calibration testing, but they are often inadequate because they do not address a whole host of issues (see Section 1.3).

Furthermore, at times the probability distribution of the dipper in space can be multimodal – and thus, non-Gaussian – because the global localization problem may require simultaneous modeling of multiple hypotheses. This is because apparent shape-variations in the dipper point-cloud may cause uncertainty in the location of the dipper, resulting in more than one likely dipper pose. Through time, subsequent laser point-clouds are used to evaluate those hypotheses, and eventually the probability distribution narrows down to a single hypothesis. Kalman filters cannot represent alternative hypotheses due to their use of a unimodal Gaussian distribution, and are therefore less suitable for this global localization problem [63–65].

The Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) deal with non-linear processes while still assuming Gaussian noise, and the Gaussian Mixtures method combines a series of Gaussian distributions to model non-Gaussian processes [63], [64]. However, these methods can become computationally expensive particularly in cluttered scenes [63]. Moreover, they continue to have drawbacks since they assume process distributions can be modelled using one or more Gaussians, and the resulting distribution may still not be sufficiently accurate [63].

Isard and Blake [63] demonstrate an experiment in which the unimodal Kalman filter tracker was distracted by scene clutter and lost track of the object of interest, because a false peak produced by the clutter caused the Kalman filter to mistakenly discard the lower peak that in fact represented the object of interest. Isard and Blake [63] then applied a multimodal particle filter tracker to the same problem. The particle filter carried several hypotheses simultaneously and hence recovered rapidly from clutter distractions [63]. Similar results were obtained in [68] where an EKF lost track completely whereas a Particle Filter provided excellent tracking results.

Particle filters (PF) are efficient algorithms commonly used for tracking purposes, because they can appropriately simulate non-Gaussian and multimodal probability distributions associated with sensor observations and object dynamics in tracking processes [63–67]. Particle filters estimate a set of *hidden variables* based on consecutive *observations*. For dipper tracking, the hidden variables are the dipper pose variables, and the observations are the laser point-clouds of the shovel. Using a set of random particles to represent the probability distribution of the hidden variables, a PF appropriately simulates complex distributions [65], [68]. Rather than using analytical functions, the PF's sampling approach ensures its low computational cost and real-time performance [15], [68] – an important requirement for the dipper tracking application. It is shown that if a sufficient number of particles are used, the PF generates an equivalent probability distribution to the analytical representation [68]. Thus, this work applied particle filtering to global localization of the dipper.

A commonly used variation of particle filtering is *Sequential Importance Resampling*, commonly known as the *Bootstrap Particle Filter* (BPF) [68] or the *Condensation Particle Filter* [63], which has been designed for tracking objects moving within cluttered environments based on their contours [41], [51], [63], [67–70]. The BPF utilizes weighted particles. The *i*-th particle of the set at time *t* is denoted by $\left\langle s_t^{(i)}, w_t^{(i)} \right\rangle$, with $s_t^{(i)}$ representing its pose variables from the set s_t at time *t*, and $w_t^{(i)}$ representing its weight. A new laser point-cloud obtained at time *t* is represented by r_t. Based on a new laser point-cloud, the BPF updates each particle of its set using $p\left(r_t \middle| s_t^{(i)}\right)$, the probability of r_t given $s_t^{(i)}$. Once the particle set is updated, the algorithm's resulting estimation of the dipper pose is calculated, which is represented by $\bar{s}_t$ (see Figure 3-6).

The BPF introduces an *importance resampling* step in order to avoid degeneracy: the state in which all but one particle has a weight close to zero [68]. A degenerated particle filter spends most of its processing resources on particles of no value [68]. The importance resampling step helps the particle filter simulate the probability distribution more accurately and more efficiently [68]. The implementation of the Bootstrap Particle Filtering and its resampling step is shown in Figure 3-6. A detailed analysis of BPF can be found in [68]. The following discusses each step of the BPF in the context of dipper pose estimation:

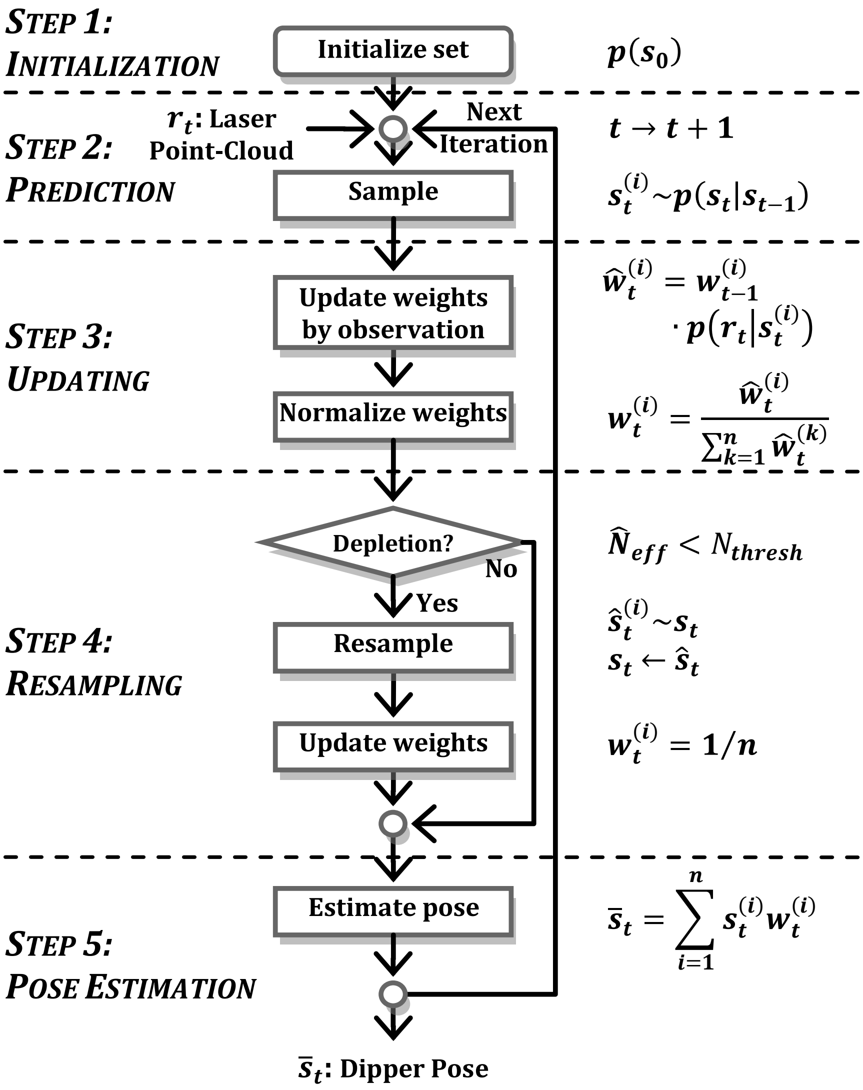

STEP 1: INITIALIZATION
Initialize set
$p(s_0)$
r_t: Laser Point-Cloud
Next Iteration
$t \rightarrow t+1$
STEP 2: PREDICTION
Sample
$s_t^{(i)} \sim p(s_t|s_{t-1})$
STEP 3: UPDATING
Update weights by observation
$\widehat{w}_t^{(i)} = w_{t-1}^{(i)} \cdot p\left(r_t|s_t^{(i)}\right)$
Normalize weights
$w_t^{(i)} = \frac{\widehat{w}_t^{(i)}}{\sum_{k=1}^{n} \widehat{w}_t^{(k)}}$
Depletion?
No
Yes
$\widehat{N}_{eff} < N_{thresh}$
STEP 4: RESAMPLING
Resample
$\hat{s}_t^{(i)} \sim s_t$
$s_t \leftarrow \hat{s}_t$
Update weights
$w_t^{(i)} = 1/n$
STEP 5: POSE ESTIMATION
Estimate pose
$\bar{s}_t = \sum_{i=1}^{n} s_t^{(i)} w_t^{(i)}$
$\bar{s}_t$: Dipper Pose

Figure 3-6: Bootstrap Particle Filter Flowchart

Step 1: Initialization

The first step in the BPF is to initialize the particle set. The particles are drawn from the initial probability distribution of the pose variables, $p(s_0)$:

$$s_0^{(i)} \sim p(s_0) \tag{3-3}$$

where ~ represents the draw operation in which new samples are generated based on the given distribution – $p(s_0)$ in this case. Note that since the initialization step occurs at the system start up, no previous information is available regarding the dipper pose, and therefore $p(s_0)$ is an even probability distribution across the dipper's accessible workspace. Thus, in this work, the sample draw operation during initialization generates a set of particles evenly distributed across the dipper workspace.

Step 2: Prediction

Every time a new point-cloud is available from the laser scanner, a new BPF iteration starts with the prediction step: a new set of particles, $\boldsymbol{s}_t$, are drawn based on the distribution of the previous particle set, $\boldsymbol{s}_{t-1}$. To draw each of the new particles $\boldsymbol{s}_t^{(i)}$ in the prediction step, a particle $\boldsymbol{s}_{t-1}^{(k)}$ is randomly selected from the previous particle set, with the selection chances being proportional to the particle weights in the set $\boldsymbol{s}_{t-1}$. Consequently, using randomization based on the probability distribution of the transition prior $p\left(\boldsymbol{s}_t^{(i)} \middle| \boldsymbol{s}_{t-1}^{(k)}\right)$, the new particle $\boldsymbol{s}_t^{(i)}$ is generated from the old particle $\boldsymbol{s}_{t-1}^{(k)}$. In the case of dipper tracking, the kinematic constraints of the dipper are used to obtain a prediction's probability distribution to be used as the transition prior.

In summary, the prediction step considers the possible previous poses of the dipper – which is represented by the set $\boldsymbol{s}_{t-1}$ – and predicts where the dipper might currently be based on the dipper's kinematic constraints.

Step 3: Updating

The weights of the new particle set needs to be updated based on the recent laser scanner point-cloud, r_t. For any particle, $\boldsymbol{s}_t^{(i)}$, the importance weight is proportional to the image distance between the dipper model at pose $\boldsymbol{s}_t^{(i)}$, and the new laser scanner image:

$$\widehat{w}_t^{(i)} = w_{t-1}^{(i)} p\left(r_t \middle| s_t^{(i)}\right) \qquad (3\text{-}4)$$

where $p\left(r_t \middle| s_t^{(i)}\right)$ is calculated using the point-cloud distance (see Section 3.3.1) between the laser point-cloud r_t and the dipper model point-cloud at pose $\boldsymbol{s}_t^{(i)}$. Subsequently, each particle weight is divided by the sum of all particle weights to normalize the weights (i.e. the sum of all weights adds up to 1):

$$w_t^{(i)} = \frac{\widehat{w}_t^{(i)}}{\sum_{k=1}^{n} \widehat{w}_t^{(k)}} \quad \Rightarrow \sum_{i=1}^{n} w_t^{(i)} = 1 \qquad (3\text{-}5)$$

Step 4: Resampling

A common problem with importance sampling algorithms is the depletion of the particles [68]. Due to the nature of particle filtering, the variance in the particle set increases with time. This makes the particle weights degenerate, causing the situation where the weights of all but one particle approach zero. Other than reducing the PF's accuracy, this problem also degrades efficiency since a considerable amount of time is spent on particles that are practically negligible [68].

The BPF is a *Sampling Importance Resampling* (SIR) algorithm [63]. The SIR algorithms solve the degeneracy problem by performing a Resampling step once the particle set is found to be degenerate. A measure of the degeneracy is the *effective sample size* value defined in [71]:

$$\widehat{N}_{eff} = \frac{1}{\sum_{i=1}^{n}\left(w_t^{(i)}\right)^2} \qquad (3\text{-}6)$$

If $\widehat{N}_{eff}$ is smaller than a defined threshold N_{thresh}, SIR generates a new set of particles. The new particles, $\hat{s}_t$, are sampled proportional to the weights of the current particles, s_t: the particles with smaller weights are removed while the particles with larger weights are multiplied [63]:

$$\hat{s}_t^{(i)} \sim s_t^{(i)}, \qquad s_t^{(i)} \xleftarrow[replace]{} \hat{s}_t^{(i)} \qquad (3\text{-}7)$$

Once the new particle set is generated, the weights must be updated uniformly:

$$w_t^{(i)} = \frac{1}{N} \tag{3-8}$$

Step 5: Pose Estimation

Finally, the PF's resulting estimation of the dipper pose, $\bar{s}_t$, is calculated based on the expectation value of the particle distribution:

$$\bar{s}_t = \sum_{i=1}^{n} s_t^{(i)} w_t^{(i)} \tag{3-9}$$

Particle filters have been used often for tracking purposes on laser scanner data [15], [65], [66] as they can potentially incorporate the observational data (i.e., the dipper's geometric model), temporal data, and the kinematic constraints, all at once. Ma and Ellis [56] used a PF for point-cloud registration in computer-assisted surgery. Yaqub and Katupitiya [65] used a PF and a laser scanner to track robot motion. Sandhu et al. [44] propose a generic pose estimation algorithm that integrates PF and ICP. Finally, Kim and Kim [57] propose a PF and ICP integration to track human body motion. Both Sandhu et al. and Kim et al. demonstrate that the integration of ICP and a PF enhances the reliability of the registration algorithm by consistently producing successful registration [44], [57]. The differences between previous works and the present proposal are described in Section 3.3.6.

Accuracy vs. Speed

For the dipper tracking application, the particle filtering step performs a coarse registration: a global search on the entire workspace to locate the dipper consistently but not very accurately. In terms of localization accuracy, BPF's performance is limited by the size of the particle set, given that it represents the distribution of the pose variables by particles [68]. While a large particle set simulates the probability distribution of the search variables accurately and reliably, it requires more processing power and it can make the particle filter computationally expensive. Since the dipper tracking is a real-time application, the size of the particle set must remain relatively small (few hundred). Thus, the BPF used in this work lacks the accuracy of a fine registration algorithm such as ICP which was discussed in Section 2.2.2.

Adaptivity

An advantage of particle filtering is the possibility to adjust the size of the particle set at run-time. Depending on the accuracy requirement and the processing power available, the size of particles can change from time to time. For example, during the initialization phase where the dipper pose is completely unknown, a larger particle set can be used in order to help the PF converge quickly. Once the dipper pose is found with a high degree of accuracy, the size of the particle set can be reduced to speed up the algorithm. The run-time adjustability makes the PF an adaptive algorithm – it can change its behaviour based on available resources. Such adaptivity makes particle filtering suitable for real-time applications such as dipper tracking.

3.3.5 Integration of the BPF and ICP

In this work, a BPF implementation is proposed that utilizes a Distance Transformation [72] and an ICP algorithm to obtain accurate and reliable tracking results in real-time. Distance Transformation (DT) is utilized in the BPF to reduce computational cost and obtain real-time performance.

3.3.5.1 Updating Particle Weights using Distance Transformation

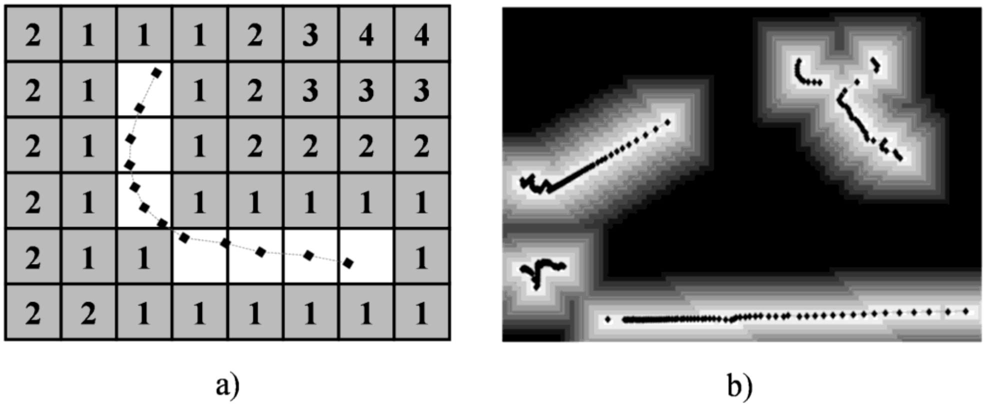
Updating the particle weights after obtaining a new laser point-cloud is an expensive computational step because it requires repeated use of the $d\left(P, M^{s_i}\right)$ operation. To obtain the likelihood probability $p\left(r_t \middle| s_t^{(i)}\right)$, the distance of the laser point-cloud r_t to the dipper model point-cloud at pose $s_t^{(i)}$ can be calculated (see Section 3.3.1 for a discussion on point-cloud distance calculation). The Euclidean distance of the point-clouds has been found to provide a reliable measure for weighting the particles [65]. The most common distance evaluation, such as Equation (3-1) used in ICP, requires matching each point in the model point-cloud to its closest point in the laser point-cloud. As described in Section 3.3.1, a kd-tree can speed up the search by making it an $O\left(n \cdot \log\ n\right)$ operation. Since matching is performed for every particle of the BPF, its repeated use makes it the most time consuming part of the algorithm.

To reduce the processing time, the matching step can be eliminated by using a DT. By calculating the *distance map* of a laser point-cloud, every $d\left(P, M^{s_i}\right)$ calculation can be performed in *O*(*n*) as the matching step will no longer be required. By eliminating the matching step, weighting all BPF particles can be done without significant computational cost. A distance map of the laser point-cloud must be calculated once per laser point-cloud, which can be done efficiently without jeopardizing the system's real-time performance. Therefore, the added efficiency of the proposed BPF makes the real-time performance of the system possible.

3.3.5.2 Distance Transformation and Its Advantages

A distance transform of a point-cloud P, is a two-dimensional image, T. Each pixel in T represents its distance to the closest point in the point-cloud P. Figure 3-7.a shows the DT image, also known as a *distance map,* for a small point-cloud represented by the dots. The number within a cell (i.e. pixel) represents the distance of that pixel to the closest dot, measured in pixels. Note that the white cells have a distance value of zero. Figure 3-7.b illustrates the distance map for an actual laser point-cloud of a rope shovel. The darker cells are further away from the laser points.

In order to perform a DT, the workspace has to be discretized. The size of each distance map pixel (the square cells of Figure 3-7.a defines the accuracy of the DT. High resolution discretization – smaller cell size – results in higher accuracy, while requiring more computation.



2	1	1	1	2	3	4	4
2	1		1	2	3	3	3
2	1		1	2	2	2	2
2	1		1	1	1	1	1
2	1	1					1
2	2	1	1	1	1	1	1

Figure 3-7: Distance Transformation

a) A distance map of a point-cloud object. White cells are the pixels that include a laser point – zero distance – and the numbers within the gray cells represent their distance to the closest laser point. b) A laser point-cloud of the rope shovel super-imposed on its distance map. The darker pixels are further away from the laser points.

Using the algorithm introduced in [48], the DT can be obtained in real-time [72]. Once the distance map of the point-cloud P is acquired, calculating the point-cloud distance between P and M can be done quickly in $O(n)$: First, M is overlaid on the distance map, making each point, $\boldsymbol{m}_i$, fall within a pixel, t_i, in T. Thus, the pixel t_i corresponds to the point $\boldsymbol{m}_i$. Thus, the distance between P and M can be calculated as the average of the pixels in T that correspond to points in M.

$$d(P,M) = \frac{1}{n}\sum_{i=1}^{n} T(\boldsymbol{m}_i) \qquad \text{(3-10)}$$

where the mapping function, $T(\boldsymbol{x})$, returns the pixel, and thus a distance value, in the image T that corresponds to the point $\boldsymbol{x}$.

Based on experimentation with different number of particles, it was found that our particle filter requires at least 100 particles to obtain consistently reliable tracking results. Thus, to update the weight of thousands of particles per second in real-time (40 laser point-clouds/sec * 100 particles), the computational cost of the point-cloud distance calculation method is critical. Since DT performs point-cloud distance calculations in *O*(*n*), it is computationally less expensive compared to the kd-tree's *O*(*n* log *n*) operation for a point-cloud size of *n*=361 generated by the laser scanner used in this work. In our experiments, the kd-tree-based distance calculation for 100 particles required 121.2ms on average, which significantly improved the 2,186ms required by an exhaustive $O(n^2)$ matching implementation for the same number of particles. The DT-based operation suggested in this work further reduces the required computation time to 18.6ms – a constant ~4.8ms required to calculate the distance map, and ~0.138ms to calculate $d(P,M)$ for each of the 100 particles using the distance map. With a target processing frame-rate for a real-time application considered to be greater than 10 frames/sec [19], the DT approach is a feasible solution for obtaining the desired performance, and given the constant overhead in calculating the distance map, it is also a scalable method in case more particles are needed.

It must be noted that a kd-tree would provide a more accurate point-cloud distance calculation than DT, given that DT's workspace discretization affects its accuracy. However, through experimentation, it was found that the particle filter's tracking accuracy when using a DT is very close to that of an accurate distance calculation method like a kd-tree. We believe this is due to the use of ICP to enhance the accuracy of the PF pose estimate, and hence eliminating potential estimation inaccuracy caused by the use DT. Therefore, with similar tracking performance as with a kd-tree, very small memory requirements (KBs), and lower computational cost than a kd-tree, DT was chosen for point-cloud distance evaluation.

3.3.5.3 Refined Pose Estimation using the ICP algorithm

As described earlier, to increase the accuracy of the BPF's coarse registration without using a large particle set, a refinement step is introduced at the end of the BPF's operation. The refinement step uses the highly accurate ICP algorithm to minimize the error through fine registration.

As described in Section 3.3.2, ICP requires a good initial pose estimate since it only performs a local search. If the initial dipper pose is not accurate enough, ICP can fail to find the dipper. The BPF's estimated pose, though, is an adequate initial estimate for ICP. The *ICP-Aided BPF*, thus, enjoys an estimation accuracy similar to that of ICP while compensating for ICP's biggest drawback – the lack of a global search. Figure 3-8 shows the overall system design.

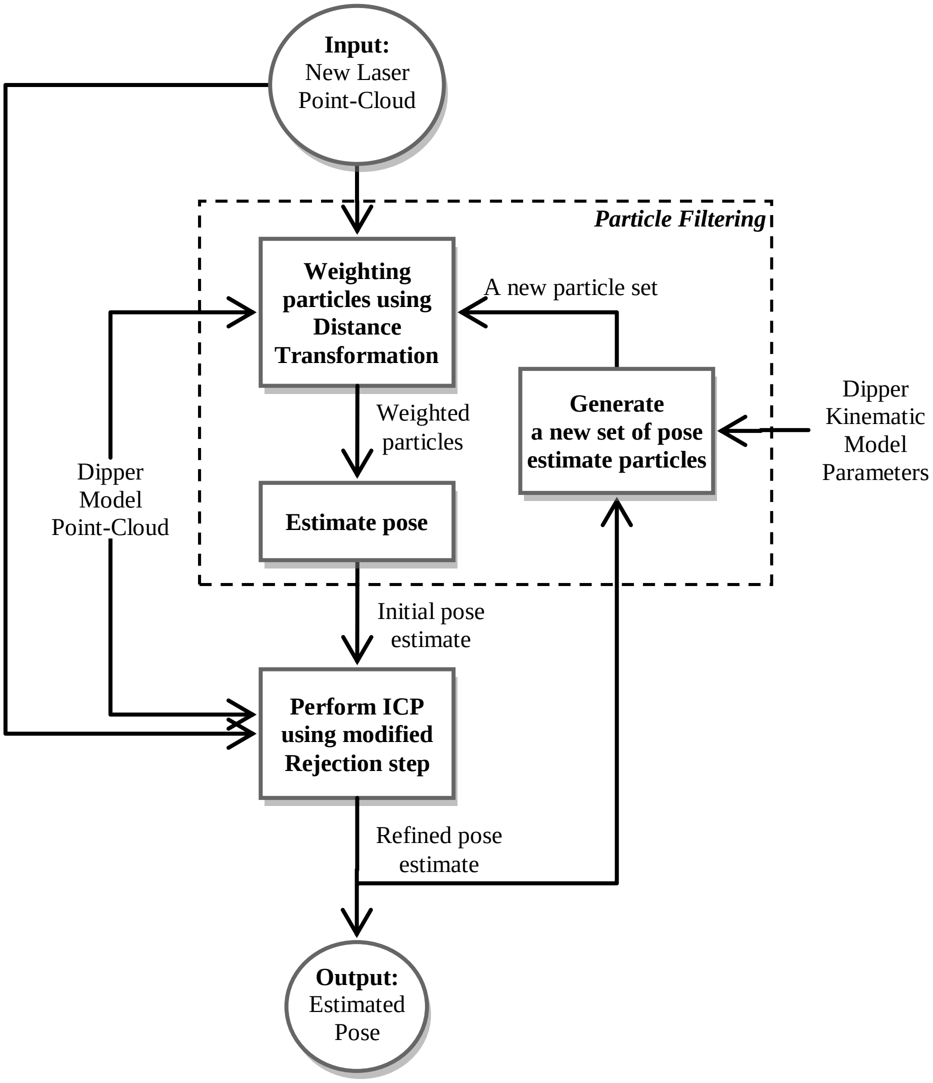


Figure 3-8: The Flowchart of the ICP-Aided BPF

3.3.6 Comparison of ICP-Aided BPF to Previous Work

A number of PF-based point-cloud registration papers were briefly discussed in Section 3.3.4. Having explained the ICP and PF integration proposed in this work, in this section the differences between these algorithms and the proposed method are further discussed.

Yaqub and Katupitiya [65] use scan-matching and a particle filter to track robot motion. Scan-matching algorithms are designed for fully overlapping point-clouds, which makes them unsuitable when dealing with partially overlapping point-clouds in object tracking.

The proposed PF-based range image registration method by Ma and Ellis [56] requires a large set of 5000 particles to achieve reasonable accuracy. This leads to high computational cost [56] and thus lack of real-time performance. The proposed ICP-Aided BPF, on the other hand, uses ICP to perform high-accuracy fine registration. Thus, with a sample size of 100 particles, it can achieve accuracy with low computational cost.

Kim and Kim [57] propose an integration of ICP and PF to track human body motion. The proposed integration is based on a threshold criterion: if ICP's registration error increases beyond a set threshold, PF is used instead to track the target. While better reliability is reported, the registration accuracy is not consistent as the system switches between ICP and PF. Furthermore, this integration approach requires a threshold parameter to be set, which can be hard to estimate and can vary in different configurations. No method was provided for obtaining a desirable threshold.

Finally, the ICP and PF integration algorithm proposed by Sandhu et al. [44] use ICP on each particle to update and weight it. The updating is done by moving each particle according to its ICP pose estimate. Subsequently, the ICP's registration error is used to weight the particle. An important drawback of this method is that it uses ICP on every particle, which can be computationally expensive for real-time applications. Moreover, as pointed out by the authors, the number of ICP iterations on particles has to be carefully adjusted; while too few iterations would not move the particles to regions of interest, too many iterations can move all particles to local minima [44]. The consequences of this fine-tuning problem are twofold. First, finding the appropriate number of iterations can be difficult, and the number can vary in different configurations. Second, since the number of ICP iterations has to be limited, the accuracy of the final pose estimate is limited as well. In contrast, in our work the ICP algorithm runs only once after the particle filter provides an estimate. Hence, high accuracy is obtained with a low computational cost.

3.4 Shovel Arm Geometry Extraction

Arm geometry extraction is achieved by considering the rope shovel to be a robot and by applying inverse kinematics to the reference point on the dipper arm. Once the pose of the dipper is known, appropriate digging and dumping trajectories could be planned and executed using the crowd extension and hoist rope length as control variables.

3.4.1 Reference Frames and Forward Kinematics

Figure 3-9 provides a schematic of the reference frames on the rope shovel. Each frame, F_i, includes x_i, y_i, z_i axes. To prevent the figure from being overcrowded, the third axis for the frames has not been drawn but can be obtained by following the right hand rule for coordinate systems. Notable frames of reference are the base frame, F_b; the laser scanner, F_L; the boom foot-pin, F_p; Joint 1, F_1, which is centered on the shipper shaft; Joint 2, F_2, which is located at the contact point between the shipper shaft and the dipper arm; the torsion bar that is tracked by the laser scanner, F_3; the bail connection where the rope is attached to the dipper, F_{bc}; the dipper teeth, F_t; and the centre of the boom sheave, F_{bs}. The base frame is arbitrarily located at the same elevation as the boom foot-pin frame. Locating the base frame at ground level is not optimal since track wear changes the Denavit-Hartenberg parameters.

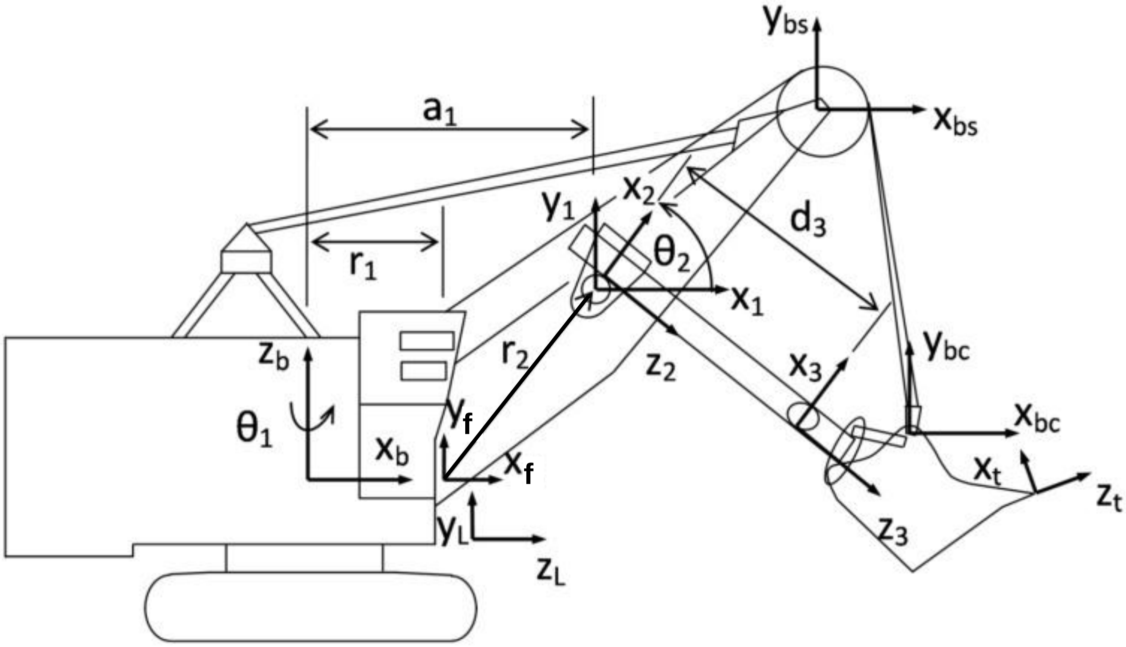


Figure 3-9: Reference Frames on the Rope Shovel

The Denavit-Hartenberg [73] parameters are provided in Table 3-1. Following convention, θ_1 is the rotation of the shovel about z_b; θ_2 is the rotation of Frame 3 with respect to Frame 2 about z_1, i.e., the rotation of the dipper arm about the shipper shaft and is controlled by the hoisting of the cable attached to the bail connection on the dipper; and d_3 is the extension of the dipper arm along z_2. The extension of the dipper arm is determined by two variables: $d_{c,}$ which represents the amount of extension or retraction by crowd motor, and $a_2 \times \theta_2$, which is the change in dipper extension value caused by the rotation of the dipper arm about the shipper shaft. These variables are shown in Figure 3-10.

Table 3-1: DH Parameters for the Rope Shovel

Joint *i*	$\boldsymbol{\theta_i}$	$\boldsymbol{d_i}$	$\boldsymbol{a_i}$	$\boldsymbol{\alpha_i}$
1	θ_1	$r_2 \sin\beta$	$r_1 + r_2 \cos\beta$	$\pi/2$
2	θ_2	0	a_2	$\pi/2$
3	0	$d_c + \theta_2 a_2$	a_3	0

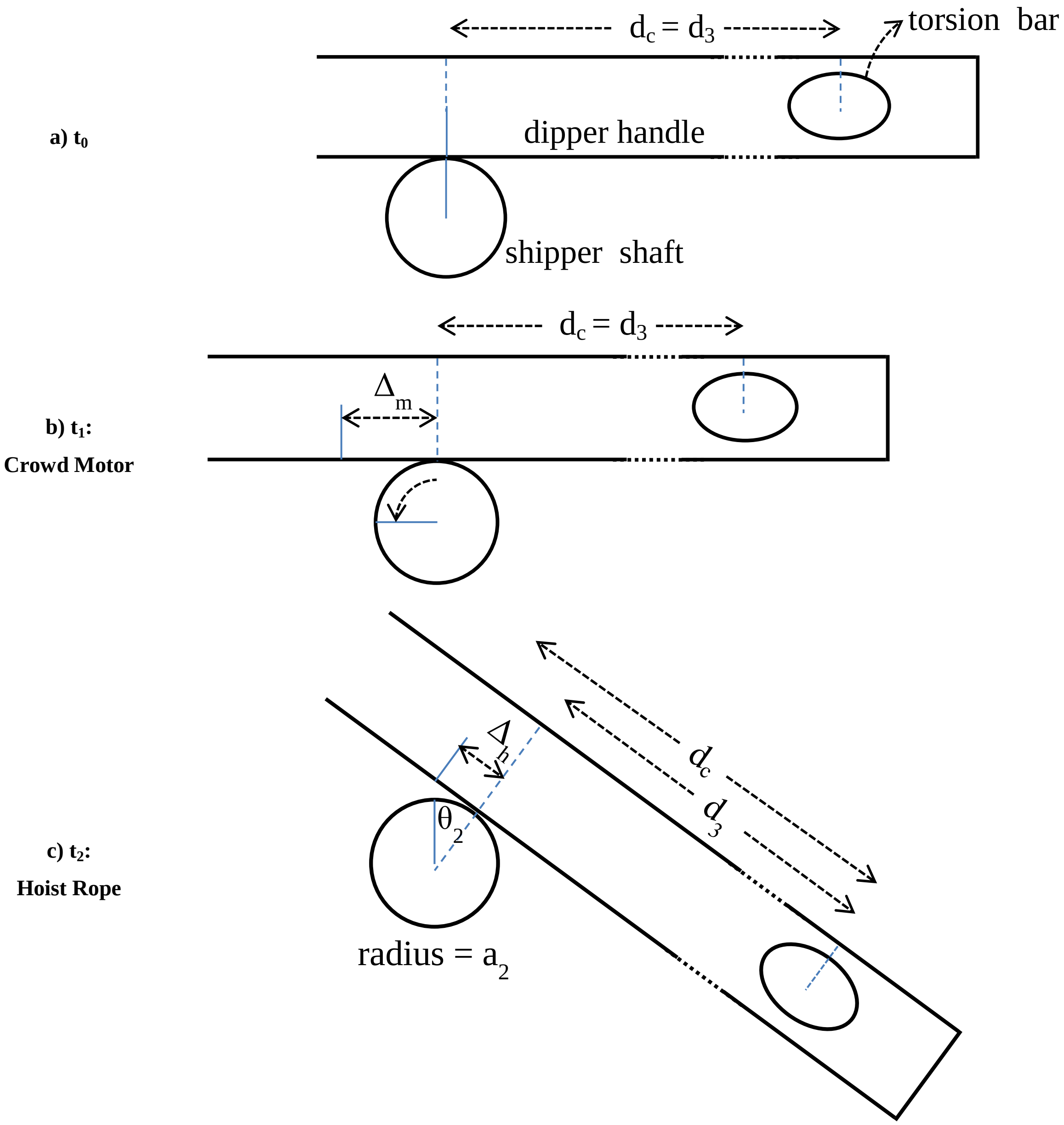


Figure 3-10: Crowd Extension Variables

a) Dipper handle is shown on shipper shaft. The crowd extension, d_3 , changes based on two inputs: crowd motor and hoist rope. b) The dipper handle is shown being retracted by the crowd motor turning the shipper shaft. The retraction by the crowd motor, identified as Δ_m , changes the value of d_c. c) The hoist rope (not shown) changes the angle of the dipper handle. This changes the point-of-contact between dipper handle and shipper shaft, which is the reference point for measuring d_3. Thus hoisting changes the dipper extension by $\Delta_h = a_2 \times \theta_2$. Note that d_c is measured between center of the torsion-bar (on the right) and the point-of-contact between dipper handle and shipper shaft when $\theta_2 = 0$ (on the left).

As shown in Figure 3-9, a_1 is the distance from F_b to F_1 along x_1; r_1 is the distance from F_b to F_p along x_p; and the distance from F_B to F_1 is r_2 and the angle of this vector from the horizontal is β. The radius of the shipper shaft is a_2 (not shown) and F_3 is separated from F_2 by a distance of d_3 along z_2 and offset by a_3 (not shown) along x_3.

Using forward kinematics, the pose of the torsion bar, F_3, with respect to the shipper shaft, F_1 can be determined as,

$$\boldsymbol{r}_{1,3}^{1} = \begin{bmatrix} (a_2 + a_3)\cos\theta_2 + d_3\sin\theta_2 \\ (a_2 + a_3)\sin\theta_2 - d_3\cos\theta_2 \\ 0 \end{bmatrix} \qquad (3\text{-}11)$$

where $\boldsymbol{r}_{i,j}^{k} = \left[x_{i,j}^{k} \; y_{i,j}^{k} \; z_{i,j}^{k}\right]^{T}$ is the vector from Frame i to Frame j referenced from Frame k, where the notation from [74] has been used.

3.4.2 Inverse Kinematics

The laser scanner is mounted under the boom-foot pin. Measurements are taken to determine the location of the torsion bar with respect to the laser, referenced from the laser's frame, F_L,

$$\boldsymbol{r}_{L,3}^{L} = \left[x_{L,3}^{L}, \quad y_{L,3}^{L}, \quad z_{L,3}^{L}\right]^{T} \qquad (3\text{-}12)$$

where $x_{L,3}^{L}$, $y_{L,3}^{L}$, and $z_{L,3}^{L}$, are the measurements obtained from the laser. Since the laser takes planar measurements, $x_{L,3}^{L} = 0$. This vector is transformed into a vector referenced from the shipper shaft, F_1,

$$\boldsymbol{r}_{1,3}^{1} = \left(\boldsymbol{T}_{1}^{L}\right)^{-1} \boldsymbol{r}_{L,3}^{L} \qquad (3\text{-}13)$$

where $\boldsymbol{T}_{1}^{L}$ is the homogeneous transformation matrix of Frame F_1 with respect to F_L,

$$\boldsymbol{T}_1^L = \begin{bmatrix} 0 & 0 & -1 & x_{L,1}^L \\ 0 & 1 & 0 & y_{L,1}^L \\ 1 & 0 & 0 & z_{L,1}^L \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3-14}$$

where $x_{L,1}^L$, $y_{L,1}^L$, and $z_{L,1}^L$ are the distances from F_L to F_1 along the axes of F_L. Based on careful installation, the laser scanner and Frame 1 are aligned, giving $x_{L,1}^L = 0$. The other distance parameters are manually measured during installation. Carrying out the transformation from Equation (3-13) yields,

$$\boldsymbol{r}_{1,3}^1 = \left[z_{L,3}^L - z_{L,1}^L, \quad y_{L,3}^L - y_{L,1}^L, \quad 0\right]^T \tag{3-15}$$

Once the vector $\boldsymbol{r}_{1,3}^1$ is known, inverse kinematics can be used to determine the joint angles, θ_2 and d_3. Squaring and adding the first two terms of Equation (3-11) and Equation (3-15) , setting them equal to each other, and solving for d_3 yields,

$$d_3 = \sqrt{\left(z_{L,3}^L - z_{L,1}^L\right)^2 + \left(y_{L,1}^L - y_{L,3}^L\right)^2 - (a_2 + a_3)^2} \tag{3-16}$$

The rotation of the dipper arm is obtained by considering the geometry of the dipper, Figure 3-11,

$$\theta_2 = \alpha + \psi \tag{3-17}$$

where

$$\alpha = \tan^{-1}\left(\frac{d_3}{a_2 + a_3}\right) \tag{3-18}$$

and

$$\psi = \tan^{-1}\left(\frac{y_{L,3}^L - y_{L,1}^L}{z_{L,3}^L - z_{L,1}^L}\right) \tag{3-19}$$

Once d_3 and θ_2 are obtained, the extension (retraction) of the dipper arm, d_c, controlled by the crowd motor can be determined as

$$d_c = d_3 - a_2\theta_2 \tag{3-20}$$

The motion of the dipper arm can then be related to the crowd motor via the crowd gear train with a gear ratio of n_{cm},

$$\theta_{cm} = \frac{d_c}{n_{cm}} \tag{3-21}$$

The inverse kinematics to determine the length of hoist rope has previously been published [7]. However, their calculations assumed the rope to have a constant wrap around the boom sheave; from Figure 3-11, $\hat{\alpha}_{bs}$ is constant. As they pointed out, the rope wrap around the boom sheave depends upon the configuration of the dipper arm. The following new analysis takes the configuration into consideration and adjusts $\hat{\alpha}_{bs}$ by α_{bs}.

In order to determine the wrap of the rope around the boom sheave, two vectors of interest are the distance between the centre of the boom sheave and the edge of the sheave where the rope departs from the sheave,

$$\boldsymbol{r}_{bs,r}^{bs} = \left[x_{bs,r}^{bs}, \quad y_{bs,r}^{bs}, \quad 0\right]^T, \tag{3-22}$$

and the distance from the bail connector to the edge of the sheave where the rope departs,

$$\boldsymbol{r}_{bc,r}^{bs} = \boldsymbol{r}_{bs,r}^{bs} - \boldsymbol{r}_{bs,bc}^{bs} \tag{3-23}$$

where $\boldsymbol{r}_{bs,bc}^{bs}$ is the location of the bail connector with respect to the center of the boom sheave. All three vectors are referenced from the frame located at the centre of the sheave. The fixed length of an equalizer bar that is located between the bail connector and the rope is absorbed into the length of the rope. It is assumed that the equalizer bar and the attached rope are aligned, although high friction at the equalizer bar-bail connection can prevent perfect alignment.

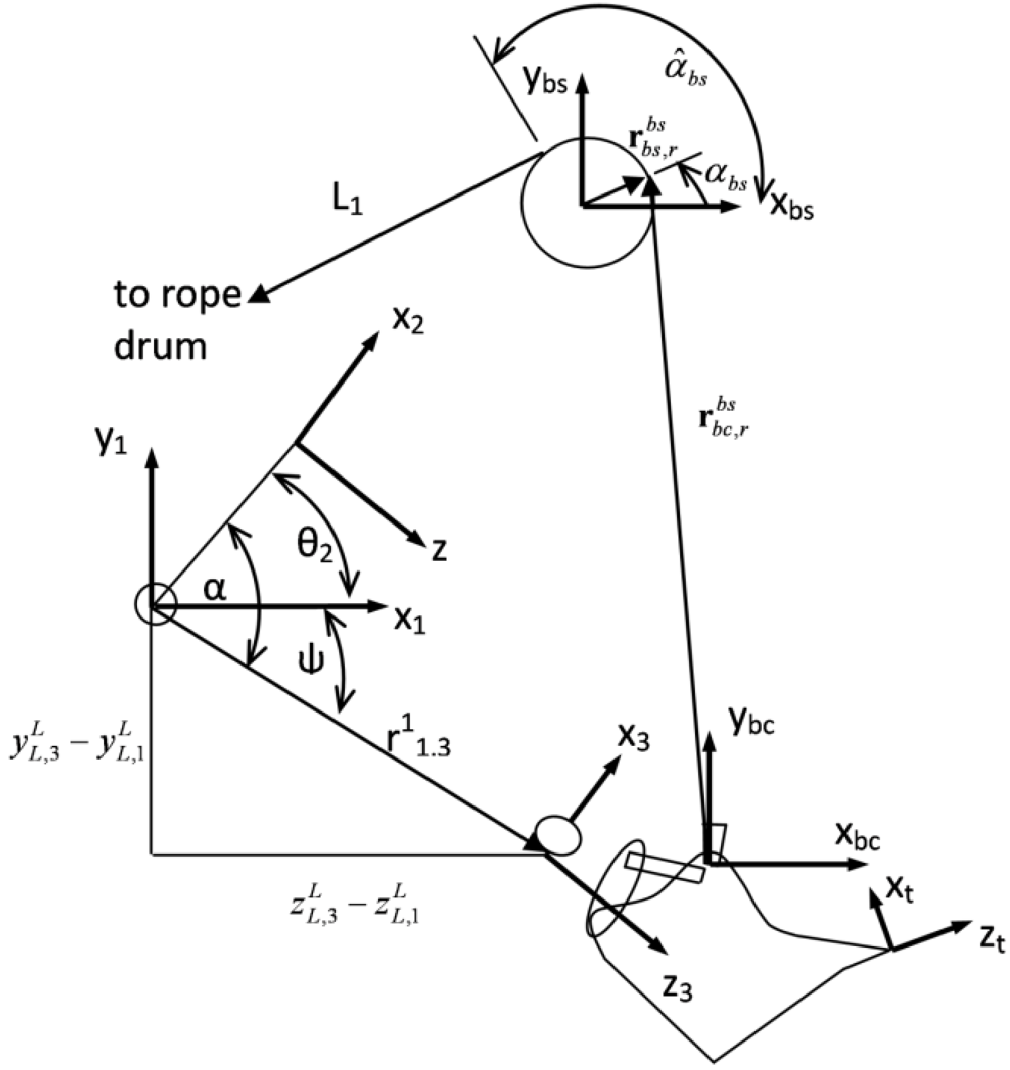


Figure 3-11: Geometry of the Dipper Arm for Inverse Kinematics

The location of the bail connector can be determined from the laser point-clouds as the transformation from the torsion bar to the bail connector, $\boldsymbol{T}_{bc}^{3}$, is fixed, as is the transformation between the shipper shaft and the centre of the boom sheave, $\boldsymbol{T}_{bs}^{1}$. The homogeneous transformation relating the bail connector frame to the boom sheave can then be determined,

$$\boldsymbol{T}_{bc}^{bs} = \left(\boldsymbol{T}_{bs}^{1}\right)^{-1} \boldsymbol{T}_{3}^{1} \boldsymbol{T}_{bc}^{3} \tag{3-24}$$

From this relation, the location of the bail connector with respect to the boom sheave can be extracted,

$$\boldsymbol{r}_{bs,bc}^{bs} = \left[x_{bs,bc}^{bs}, \quad y_{bs,bc}^{bs}, \quad 0\right]^{T} \tag{3-25}$$

and $\boldsymbol{r}_{bc,r}^{bs}$ from (3-23) is written as,

$$\boldsymbol{r}_{bc,r}^{bs} = \left[x_{bs,r}^{bs} - x_{bs,bc}^{bs}, \quad y_{bs,r}^{bs} - y_{bs,bc}^{bs}, \quad 0\right]^{T} \tag{3-26}$$

Two equations are developed to solve for the two unknowns in Equation (3-22) and Equation (3-26). First, an assumption is made that the rope departs the sheave at a right angle, mathematically this can be expressed as the dot product of $\boldsymbol{r}_{bs,r}^{bs}$ and $\boldsymbol{r}_{bc,r}^{bs}$ being equal to zero,

$$x_{bs,r}^{bs}\left(x_{bs,r}^{bs} - x_{bs,bc}^{bs}\right) + y_{bs,r}^{bs}\left(y_{bs,r}^{bs} - y_{bs,bc}^{bs}\right) = 0 \tag{3-27}$$

A second equation is formed by considering that the magnitude of $\boldsymbol{r}_{bs,r}^{bs}$, denoted as $r_{bs} = \left\|\boldsymbol{r}_{bs,r}^{bs}\right\|$, is fixed,

$$r_{bs}^{\ 2} = \left(x_{bs,r}^{bs}\right)^{2} + \left(y_{bs,r}^{bs}\right)^{2} \tag{3-28}$$

Solving Equations (3-27) and Equation (3-28) for $x_{bs,r}^{bs}$ and $y_{bs,r}^{bs}$ yields,

$$x_{bs,r}^{bs} = \frac{r_{bs}\left[r_{bs}\, x_{bs,bc}^{bs} \mp y_{bs,bc}^{bs}\sqrt{\left(x_{bs,bc}^{bs}\right)^{2} + \left(y_{bs,bc}^{bs}\right)^{2} - r_{bs}^{\ 2}}\right]}{\left(x_{bs,bc}^{bs}\right)^{2} + \left(y_{bs,bc}^{bs}\right)^{2}} \tag{3-29}$$

and

$$y_{bs,r}^{bs} = \frac{r_{bs}\left[r_{bs}\, y_{bs,bc}^{bs} \pm x_{bs,bc}^{bs}\sqrt{\left(x_{bs,bc}^{bs}\right)^{2} + \left(y_{bs,bc}^{bs}\right)^{2} - r_{bs}^{\ 2}}\right]}{\left(x_{bs,bc}^{bs}\right)^{2} + \left(y_{bs,bc}^{bs}\right)^{2}} \tag{3-30}$$

Two solutions are obtained. Observing that the boom sheave is located to the left of the rope, choose the solution that satisfies,

$$\boldsymbol{r}_{bc,r}^{bs} \times \boldsymbol{r}_{bs,r}^{bs} \leq 0 \qquad (3\text{-}31)$$

From here α_{bs} can be determined as,

$$\alpha_{bs} = \text{atan2}\,(y_{bs,r}^{bs}, x_{bs,r}^{bs}) \qquad (3\text{-}32)$$

and the rope length, *h*, between the sheave and the bail connector can be determined as,

$$h = \left\| r_{bc,r}^{bs} \right\| = \sqrt{\left(x_{bs,r}^{bs} - x_{bs,bc}^{bs}\right)^2 + \left(y_{bs,r}^{bs} - y_{bs,bc}^{bs}\right)^2} \qquad (3\text{-}33)$$

The total rope length is then

$$L = L_1 + (\hat{\alpha}_{bs} - \alpha_{bs})r_{bs} + h \qquad (3\text{-}34)$$

3.5 Experiments

3.5.1 Comparison of ICP, BPF and ICP-Aided BPF

A study was initiated on a mining shovel to compare three algorithms: ICP, the BPF, and the ICP-Aided BPF.

Setup: A SICK LMS221 laser scanner was mounted beneath the shovel boom of a P&H 2800 rope shovel at a mining operation in British Columbia (Figure 3-12). The laser scanner has a mean systematic error of 5.0cm over its operating range, with 1cm of standard deviation. Figure 3-1.b shows the relative position of the laser scanner to the dipper and truck. The laser scanner acquired 20,000 frames during 20 minutes of shovel operation.

Accurate measurements of the dipper pose in the workspace were not available at the time of our first experiment. However, based on the laser point-clouds, a comparison between manually assessed dipper pose and computer estimated dipper pose was made and the error between these was calculated.

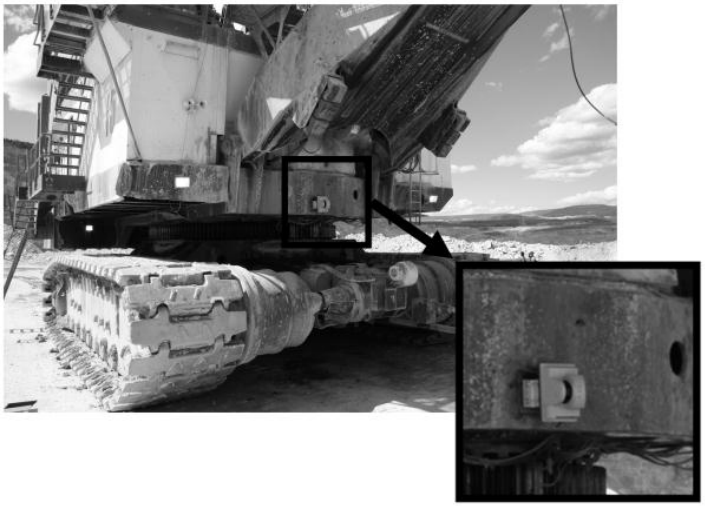


Figure 3-12: Vertical Installation of a Laser Scanner Underneath the Shovel Boom

The dipper pose was manually identified on all laser point-clouds. The manual assessment process used an ICP algorithm to track the dipper in all recorded laser frames, while a user observed the result for every frame. As soon as the ICP result was not perfectly aligned with the dipper, the user manually adjusted the estimated dipper pose. To adjust the pose, the user moved a dipper model point-cloud over the laser point-cloud in order to find the location that best matches it. The user was able to adapt quickly to the common problems such as shape variation and occlusion.

To evaluate the manual assessment process, a perturbed[4] point-cloud of the dipper model was located at a series of random positions in the workspace and the locations were recorded. The user was then asked to identify the dipper positions. Comparing 100 position identifications with the actual recorded positions,

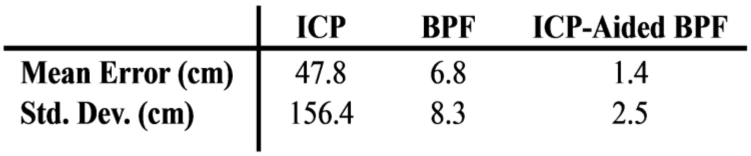
[4] The perturbing of the point-cloud was done to simulate laser measurement noise: each laser measurement value in the point-cloud was changed by a random amount based on the laser's normal error distribution reported by the manufacturer.

the manual assessment error was found to be a Gaussian distribution with a mean (μ) of 4.7cm and a standard deviation of 2.9cm (σ).

Results: To evaluate each algorithm's performance, its pose estimations were compared to the manually identified pose values by computing the Euclidean distance between the two.

The ICP algorithm obtained a mean error of 47.8cm with a standard deviation of 156.4cm. The BPF was better than ICP with a mean error of 6.8cm and a standard deviation of 8.3cm. Finally, the ICP-Aided BPF, the proposed pose estimation method in this research, obtained a mean error of 1.4cm and a standard deviation of 2.5cm (see Table 3-2). The 1.4cm is not the error between the ICP-Aided BPF's and the true dipper pose because there is an inherent error in the manual assessment data of 4.7cm. Note that the error reported for the ICP algorithm is high because of its frequent tracking failures. The term "reliability" is used to mean freedom from tracking failures.

Table 3-2: Error Between ICP, BPF, and ICP-Aided BPF Algorithms, and Manual Pose Assessments



	ICP	BPF	ICP-Aided BPF
Mean Error (cm)	47.8	6.8	1.4
Std. Dev. (cm)	156.4	8.3	2.5

Performance comparison between ICP, a BPF with no ICP, and an ICP-Aided BPF, based on laser scanner data representing 15 dig-dump cycle. The errors represent the Euclidean distance between the manually assessed (*x*, *y*) dipper positions, and the positions obtained by each algorithm.

3.5.2 ICP-Aided BPF vs. Joint Angle Sensors

Setup: Given the indication that the ICP-Aided BPF would be the fastest and most accurate, the P&H 2800 shovel was instrumented with arm geometry sensors to evaluate the laser-based joint variable estimations. A linear range finder (SICK DT500) with a worst-case accuracy of 3mm was used to measure the crowd extension (d_3), and the accelerometer-based angle sensor of [54] with a worst-case accuracy of 1.6˚ was used to obtain the crowd angle (θ_2).

Inverse kinematics was applied to the ICP-Aided BPF's dipper positioning results to obtain the shovel joint variables, and a comparison was made between the laser-based joint variable results and the sensor-based joint variable readings.

Results: The crowd and hoist lengths calculated from the inverse kinematics are shown in Figure 3-13.a. The dig-dump cycle in the figure started with the operator dumping into the truck at time *t1*, followed by hoisting the dipper up during the *t1*-*t2* time interval. The shovel is then swung over to the ore face while lowering and retracting the dipper during the *t2*-*t3* interval. Subsequent to time *t3*, the dipper is extended and hoisted into the ore face and then swung back for dumping at time *t4*. The dipper position is illustrated in Figure 3-13.b for *t1*, *t2*, and *t3*.

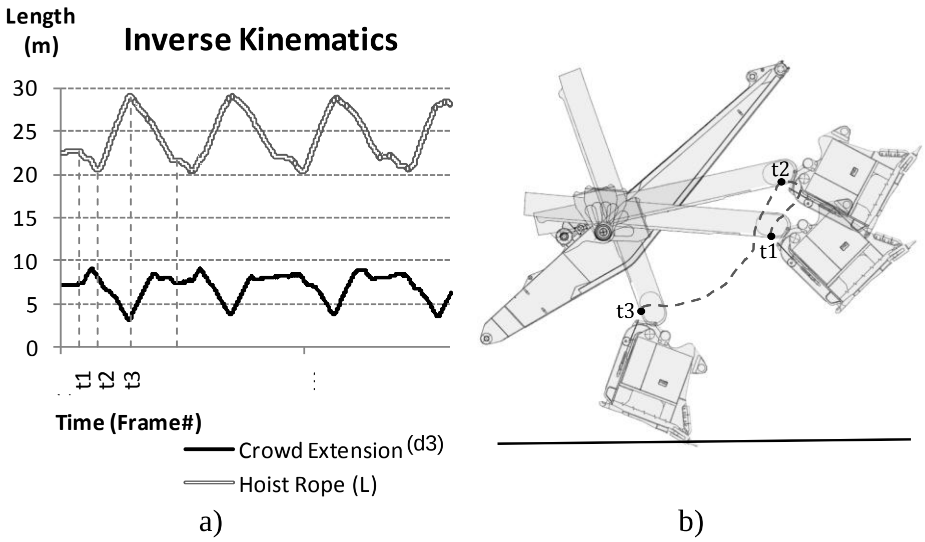
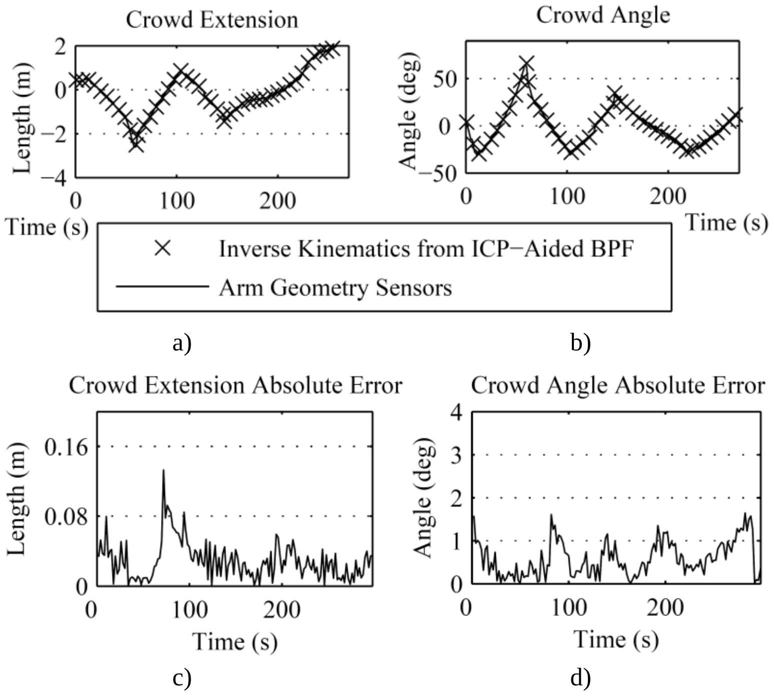


Figure 3-13: Inverse Kinematics Results of the ICP-Aided BPF on a Rope Shovel

a) The inverse kinematic results for the crowd extension (*d_3*) and the hoist rope length (*L*). b) A graphical representation of the dipper at times *t1*, *t2*, and *t3*. The operator dumps into the truck at *t1*, hoists the dipper up (*t1*-*t2* time interval), swings over to the ore face while lowering and retracting the dipper (*t2*-*t3* interval), and extends the dipper into the ore (*t3*-*t4* interval).

Figure 3-14 presents a comparison between the arm geometry variables extracted from the ICP-Aided BPF's dipper positioning, and the readings from the arm geometry sensors. The correspondence of the sensor values and the algorithm results validates the proposed approach.

Table 3-3 presents the accuracy of the arm geometry estimation results with respect to the arm geometry sensors. Moreover, forward kinematics is applied to the readings from the arm geometry sensors to estimate the dipper pose. The result is compared to the ICP-Aided BPF's dipper pose estimates to determine its accuracy (see Table 3-3).



a) b)

c) d)

Figure 3-14: Error in Inverse Kinematics Results of the ICP-Aided BPF on Rope Shovel

a) Comparing the algorithm's crowd angle estimates (θ_2) with the accelerometer readings. b) Comparing the algorithm's crowd extension estimations (d_3) with the range finder readings. c) The error graph showing the difference between the sensor data and the algorithm's crowd extension values. d) The error graph for the algorithm's crowd angle values.

Table 3-3: ICP-Aided BPF vs. Joint Variable Sensors

	Dipper Position	Crowd Extension	Crowd Angle
Mean Error	6.7 cm	2.9 cm	0.6°
Std. Dev.	4.8 cm	2.2 cm	0.4°
Max Error	21.5 cm	13.3 cm	1.6°

The ICP-Aided BPF's dipper pose and joint variable estimations compared to the readings from the arm geometry sensors. The dipper pose error is the Euclidean distance between the ICP-Aided BPF's dipper pose estimation, and the dipper pose determined by forward kinematics acting on joint variable sensors.

3.6 Discussion

3.6.1 Value of Integrating BPF and ICP

The manually-assessed results presented in Section 3.5.1 compare the dipper tracking performance of the ICP algorithm, the BPF, and the ICP-Aided BPF. When used on a rope shovel without a global search, ICP demonstrates considerable inaccuracy (mean error: 47.8cm) and instability (standard deviation: 156.4cm). ICP's error is very high because of its frequent tracking failures. The failures are due to problems such as occlusion and aliasing that cause a loss of geometric features in dipper point-clouds.

The BPF method is significantly more accurate (mean error: 6.8cm) and stable (standard deviation: 8.3cm). It can be seen that the BPF's global search has a significant advantage to ICP's local search approach.

Finally, the integration of BPF and ICP obtains the best results with a mean error of 1.4cm and standard deviation of 2.5cm. While the BPF provides reliable results – no registration failures – the ICP-Aided BPF considerably improves the tracking accuracy by using ICP to perform a fine registration based on the BPF's result. The improvements in accuracy and reliability support the decision to integrate the ICP algorithm and the BPF.

3.6.2 Performance of the ICP-Aided BPF

The performance of the proposed method can be evaluated based on three factors: real-time processing, accuracy, and reliability.

3.6.2.1 Real-time Processing

Running a single-threaded process on a 1.86GHz CPU, the ICP-Aided BPF performed each estimation in 51.2ms, thus obtaining a real-time performance of 19.5Hz. Further implementation optimization is possible.

3.6.2.2 Accuracy

ICP-Aided BPF's average bucket positioning error of 6.7cm is 0.3% of the dipper's reach, which is close to the laser scanner's mean error of 5.0cm. However, the maximum error of the system is higher at 21.5cm. An important cause for the higher maximum error is likely partially due to the limitations in the joint sensors used for accuracy evaluation. The joint angle sensors have a worst-case accuracy of 1.6˚, which could lead to a dipper position error of up to 26.0cm at the maximum crowd length of 9.3m. This means that due to the limitations of the arm geometry sensors selected for evaluating the laser-based ICP-Aided BPF, the obtained 21.5cm maximum positioning error is as good as one can expect. Other sources of dipper positioning error are the point-cloud tracking challenges introduced in Section 2.2.1 such as occlusion and aliasing.

The laser-based bucket tracking system proposed by Duff is reported to be accurate within 10cm [11]. Duff’s system was installed on a Hitachi EX36000 hydraulic excavator with an arm reach of 12m. The experimentation in this work was performed on a P&H2800 rope shovel with an arm reach of 22.5m. While the laser scanner error does not increase significantly as the bucket moves away from the sensor, the number of laser points on the bucket decreases linearly. A decrease in the number of laser points on the bucket causes aliasing and thus a loss of geometric detail. Therefore, when tracking the dipper of a shovel with a long arm reach, a higher positioning error is expected. Additionally, occlusion has been observed to be a cause for the occasional high error values. For example, the worst case error of 21.5cm occurred when the torsion bar was partially occluded by the boom.

3.6.2.3 Reliability

In the publicly-available reported summary of their work, Duff [11] reports a lack of reliability in bucket tracking caused by a loss of geometric features on the bucket. When ICP is used alone, we observed similar unreliability and frequent tracking failure. When using the ICP-Aided BPF, a decrease in accuracy was observed when facing occlusions. However, our experiments demonstrate that the proposed bucket tracking algorithm is consistently capable of reliable bucket tracking, notably during the digging cycle or when faced with occlusion. Further work would be required to determine long-term reliability.

Depending on the location of the dipper in space, the laser point-clouds can represent the entire dipper in as few as 20 laser points, half of which correspond to the portion of the dipper used for tracking. Thus, because of the proposed algorithm’s capability to reliably track such small and non-distinctive point-clouds, the presented approach can be used for tracking various other robot end-effectors, such as the buckets of smaller hydraulic excavators used in construction.

3.6.3 Estimation of Dipper Tracking Uncertainty

In the real world, severe cases of occlusion and aliasing can occur, raising the uncertainty in the dipper tracking results. The tracking uncertainty can be estimated by two measures in the ICP-Aided BPF algorithm: first, the distribution of the BPF particles presents a reliable measure for the confidence level of the tracking results. Lack of particle concentration or multiple concentrations in different regions represent high uncertainty in the outcome.

A second measure for estimating the tracking uncertainty is the registration error obtained from the ICP algorithm. Although a small ICP error value does not represent high confidence in the tracking result (mainly due to ICP's susceptibility to local minima), a large ICP error value undoubtedly represents a poor result. In the past, Kim and Kim have used the ICP registration error to gauge the uncertainty of the results [57].

BPF particle distribution and ICP registration error provide potential means to estimate the tracking uncertainty and inform the operator when the uncertainty is high. However, this was not implemented in this work as it was not necessary for fulfilling the objective of the project at this stage.

3.6.4 Model Point-Cloud Selection

Choosing an adequate dipper model is important for obtaining a reliable dipper tracking performance. In this work, the dipper was first positioned in the scene without occlusions by other objects. The dipper model was then extracted from the laser point-clouds, using the segment of the point-cloud that corresponds to the torsion bar and the dipper (see Figure 3-15).

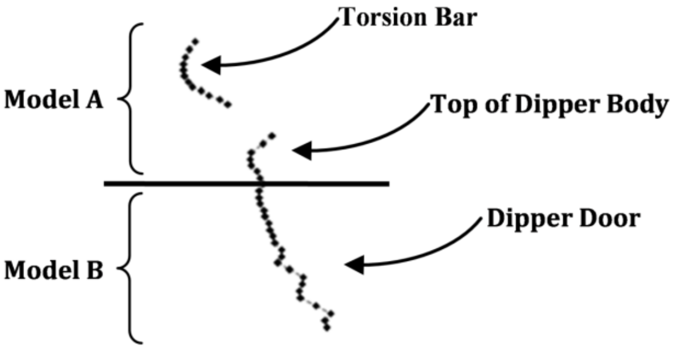


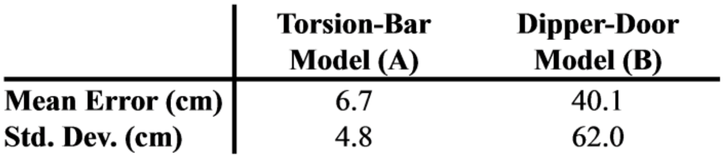
Figure 3-15: Segments of the Shovel Dipper

The dipper point-cloud divided into two segments each representing a rigid body. Model A consists of the torsion bar and the top of the dipper, and Model B consists of the dipper door.

The point-cloud in Figure 3-15 is divided into two sections: A) the upper section consisting of the torsion bar and the top of the dipper that moves with the dipper handle, and B) the lower section representing the dipper door that has an extra degree of freedom – the door's rotation about the door joint. A model point-cloud must be a rigid body. Thus, it can only be either of the two segments, since the two do not form a rigid body when put together due to the dipper door opening up.

Table 3-4 presents the performance of the ICP-Aided BPF algorithm when using both the torsion-bar model (Model A) and the dipper-door model (Model B), when compared with the joint variable sensor readings. The performance degrades when Model B is used since the dipper-door becomes considerably occluded by the truck during the dumping stage. The torsion-bar model, on the other hand, suffers from less shape variation and occlusion. Hence, due to its better performance, this work used the torsion-bar model for dipper tracking.

Table 3-4: Performance Comparison of the Two Model Point-Clouds



	Torsion-Bar Model (A)	Dipper-Door Model (B)
Mean Error (cm)	6.7	40.1
Std. Dev. (cm)	4.8	62.0

3.7 Conclusions

This work presents a novel approach for tracking a mining shovel’s dipper in its workspace, which can be used to extract the shovel’s arm geometry. The novelty of this work resides in its integration of a coarse registration algorithm – BPF – and a fine registration algorithm – ICP – to obtain fast, accurate and reliable performance.

The proposed Bootstrap Particle Filter provides a global search and uses a geometrical model of the dipper that is defined from a laser point-cloud of the shovel. Temporal data and the dipper's kinematic constraints reduce the size of the search space. To achieve required system performance, the inclusion of the Distance Transformation for particle evaluation enables the algorithm to run in real-time. The output of the BPF is passed to an ICP local search step to improve the accuracy.

The proposed system is a generic end-effector tracking solution that addresses a number of critical problems in laser-based tracking applications: the partial overlapping problem, the need for accurate initialization for fine registration, the challenges of tracking non-distinct and highly varying object point-clouds in a dynamic scene, and problems with occlusion and aliasing.

With the laser scanner utilized in this project, the mean dipper positioning error is shown to be within 6.7cm (0.3% of the dipper’s effective reach). For applications requiring dipper tracking and arm geometry extraction, the ICP-Aided BPF algorithm is capable of producing this accuracy in real-time. To the best of our knowledge, the presented approach obtains the most accurate and reliable real-time laser-based end-effector tracking achieved to date. Moreover, compared to the joint variable sensors, the laser scanner is simpler to mount and could potentially reduce maintenance costs and machine downtime.

The work presented here shows promising results for proximal joint variable extraction on large excavators. However, additional work is required before this can be utilized on a mine site. In this work, the extrinsic calibration of the laser scanner's coordinate frame – the distance vector between the base frame F_b and the laser scanner F_L in Figure 3-9 -- was obtained using manual measurements and with the help of the manufacturer drawings of the laser scanner and the shovel. Future work needs to investigate a general approach for estimating the extrinsic calibration parameters of the laser scanner, using a series of pre-designed calibration experiments for instance. For the proposed approach to be used in mining operations as a safety system, manual installation may be unreliable, especially when done by personnel not familiar with computer vision. A simple and accurate calibration procedure is required for installation, and once the installation is completed, field procedures are needed to ensure the collision avoidance system operates within the required accuracy tolerance.

Chapter 4

Active and Passive Sensor Fusion Using Bilateral Filtering on Depth Probability[5]

Depth perception is a fundamental problem in computer vision with many applications such as robot navigation and 3D video. Depth sensors capture *depth images* – an image in which pixel values represent distance to the sensor. A 3D laser scanner point-cloud can be transformed into a depth image – and vice versa. Significant progress has been made on depth sensing using passive (e.g., stereo) and active (e.g., laser, TOF) sensing. However, inaccurate results of stereo imaging caused by the ambiguity problem in stereo matching, and the slow, noisy and sparse results of laser scanners and time-of-flight (TOF) cameras fail to meet the requirements of many applications. For instance, as described in Introduction, the planar laser scanner used in Chapter 3 cannot be used to extract certain workspace information such as surface geometry and dipper load profile.

In recent years, fusion of an active 3D depth sensor (e.g., 3D laser scanner or a TOF camera) with a passive sensor (i.e., intensity cameras) has shown potential for addressing the limitations discussed. For fusion, the two sensors are positioned next to each other and have a similar viewing angle so that they simultaneously capture the same scene. The choice of the 3D depth sensor depends on the application. Indoor applications can use TOF cameras that have a higher frame-rate than 3D laser scanners, whereas outdoor applications require the use of laser scanners that have a longer measurement range and are better suited for handling environmental factors.

[5] A version of this chapter has been submitted for publication [75].

Super-resolution algorithms are commonly used for integrating images from an active sensor and a passive sensor (i.e., “active-passive sensor fusion”) [30–32], [76], [77]. Super-resolution is a technique for obtaining a high-resolution image by enhancing information from a low-resolution system. In effect, it attempts to upscale a smaller image while intelligently adding high-frequency information to it in order to avoid blurring. Super-resolution methods can take advantage of multiple image sources. In the case of active-passive fusion, small images from an active depth source are upsampled with the help of high-frequency data extracted from a high-resolution (passive) camera.

This chapter proposes a super-resolution algorithm for integrating a 3D laser scanner and camera images. The proposed algorithm is referred to as *Super-resolution on Depth Probability* (SDP). SDP is a computationally efficient method that uses bilateral filtering – an edge-preserving smoothing filter – to upsample small laser depth images with the help of a high-resolution camera. The proposed method obtains considerably more accurate results (an error reduction of up to 6.8x) compared to the literature, with sharper and more realistic edge definition. This is shown quantitatively using a standard set of benchmark camera and depth sensor images (Middlebury Stereo [78]) as well as a set of laser-scanned scenes obtained in this work. The proposed method is well-suited for the active depth sensor’s sparse depth measurements which are not evenly-distributed in the *x*-*y* image coordinates. Moreover, experiments show that the proposed method requires minimal parameter adjustment across different scenes, which contributes to making it practical for real-world applications.

4.1 Background

Active depth sensing has been integrated with camera images in many applications such as environment modelling and robot navigation. Many works in the literature [17], [79], [80] propose *decision-level* integrations in which data from individual sensors are processed separately, and used together during the decision-making process [81]. For instance, in [17], [79], a real-time obstacle detection system is proposed for autonomous vehicles that uses a camera to find road boundaries, and a laser scanner to find the obstacles within those boundaries. Other works [82–84] perform *feature-level* integration which means combining features extracted separately from each sensor data [81]. Examples include [82] that maps the robot's environment by extracting 3D features from laser and stereo-cameras separately, and places them in a single map.

The focus of this work, however, is on *data-level* integration, which directly integrates the data from the sensors prior to processing, rather than extracting and integrating features by processing each sensor image separately [81]. Data-level integration of a depth sensor and a camera is proposed to obtain enhanced depth images in terms of depth accuracy and spatial resolution. Many works in the literature have proposed data-level fusion methods [28], [30–32], [34–36], [76], [77], [85–104]. In applications such as Geographical Information System (GIS) mapping and environment modelling, texture-mapping algorithms are proposed that register depth measurements with color information to create a virtual model of the scene [85–93]. These algorithms provide color values for every depth measurement, but do not improve the spatial resolution of the depth sensor image and therefore require many minutes for the depth sensor to capture a dense 3D image [86], [105]. On the other hand, super-resolution algorithms – as introduced in Section 1.4.2 – integrate data from a depth sensor with an image from a camera by upsampling the low-resolution depth sensor image with the help of the high-resolution camera image [28], [30–32], [34–36], [94–96], [106]. The upsampling approach has also been applied to the integration of a depth sensor with two or more intensity cameras [76], [77], [96–102], [104]. This chapter proposes a super-resolution method for a single camera and a depth sensor, in this case a laser scanner, in order to obtain 3D images with high accuracy (similar to the laser) and high spatial resolution (similar to the camera).

By extracting high-frequency information from the camera's *x-y* image plane and introducing that into an upsampled laser image, super-resolution generates depth estimates at pixel locations without a depth sensor measurement. To demonstrate the value of this, consider the following: At a 30 meter range, a typical laser scanner with 0.5° angular resolution takes depth measurements that are 26.2cm apart horizontally. Thus the laser measurements representing object edges may be up to 26.2cm off the actual edge. The depth measurement error of the laser scanner as well as its unreliability on edge measurements (i.e., the laser beam-splitting effect causing edge averaging), further add to the above to make applications such as object localization and tracking inaccurate when using laser data alone. Fusing a camera with the sparse laser data, however, can address the above problem. An 800x600 camera image with a 66˚ HFOV, for instance, lowers the horizontal distance between neighbouring pixels at a similar 30 meter range, from 26.2cm to only 4.3cm. Higher resolution cameras can also be employed to further improve the spatial *x-y* resolution. The performance of this method is dependent on the camera's ability to capture the high-frequency information of interest (e.g., the edges), which may be limited by issues such as lighting and visibility.

In super-resolution, it is generally assumed that depth and color discontinuities coincide [31], [34], [36], [94], [97], although no experimental work was found to validate this assumption. While this is most often the case, there are exceptions: 1) In highly textured areas, color edges do not correspond to depth edges. This can cause "texture-copying" artefacts [34] in which textures from a camera image translate into incorrect geometries in the upsampled depth image. 2) When neighbouring objects are similarly colored, or when lighting is not suitable (e.g., saturated, dark, etc.), color edges may not exist where depth edges are located. This causes blurring of the depth edges when upsampling [34]. 3) Intrinsic sensor calibration errors and *extrinsic calibration errors* in sensor alignment cause the correlated color-depth discontinuities to appear at slightly different locations. This, too, can cause edge-blur in the upsampled depth image.

Two classes of super-resolution algorithms have been widely discussed in the literature of active-passive sensor fusion: a probabilistic approach using a Markov Random Field (MRF) model, and a bilateral filtering approach.

4.1.1 The Markov Random Field (MRF) Method

The MRF approach was first suggested by Diebel and Thrun [31]. They introduce two cost functions: 1) a depth measurement potential, used to make sure the upsampled depth image matches the original laser depth image, and 2) a depth smoothness prior, used to interpolate the depth values on uniform color textures and create smooth surfaces. The objective is then to find an upsampled depth image that best minimizes the sum of the two cost functions. Conjugate gradient is used to perform the MRF optimization [31]. In this chapter, we refer to Diebel and Thrun's approach, as the MRF method.

Figure 4-1 shows the Markov Random Field network designed by Diebel and Thrun in [31]. On the top layer, sparse laser depth measurements (L) are provided, and in the bottom layer, dense camera image pixels (I) are given. The middle layer is the depth image (D) that is being created using this super-resolution technique. Note that the density of the reconstructed depth image is similar to the camera image in the bottom layer. The gradient nodes on the bottom and middle layer – image intensity gradient δI, and depth discontinuities δD – connect the two layers together, and serve to provide the camera image information to the depth image that is being reconstructed.

Two potentials are defined within the cost function: depth measurement potential, and depth smoothness prior. An optimization technique is then used which iteratively estimates the depth pixels D in order to minimize the cost function.

Depth measurement potential is as follows:

$$\Psi = \sum_{i \in L} k\ (D_i - L_i)^2 \qquad (4\text{-}1)$$

where k is a constant weight placed on depth measurements [31]. This potential evaluates the distance between the reconstructed depth image and the laser measurements where they are available.

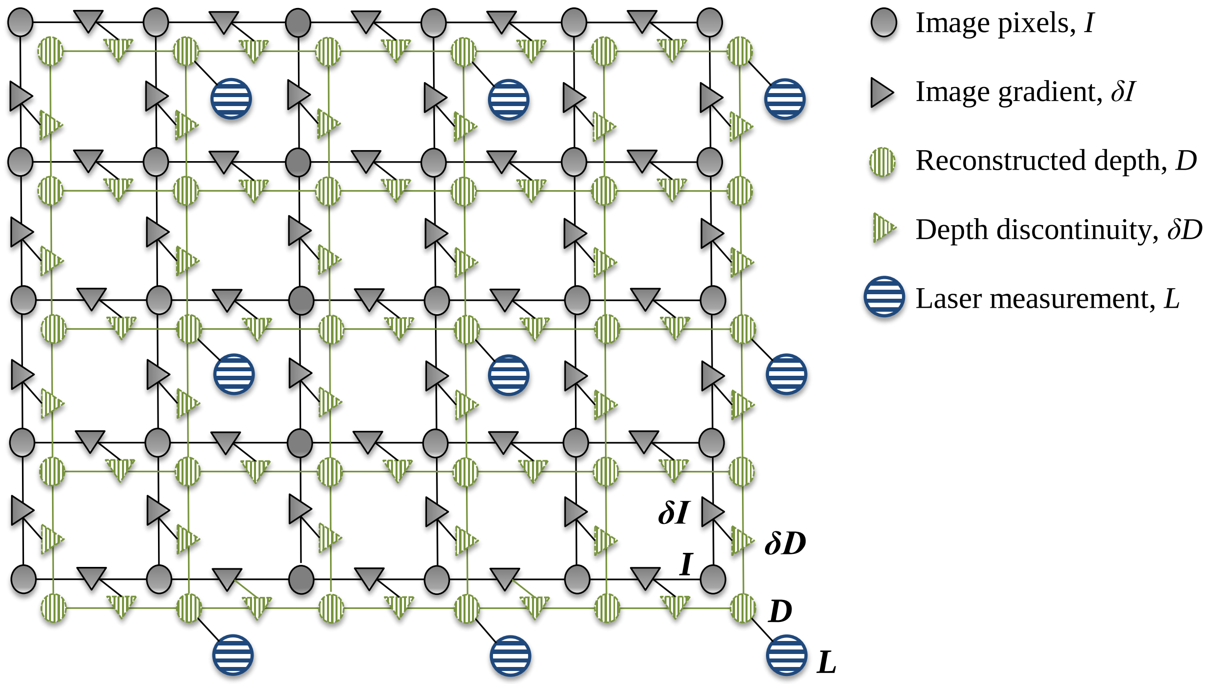


Figure 4-1: Markov Network in for Active-Passive Super-resolution

The network consists of five types of nodes: laser measurements (*L*), image pixels (*I*), image pixel gradients (*δI*), reconstructed depth pixels (*D*) and depth discontinues in the reconstructed depth image (*δD*). As shown laser measurements are sparse, compared to the image pixels.

The depth smoothness prior is defined as:

$$\Phi = \sum_{i \in I} \sum_{j \in N(i)} w_{ij} \left(D_i - D_j\right)^2 \qquad (4\text{-}2)$$

where $N(i)$ is the set of all pixels adjacent to i [31]. This potential introduces a smoothness constraint in the reconstructed depth image. The weighting factors, w_{ij}, are the variables that link the bottom and middle layer of the Markov network together:

$$w_{ij} = e^{-c\|I_i - I_j\|^2} \qquad (4\text{-}3)$$

where $\|I_i - I_j\|$ is the Euclidean distance between the RGB values of two image pixels. Constant c defines how much the algorithm tolerates depth smoothing across color edges: higher values make the system less forgiving of color variations, thus less depth smoothing will occur across a color edge.

The Markov Random Field network is then defined based on Ψ and Φ:

$$p(D|I, L) = \frac{1}{Z} e^{-1/2\,(\Psi+\Phi)} \qquad (4\text{-}4)$$

where Z is a normalizing factor [31]. Diebel and Thrun suggest the well-known conjugate gradient (CG) algorithm [107] to estimate the posterior of the network within seconds. Other methods such as a loopy belief propagation are computationally expensive for a Markov network with millions of nodes [31].

Gould et al. [36] improve the above MRF formulation by removing its implicit preference for fronto-parallel planes. Zhu et al. [77], on the other hand, use a different MRF model to integrate a TOF camera with disparity data from a stereo-camera. Thus, texture information is not directly used for super-resolution. Belief propagation is applied to solve the optimization problem [77].

4.1.2 The Yang Method

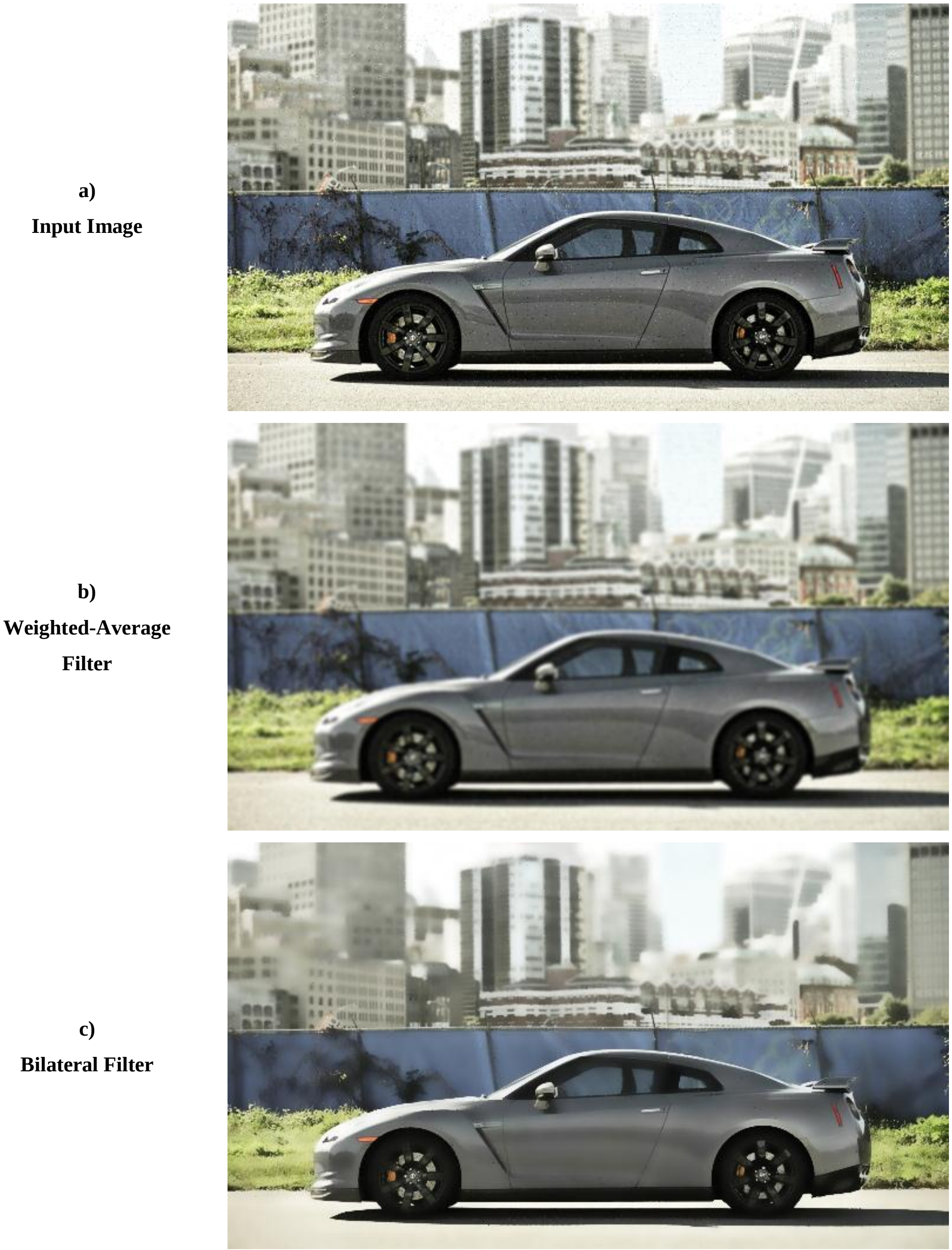
Bilateral filtering [33] is a smoothing filter similar to a weighted-average filter that reduces intensity variation among neighbouring camera pixels to remove noise. This is done by using a convolution matrix to set the value of each pixel to the weighted average of values in its surrounding pixels. The kernel in a weighted-average filter is a Gaussian distribution that weights each surrounding pixel based on its Euclidean distance to the original pixel – the higher the distance, the lower the weight. In bilateral filtering, however, the filter kernel is also based on color variation in the surrounding region, with similar color pixels being assigned a higher weight. Thus, while a standard weighted-averaging filter applies a constant kernel to all image pixels, a bilateral filter calculates a new kernel for each pixel based on the color variations of the neighbouring pixels. The term bilateral filter is coined due to the fact that the filter is constructed based on two parameters: Euclidean distance of pixels, and color variation across neighbours.

The following equation represents how a bilateral filter constructs unique convolution kernels for each pixel [30], [32]:

$$\begin{aligned} f_c(x) &= e^{-\frac{|x|}{\lambda_c}}, \quad f_s(x) = e^{-\frac{|x|}{\lambda_s}} \\ F_{i,k} &= f_c\left(\frac{1}{3}\sum_{c=RGB}\left|\widehat{I}_i^{\,c} - \widehat{I}_k^{\,c}\right|\right) \cdot f_s\left(\left\|i-k\right\|\right) \end{aligned} \qquad (4\text{-}5)$$

where f_c is the color component and f_s is the spatial component of the filter. The constants λ_c and λ_s are used to configure the sensitivity of the filter to color and spatial variations, and *i* and *k* are the 2D coordinates of the pixel. *F is* the bilateral filter and $\widehat{I}$ is the camera image. Also, $\left\|i-k\right\|$ represents the Euclidean distance between pixels *i* and *k* in the image's *x*-*y* coordinate frame.

Figure 4-2 compares the results of a standard weighted-average filter to a bilateral filter, when used for image smoothing and noise removal. The idea behind bilateral filtering is to adaptively form a smoothing kernel at each pixel based on the intensity information of the neighbouring pixels, in order to preserve edges while smoothing. Bilateral filtering does not blur object edges because pixels that do not belong to the same object are less likely to affect each other due to their often larger color difference.



a)
Input Image

b)
Weighted-Average
Filter

c)
Bilateral Filter

Figure 4-2: Bilateral Filter as an Edge Preserving Weight-Average Filter

a) Original image. b) A weighted-average filter is applied to the original image. Blurring of edges are visible. c) A bilateral filter is applied to the original image. While the textures have been smoothened, the edges remain intact.

Joint bilateral upsampling (JBU) [32] is a modification of the bilateral filtering algorithm used for 3D super-resolution applications. In JBU, the filter kernel is created using the camera image. The edge-preserving filter is then applied to the low-resolution depth image from a laser or TOF sensor, in order to upsample it to the size of the camera image. This way, the texture information from the camera image guides the upsampling process of the depth image, by preserving edges and only interpolating depth values within distinct surfaces. As shown in [34], JBU achieves upsampling results comparable to the MRF method in [31].

Yang et al. [30] construct *a cost-volume* – a 3D matrix representing cost as a function of depth for all pixels – and iteratively apply a bilateral filter to each slice (*cost image*) of the cost-volume. Figure 4-3 shows the structure of a cost-volume, and Figure 4-4 shows an example of it. The result of bilateral filtering on cost-volume is then converted into a depth image using the 'winner-takes-all' approach and a quadratic subpixel resolution equation [30], described later in this chapter. We refer to this method as the Yang method.

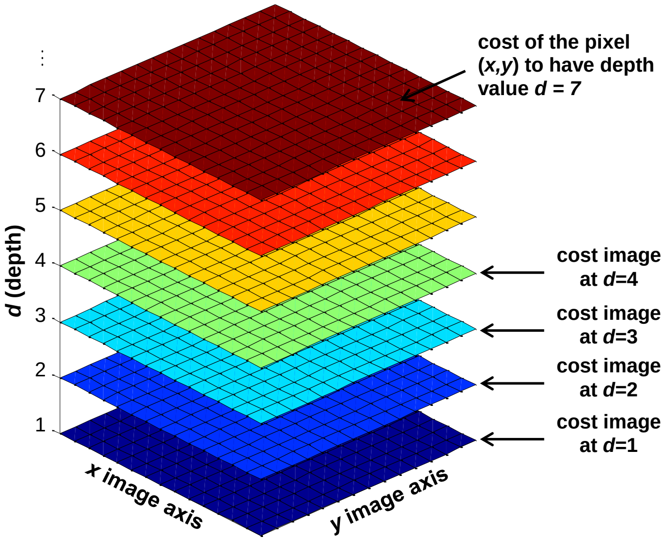
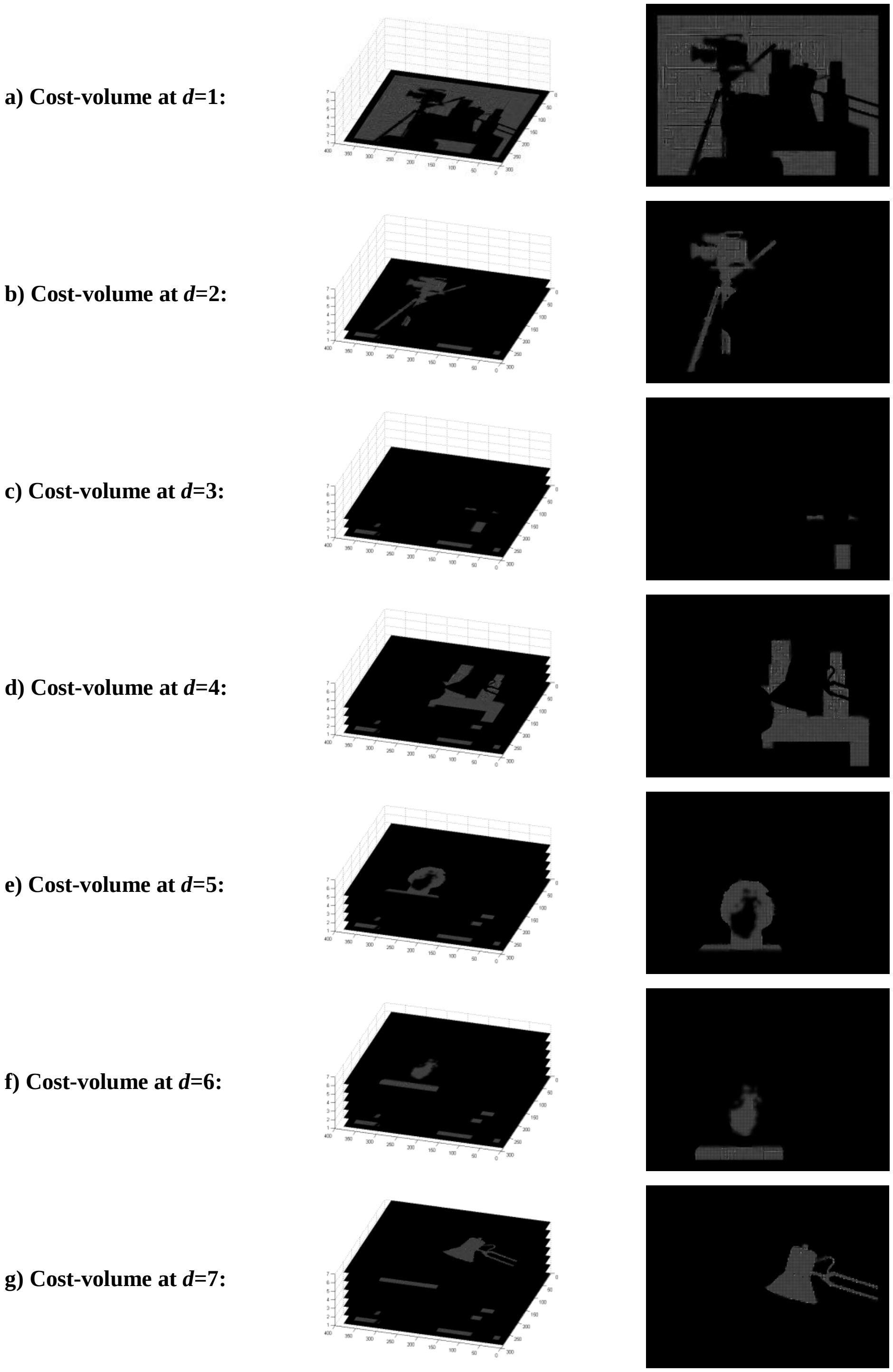


Figure 4-3: 3D Cost-Volume

Cost-volume is a 3D matrix representing cost as a function of depth (*d*) for all pixels at (*x*,*y*). As shown in the image, each vertical slice of the cost-volume represents a *cost image* in which *d* is constant.



a) Cost-volume at *d*=1:

b) Cost-volume at *d*=2:

c) Cost-volume at *d*=3:

d) Cost-volume at *d*=4:

e) Cost-volume at *d*=5:

f) Cost-volume at *d*=6:

g) Cost-volume at *d*=7:

Figure 4-4: 3D Cost-Volume, Tsukuba Example

This figure displays the cost-volume for the Tsukuba image in the Middlebury dataset. Each row shows the cost-volume at a specified depth level (*d*). On the right, the cost image at each *d* is shown. And on the left, the construction of the 3D cost-volume is demonstrated: at each depth level, the 3D cost-volume is a stack of cost images from the lower levels (i.e., starting from *d*=1 up to the current *d*).

Bilateral filtering of the cost-volume as opposed to the depth image leads to sharper edge definition compared to JBU [32]. This is because filtering a depth image inevitably causes some blurring of edges since it is the depth values that are being averaged. When filtering a cost-volume, it is the cost values – or in our algorithm, the *depth-probability* values – that are being averaged instead.

Yang's method can be outlined in the following steps:

1. Upsample the low-resolution depth image using basic interpolation, to the same resolution of the high-res camera image. This is stored as $D_{(1)}$.
2. Iterate through the following steps – *k* being the iteration number:

   2.1. Construct a cost-volume using a truncated quadratic model:

   $$C_{(k)}(i,j,d) = min\left(\xi, \left(d - D_{(k)}(i,j)\right)^2\right) \qquad (4\text{-}6)$$

   where *i* and *j* represent the *x*-*y* location of a pixel in the image plane, and *d* is a candidate depth value. Also, $D_{(k)}$ represents the depth image generated in the latest iteration. Note that to construct a cost-volume, depth is discretized into *Z* depth values. ξ is a truncation constant.

   2.2. A bilateral filter is then applied to the cost-volume: $\bar{C}_{(k)} = BilateralFilter\left(C_{(\mathrm{k})}\right)$. Section 4.2.3.2 describes how a bilateral filter is constructed and applied to a cost-volume.

   2.3. A new depth image $D_{(k+1)}$ is generated using sub-pixel resolution calculations on $\bar{C}_{(k)}$. Section 4.2.3.3 describes the equations for subpixel resolution.

   2.4. Iterations stop when a maximum number of iterations are reached or the changes in consecutive depth images, $sum(D_{(k+1)} - D_{(k)})$, is less than a desirable threshold.

The upsampling of a low resolution depth image in Yang method (step 1) makes an inherent assumption that the depth measurement pixels are evenly distributed in the *x*-*y* coordinate frame of the high resolution image. This is not the case for depth images generated by sensors such as laser scanners, which makes the upsampling step more complex. Furthermore, Yang reconstructs a new cost-volume at each iteration (step 2.1). As we explain later in Section 4.4.1, this approach throws away potentially useful information and helps erroneous depth estimates propagate in the image. Finally, the method proposed by Yang et al. does not consider sensor calibration error and occlusion problems, which can degrade accuracy of the outcome if not properly dealt with. This is because most papers with reported results, including Yang et al. [30], use the Middlebury [78] stereo images and create a synthetic TOF image by downsampling the ground-truth images, thereby avoiding problems such as noise and calibration error present in sensors.

Chan et al. [34] take a similar approach to JBU by applying a joint bilateral filter to the depth image itself, but use a TOF camera and improve the filtering equation to model its noisy depth measurements. The result reduces the texture-copy artefacts introduced by TOF noise. Moreover, Chan et al. [34] use a GPU implementation to achieve real-time computation.

Other papers also propose optimizations such as hierarchical filtering and parallelization to achieve real-time performance [28], [37], [108]. Moreover, in recent work a temporal dimension was added to the super-resolution process in order to obtain 3D video [98].

4.1.3 Goals of the Proposed SDP Work

The present work focuses mainly on accuracy of active-passive super-resolution. The goal was to improve the depth accuracy at all points in the high-resolution image close to that achieved by the active depth sensor. While the proposed work is highly parallelizable, no attempts are made to achieve parallelization, or to add a temporal dimension.

The approach to this requires that two basic problems be solved:

1. One must handle calibration errors that cause mismatched color and depth edges. If ignored, calibration error significantly degrades edge definition in the upsampled result.

2. A new formulation is needed for constructing the cost-volume and applying the bilateral filter. As well as improving accuracy, this formulation needs to be better suited for uneven point distributions obtained from active sensors, compared to similar works in the literature.

The solution to the above two problems will be seen to enhance edge definition by removing blurring observed in much of the literature, which led to the SDP algorithm.

The following section describes the three SDP algorithm steps: projection, rejection, and propagation. Section 4.3 presents quantitative results from the Middlebury [78] benchmark, as well as various outdoor scenes. Section 4.4 discusses the results and evaluates each step of the algorithm to show the contributions made by the method proposed in this chapter.

4.2 Methodology

Figure 4-5 shows the dataflow diagram of the proposed super-resolution algorithm. The *projection step* uses the calibration data (e.g., relative sensor pose) to project laser measurements into the camera image coordinates: *x*, *y* pixel locations.

Calibration errors may cause laser measurements to be projected onto wrong surfaces in the camera image. Therefore, a *rejection step* is introduced to identify and remove the projected laser pixels that may be erroneous. This is achieved by removing laser measurements lying near object edges, and in regions that are occluded in the camera image.

Afterwards, the *propagation step* is used to construct a cost-volume and perform iterative bilateral filtering on it to generate the upsampled depth image. Our cost-volume is a discretized depth-probability function of the image, and therefore, the proposed upsampling method in this work is referred to as *Super-resolution on Depth Probability* (SDP).

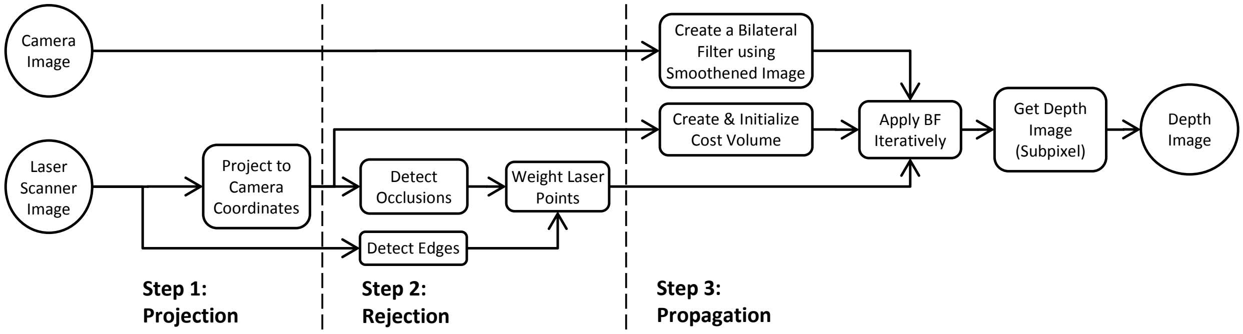


Figure 4-5: SDP's Dataflow Diagram

The dataflow diagram showing the three steps of the super-resolution algorithm proposed in this chapter.

4.2.1 Projection

Multi-sensor calibration techniques such as the Camera Calibration Toolbox for MATLAB [109] can be used to obtain the sensor alignments (i.e., extrinsic parameters), as well as the intrinsic sensor calibration parameters – See Appendix B. These parameters are used to assign each laser measurement to its corresponding camera pixel (i.e., projection).

Before projecting laser pixels into the camera image plane, the image is rectified to remove lens distortions. Laser measurements are then translated into the camera coordinate frame, which presents the same laser point-cloud as if the laser is located at the camera position instead:

$$l^{cam} = R_T \cdot l + T_T , \qquad l \in L \tag{4-7}$$

where $l:(x, y, z)$ is a laser measurement in the laser coordinate system, l^{cam} is a laser measurement in the camera coordinate frame, L is the laser point-cloud, and R_T and T_T are the rotational and translational components of the laser-camera alignment obtained through calibration. Subsequently, the laser points are projected onto the camera's 2D image plane; in other words, each laser measurement is assigned to a camera pixel:

$$\begin{aligned} x^I &= f \cdot x^{cam} \big/ z^{cam} + x_c \\ y^I &= f \cdot y^{cam} \big/ z^{cam} + y_c \\ z^I &= \sqrt{\left(x^{cam}\right)^2 + \left(y^{cam}\right)^2 + \left(z^{cam}\right)^2} \end{aligned} \qquad (4\text{-}8)$$

where x^{cam} , y^{cam} , z^{cam} are the x-y-z coordinates of laser points in the camera coordinate frame (l^{cam}), x^I and y^I are the *x*-*y* coordinates of each laser point in the image plane, z^I is the distance of the laser points to the camera (i.e., the pixel's *depth value*), f is the camera's focal length, and x_c , y_c are the coordinates of the camera's principal point. Figure 4-6 shows the projection results for a single laser scanline. The lines represent the *laser rays,* connecting the laser scanner positioned above the camera, to the points measured on object surfaces.

To verify the projection results, an infrared sensitive camera is used to view the laser spots (Figure 4-7.a). The *x*-*y* coordinates of the laser points in the image are calculated using the estimated calibration parameters – identified using cross marks in Figure 4-7. As seen, the cross marks appear in the center of the laser spots, which demonstrates a satisfactory calibration.

A single laser spot in Figure 4-7 covers hundreds of camera pixels. However, since the size and orientation of laser spots vary at different parts of the scene, for simplicity the projection step assigns each laser measurement to a single camera pixel only – the one at the center of the laser measurement spot. Later on and during the propagation step, the depth information from the center pixel is passed on to its neighbouring pixels. Alternatively, a laser measurement value could be assigned to all its corresponding pixels within the region occupied by the laser spot, as shown in in Figure 4-7.c. This work did not observe an advantage when using this alternative approach. Thus, in this work the laser measurement values are assigned to a single pixel at the center of their laser spot.

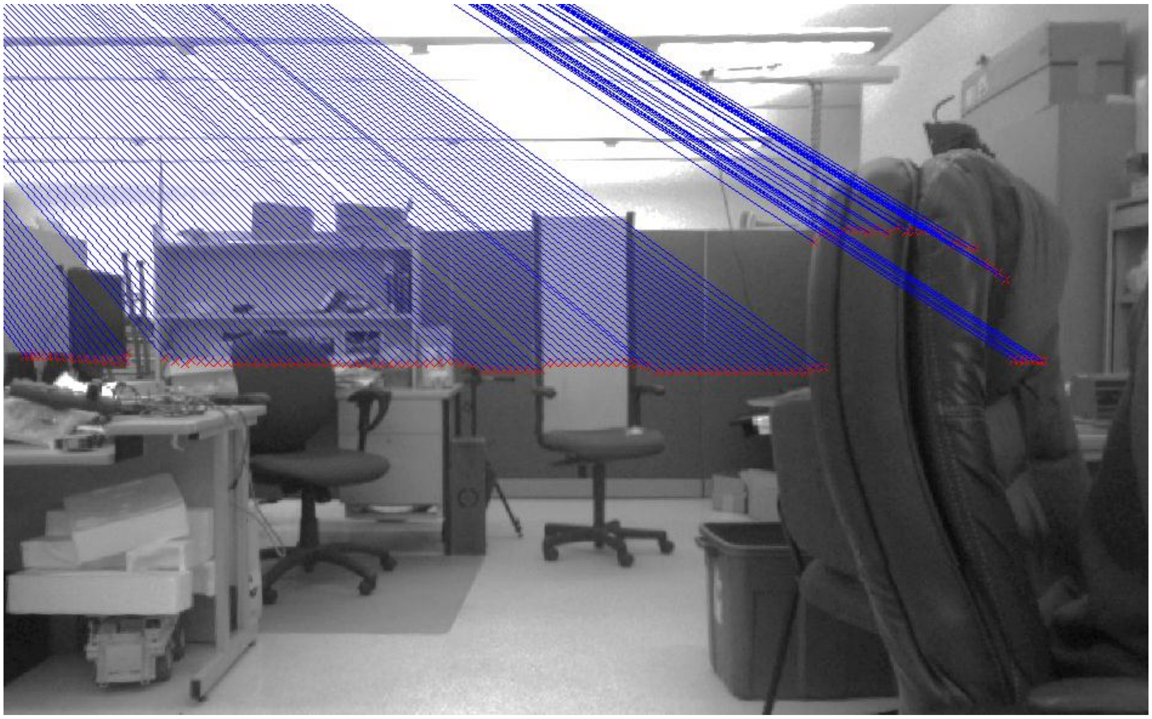
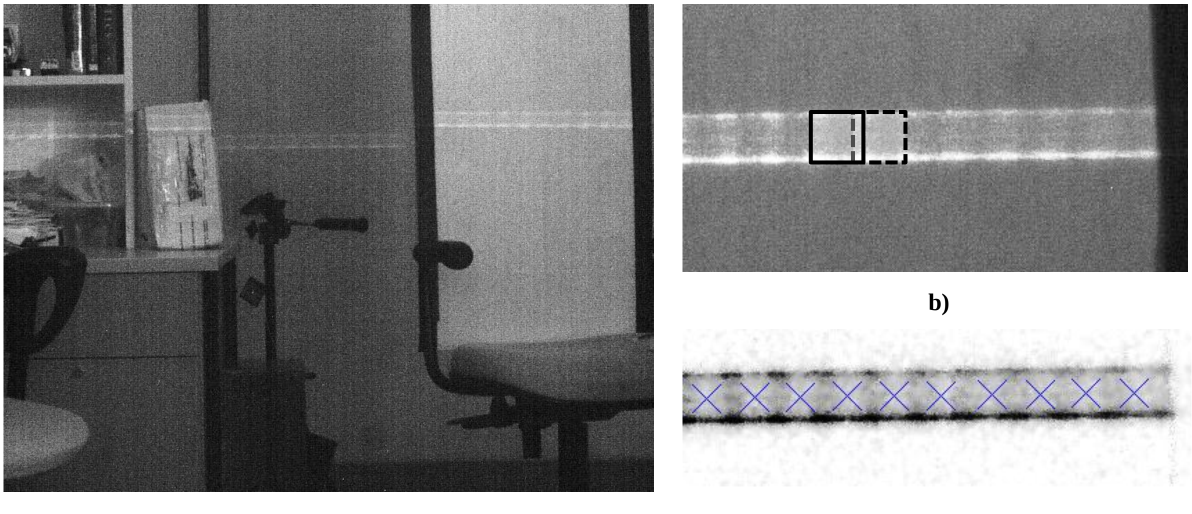


Figure 4-6: Project Results for a Laser Scanline

The projection results for a single laser scanline are displayed. The lines represent the laser rays from a scanner positioned above the camera.



b)

a)

c)

Figure 4-7: Laser Measurement Spots Viewed by Infrared-Sensitive Camera

a) The white horizontal stripe shows a series of overlapping laser scanner spots in the scene shown in the previous figure. b) Two adjacent laser spots are highlighted. The laser spots are square-shaped with a width of ~3cm. c) The laser stripe has been extracted from the above image. Cross marks identify the *x-y* coordinates of the center of each laser spot, calculated by projecting the laser measurements using the estimated calibration parameters.

4.2.2 Rejection

Due to lack of high-frequency information in laser depth images (i.e., the laser's low resolution and its inherent error in measuring edges), it is difficult to estimate the sensors extrinsic calibration parameters (the two sensors relative position and orientation) accurately. Thus, the projection step may result in laser points being assigned to incorrect camera pixels. This may cause the depth edges in the laser depth image and the color edges in the camera image to be misaligned. As discussed earlier, super-resolution algorithms rely on the assumption that edge and color discontinuities coincide. Thus, misalignment of the edges degrades the result by causing poor edge definition and introducing edge-blurring artefacts.

Inaccurate calibration is not the only source of invalid laser-camera pixel assignment. Occlusion is another significant issue. Since the sensors are in different positions, certain regions visible to the laser will be made hidden from the camera by another foreground surface. When laser measurements are projected onto the camera image, the measurements from the hidden regions are mistakenly projected onto the foreground surface. Detecting and removing the occluded measurements improves the upsampled depth image and removes sawtooth artefacts seen in Figure 4-8.

In this section, two algorithms are introduced to identify invalid laser pixels: foreground edge detection and occlusion detection. The invalid laser pixels are excluded from the upsampling process.

4.2.2.1 Foreground Edge Detection

When identifying edge measurements from the laser, it is important to minimize the rate of false-positives. High-accuracy information provided by the laser is a scarce and valuable resource, and if the edge detection algorithm is overly aggressive, it can result in entire high-frequency details such as narrow objects being lost. Therefore, it is important for the edge detection algorithm to define edges sharply and accurately. Canny edge detection [110] is used in this work because its edge definition is accurate and narrow. Edge detection is applied to the original laser depth image L – prior to projection (see Figure 4-9).

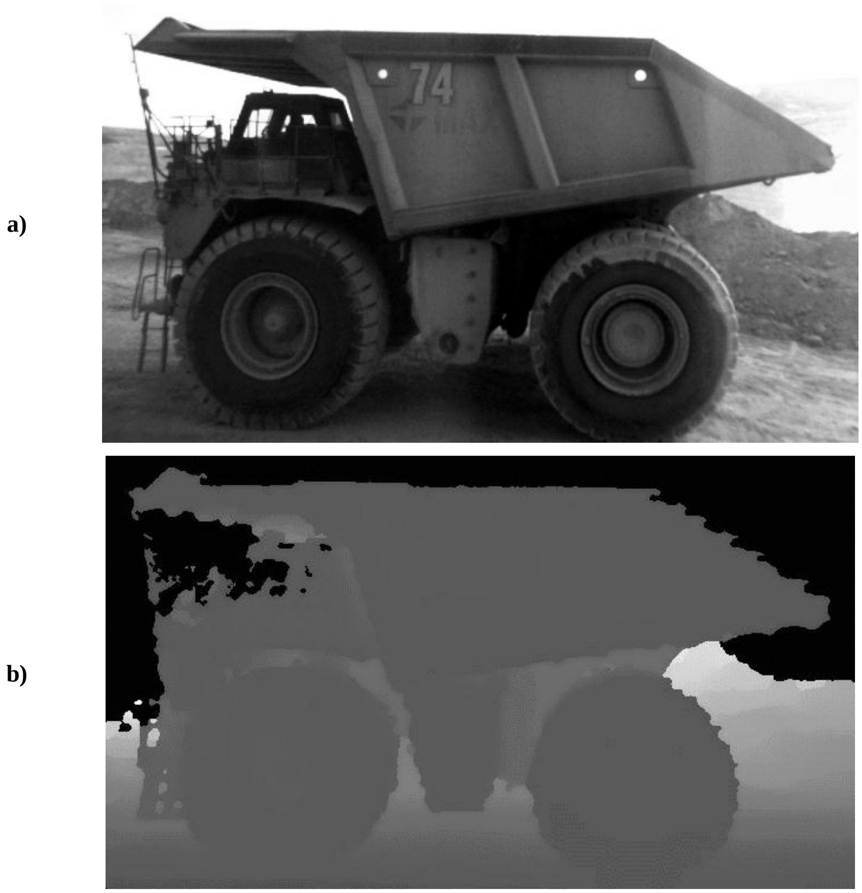


Figure 4-8: Saw-Tooth Artefacts

Sawtooth artefacts: a) the camera image of the scene, b) the upsampled depth image; note the sawtooth artefacts on the edges of the truck.

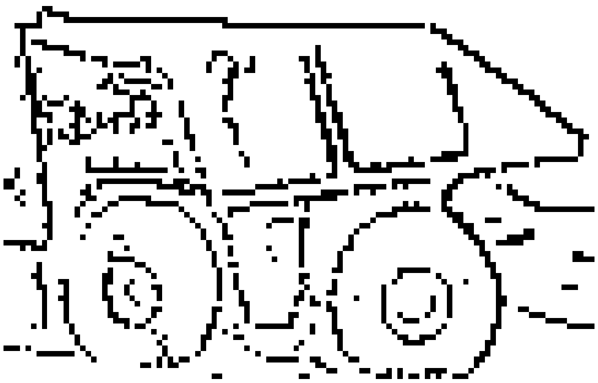


Figure 4-9: Canny Edge Detection of the Laser Depth Image in Figure 4-8

A depth edge is where a foreground and a background surface meet. With Canny edge detection[6], an edge is defined by labeling a pixel on either side. Thus, some edge pixels will belong to background surfaces, and others will be on foreground surfaces.

In camera images, changes in the camera position shift nearby points – points that belong to surfaces closer to the camera – further than distant points. Therefore, error in estimating the respective position of the two sensors (i.e., *calibration error*) causes larger misplacement in projected laser pixels of foreground (i.e., closer) surfaces. This means it is more likely for the sparse laser pixels from foreground objects to be placed on the wrong surfaces in the camera image. Thus, when removing edge pixels, it is better to remove the foreground side of an edge since its projection is more likely to be erroneous.

A simple algorithm can be applied to the Canny edge results, to identify and remove the edges measurements on foreground surfaces only. For every edge pixel *i* on surface *A*, a neighbouring pixel *k* exists that belongs to the adjacent surface *B*. The task is to first identify the neighbouring edge pixel, and then determine which of the two depth pixels, *i* or *k*, is on the foreground surface (i.e., has the smallest depth value). The first step can be performed by:

- Consider the four directions in the image plane: the vertical, horizontal, and the two diagonals (see Figure 4-10):
    - Calculate the depth gradient δ_{i-n} between pixel *i* and its neighbour *n*, and $\delta_{i-\hat{n}}$ between pixel *i* and the opposite side neighbour in that direction $\hat{n}$. Depth gradients are calculated as follows:

$$\delta_{i-j} = z_i^I - z_j^I \tag{4-9}$$

where *i* and *j* are indices of adjacent pixels in the depth image.

[6] Parameters used for Canny edge detection: σ = 1, threshold range = 5e^{-4} to 1e^{-3}.

- If the ratio between δ_{i-n} and $\delta_{i-\hat{n}}$ is greater than a threshold τ_e, then there is a depth edge between pixels *i* and *n*.

d$_1$	**v**	**d$_2$**
h	***i***	**h**
d$_2$	**v**	**d$_1$**

Figure 4-10: Directions Considered in Foreground Edge Detection

Four directions are shown in the image plane: horizontal (pixels *h-i-h*), vertical (pixels *v-i-v*) and the two diagonals horizontal (pixels *d$_1$-i-d$_1$* and pixels *d$_2$-i-d$_2$*).

Once all directions are considered, all neighbouring pixels of *i* that are on the adjacent surface *B* are identified. The second step to identify the foreground depth pixels is as follows:

- If any of the identified neighbours have a depth value less than that of pixel *i*, then they are marked as a foreground pixel.
- Otherwise, if *i* has the smallest depth value between them all, then pixel *i* itself is a foreground depth pixel.

The above algorithm can be summarized in the following equation:

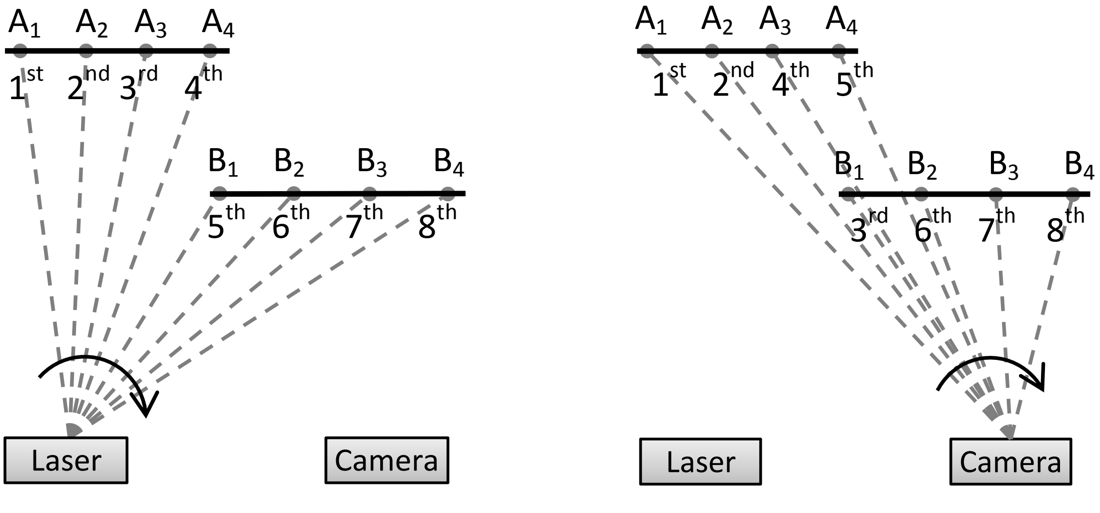
$$\widehat{E} = \begin{cases} \dfrac{\delta_{i-n}}{\left|\delta_{i-\hat{n}}\right|} > \tau_e \rightarrow n \\ \delta_{i-n} < 0 \rightarrow i \end{cases} \quad i \in E, \ \forall n \in N \qquad (4\text{-}10)$$

where E is the set of all Canny edge pixels, $\widehat{E}$ is the set of foreground edge pixels, *N* is the set of pixel *i*'s neighbours, δ_{i-n} is the difference between the depth at pixel i and its neighbouring pixel n, and $\hat{n}$ represents a neighbour on the opposite side of the pixel n (i.e., n and $\hat{n}$ are one of the matching pixel pairs in Figure 4-10). The threshold τ_e is set to 1.5.

As we will show in Section 4.4.3, foreground edge removal substantially improves edge definition in the upsampled image.

4.2.2.2 Occlusion Detection

An efficient way of detecting occluded measurements is by comparing the order they appear in a laser scanline before and after projection onto the camera image plane (4-8). Figure 4-11.a shows the laser measured points A_1-A_4 on one surface, and B_1-B_4 on another surface. In Figure 4-11.b, points A_3, A_4, and B_1, are "out of order." That is because their order of appearance in the laser's view is different than the order of appearance of the same laser measured points when seen in the camera's view (i.e., after projection; see Table 4-1). Note that points A_3 and A_4 are occluded in the camera's view.



a) b)

Figure 4-11: Ordering Constraint for Occlusion Detection

An ordering constraint can be used for detecting occlusions in the projected laser points. a) Laser measurements of two surfaces (A and B) are shown. The order of appearance of the points is written underneath them (from left the right). b) The same points, when observed from the camera's view. Some appear in a different order (A_3, A_4 and B_1). Note that A_3 and A_4 are occluded.

Table 4-1: The Order of Appearance of Points in Figure 4-11

	A_1	A_2	A_3	A_4	B_1	B_2	B_3	B_4
Laser	1st	2nd	3rd	4th	5th	6th	7th	8th
Camera	1st	2nd	4th	5th	3rd	6th	7th	8th

The order of appearance, from left to right, of the points in Figure 4-11. Points A_3, A_4, and B_1, are highlighted as "out of order," given that their order of appearance in camera's view is different than in laser's view.

In this work, an ordering constraint is proposed for identifying occluded points. The following pseudo-code demonstrates the proposed occlusion detection method:

1. Compare the order of appearance of the laser points in both sensor views, and label the points that appear in different positions as "out of order" (similar to Table 4-1).
2. Iteratively remove the most distant laser pixels (highest z^l value) among the out-of-order points.
3. Stop the iterations when the order of appearance of the points is identical in both sensors. In the above example, points A_3 and A_4 would be removed before the iterations stop. The removed points are *occluded* in the camera view.

In a naïve implementation, occlusion detection is an $O(n^2)$ problem, since every pair of points must be examined for possibly occluding one another. On the other hand, the above occlusion detection method using ordering constraint can be implemented in *O(n log(n))*.

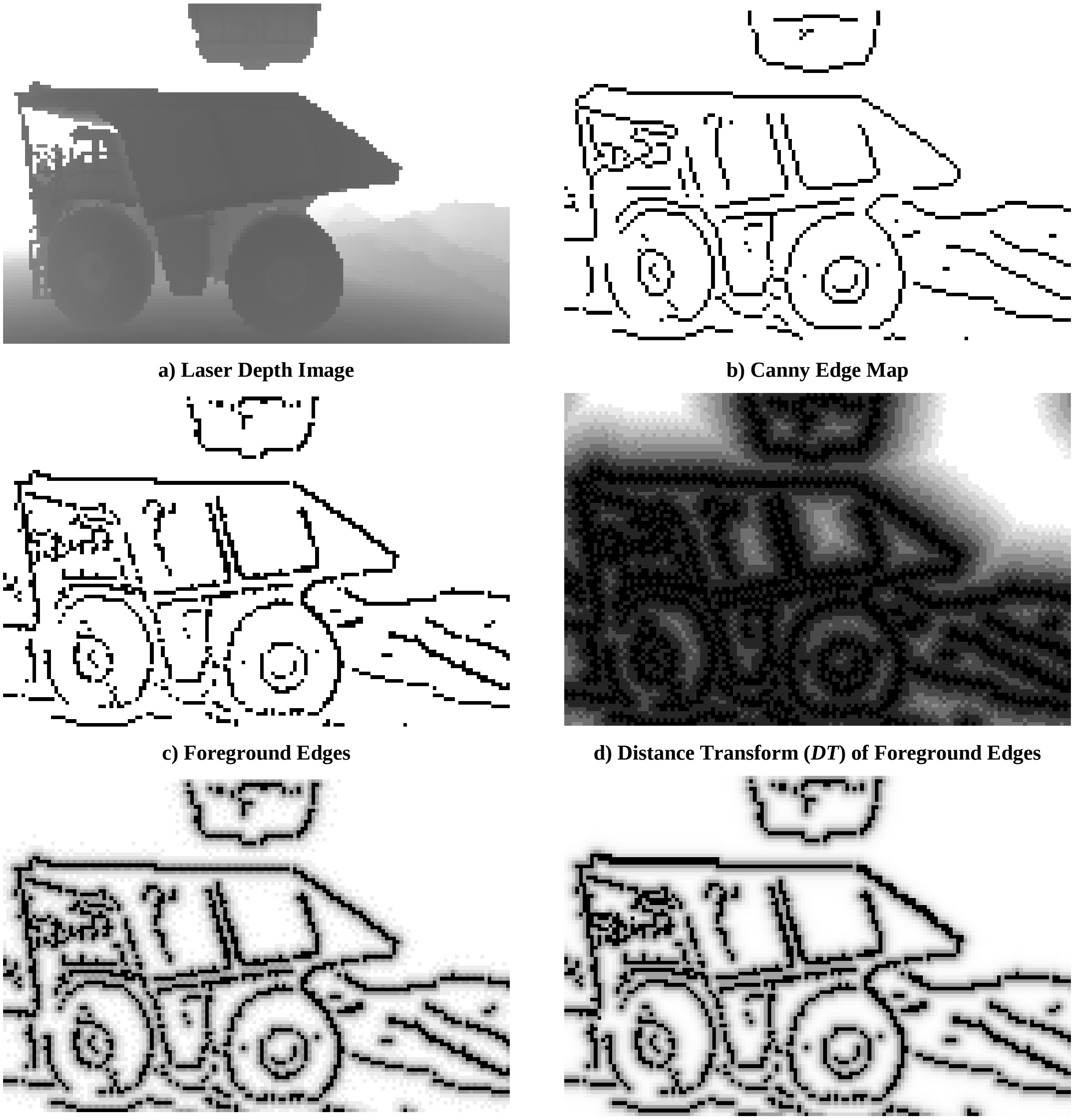
4.2.2.3 Weighting the Laser Measurements

Rejection of unreliable laser measurements is achieved by weighting the edge map and removing occluded pixels using:

$$\begin{aligned} W &= 1 - e^{\frac{-DT}{\lambda_e}} \\ W_i &= 0,\ i \in occluded \end{aligned} \tag{4-11}$$

where *DT* is a *Distance Transformation* [111] of the edge map – a binary image in which foreground edge pixels found in Section 4.2.2.1 are marked with 1's. Equation (4-11) removes all edge and occluded pixels by setting their weights to 0's. Furthermore, laser pixels that are in proximity of an edge can be weighted down by adjusting λ_e – the higher the value, the lower the weights.

Figure 4-12 demonstrates the SDP rejection step by showing the intermediate results. The depth image created using a laser point-cloud is shown in (a). Canny edge detection is applied to the laser depth image and the result is shown in (b). Edge pixels identified by Canny may belong to either side of an edge – its foreground surface or its background surface. Thus, in (c) the foreground edge pixels are found to replace the Canny edge pixels that belonged to the background surfaces – see Equation (4-10). The distance transform (*DT*) of the foreground edge map is shown in (d). Next, the first line of Equation (4-11) is applied which weights each laser depth pixel using the *DT* of the foreground edge map. The result is *W* shown in (e). Finally, the occluded laser pixels detected in Section 4.2.2.2 are also rejected (i.e., their weights in *W* are set to zero), as shown in (f). This is done by the second line of Equation (4-11).



a) Laser Depth Image

b) Canny Edge Map

c) Foreground Edges

d) Distance Transform (*DT*) of Foreground Edges

e) Weights (*W*) Before Occlusion Removal

f) Weights (*W*) After Occlusion Removal

Figure 4-12: SDP Rejection Step-by-Step Results

a) A laser depth image is shown where brighter pixels represent farther distance. b) A binary image representing the Canny edge detection result. c) A binary image of foreground edges; this is once Equation (4-10) is applied to the Canny edge map. d) Distance transformation is applied to the foreground edge map. This is *DT* in Equation (4-11). The brighter pixels are farther away from any edge pixel. e) The weights calculated using the first step of Equation (4-11). The grayscale represents weights 0 to 1 using colors black to white respectively. Thus, black pixels represent the 'rejected' pixels. f) The final image shows the result of rejecting occluded pixels as well (i.e., coloring them black), which is the second step of Equation (4-11). This image is the final weights (*W*) assigned to each laser depth pixel.

4.2.3 Propagation

After projection, a subset of image pixels has an associated laser measurement. The rejection step calculated a weight for each of those laser measurements. In the propagation step, these *laser pixels* are used to estimate depth at pixels without laser data.

Similar to the Yang algorithm, in this work a bilateral filter is applied to a cost-volume. Other works in the literature apply the filter to the depth image directly instead [32], [34]. Filtering the depth image causes blurring of the edges, which cost-volume filtering avoids.

4.2.3.1 Cost-Volume

For every pixel in the upsampled image, a normal distribution is used to represent probability as a function of depth. This *depth-probability* represents how likely it is for a pixel to be at any given depth. In the absence of more advanced error models for laser sensors, the utilized normal distribution best represents the sensor error reported by the manufacturer.

At a given depth, bilateral filtering [32], [33] has been used to pass the depth-probability value of one pixel to its neighbours with similar texture. Thus, laser pixels can propagate their depth-probability distribution to neighbouring regions.

The depth-probability distributions are discretized so that the filter can be applied to each depth slice across the image. Through experimentation, the number of discrete levels can be suitably selected to reduce computation. The discretization of the depth-probabilities constructs a cost-volume similar to the one in [30]: for each image pixel, a vector of size Z (the number of depth slices) is used to define its depth probability distribution. This creates a 3D matrix in which the x-y dimensions represent pixel locations and the z dimension represents the depth probability vector. This matrix is known as the cost-volume, and initially, all its elements are set to zero.

For each laser pixel *i*, the laser depth value in the image plane *I* is represented by z_i^I. In the pixel's depth probability vector, the two elements whose depths are closest to z_i^I, are $\overset{\frown}{d}_w$ and $\overset{\frown}{d}_{w+1}$ (see Figure 4-13). In the SDP algorithm, a quadratic curve is used to set the values of $\overset{\frown}{d}_w$ and $\overset{\frown}{d}_{w+1}$, so that the curve reaches its maximum value at z_i^I, as seen in Figure 4-13.

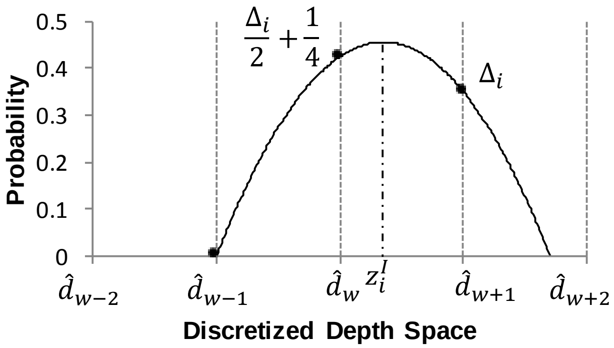


Figure 4-13: Cost-volume Initialization Using Quadratic Estimation

Cost-volume initialization is done by quadratic estimation of the probability values at discrete disparity levels adjacent to the laser-measured depth.

SDP uses the following equations to calculate the adjacent values that define the quadratic curve:

$$\Delta_i = \left| z_i^I - round\ (z_i^I) \right| \qquad (4\text{-}12)$$

Δ_i is defined as the absolute difference of pixel *i*'s laser depth and its closest discrete value. The cost at the closest discrete value is set to $(2 \cdot \Delta_i + 1)/4$, and the cost at the farther one is set to Δ_i:

$$C_i(d) = \begin{cases} \frac{\Delta_i}{2} + \frac{1}{4}, & d = \overset{\frown}{d}_c \\ \Delta_i, & d = \overset{\frown}{d}_f \\ 0, & others \end{cases} \qquad \{i : i \in L^C\} \quad (4\text{-}13)$$

where *C* is the cost-volume, $C_i(d)$ is the cost at depth *d* for pixel *i*.

The use of a quadratic equation ensures that the resolution of the laser depth measurements is not lost due to discretization. Subpixel resolution estimation is performed at the end of the propagation step to effectively reverse the discretization process and maintain the original laser depth resolution. The subpixel resolution step – Equation (4-17) – reconstructs the quadratic curve from three points in the discretized probability space. The use of a quadratic equation makes that process easy to implement with low computational cost. It is possible that a Gaussian distribution may better represent depth probability at each pixel. But due to the simplicity in use of the quadratic equation and the fact that it has been used effectively in the literature for subpixel resolution [30], this work also uses the quadratic equation.

4.2.3.2 Bilateral Filtering of the Cost Volume

Using Equation (4-5), a bilateral filter is constructed: unique kernels are computed for each pixel using the camera image. In this work the CIELAB image space is used in $\widehat{I}$. It is suggested that measuring color difference using CIELAB color-space correlates strongly with human perception and color discrimination [24]. A minor error reduction of approximately 3% was observed when using CIELAB rather than RGB.

The bilateral filter is then applied to every slice of the cost-volume individually:

$$C'_i = \frac{\sum_{k\in\Omega} C_k \cdot F_{i,k} \cdot W_k}{\sum_{k\in\Omega} F_{i,k} \cdot W_k} \tag{4-14}$$

where C' is the filtered cost-volume and Ω represents the set of all pixels within the filter support window of pixel *i*.

Using a forgetting-factor of $\alpha = 0.8$, the new cost-volume (C^{t+1}) is calculated based on C' and the previous cost-volume (C^t):

$$C_i^{t+1} = \begin{cases} (1-\alpha)\cdot C' + \alpha \cdot C^t & i \notin L \\ C_i^0 & i \in L \end{cases} \tag{4-15}$$

Note that for reliable laser pixels ($w > 0.5$), the cost vectors stay constant.

The iterative application of bilateral filtering propagates the information from pixels with high-confidence laser data to those with less, while ensuring the information is not passed beyond object edges. To observe this process, a *dataflow image* can be constructed by applying the same iterative bilateral filter to a single binary image in which the reliable laser pixels ($w > 0.5$) are set to 1 (white color). The iterative filtering of this binary image propagates the white color of the laser pixels into the neighbouring pixels. Thus, the result of this filtering process identifies the image pixels that receive significant information from neighbouring pixels by giving them light colors. In an ideal algorithm, it is expected that the pixels on object edges receive considerably less information from neighbours, and thus they must have darker colors. Figure 4-14 shows the dataflow image of the proposed algorithm in this work. It can be observed that the edge pixels are identified successfully and distinctively (dark pixels), and thus, the image shows a visually accurate object segmentation of the scene. This example shows that the algorithm in this work is able to segment surfaces, and propagate information within surfaces, not across them.

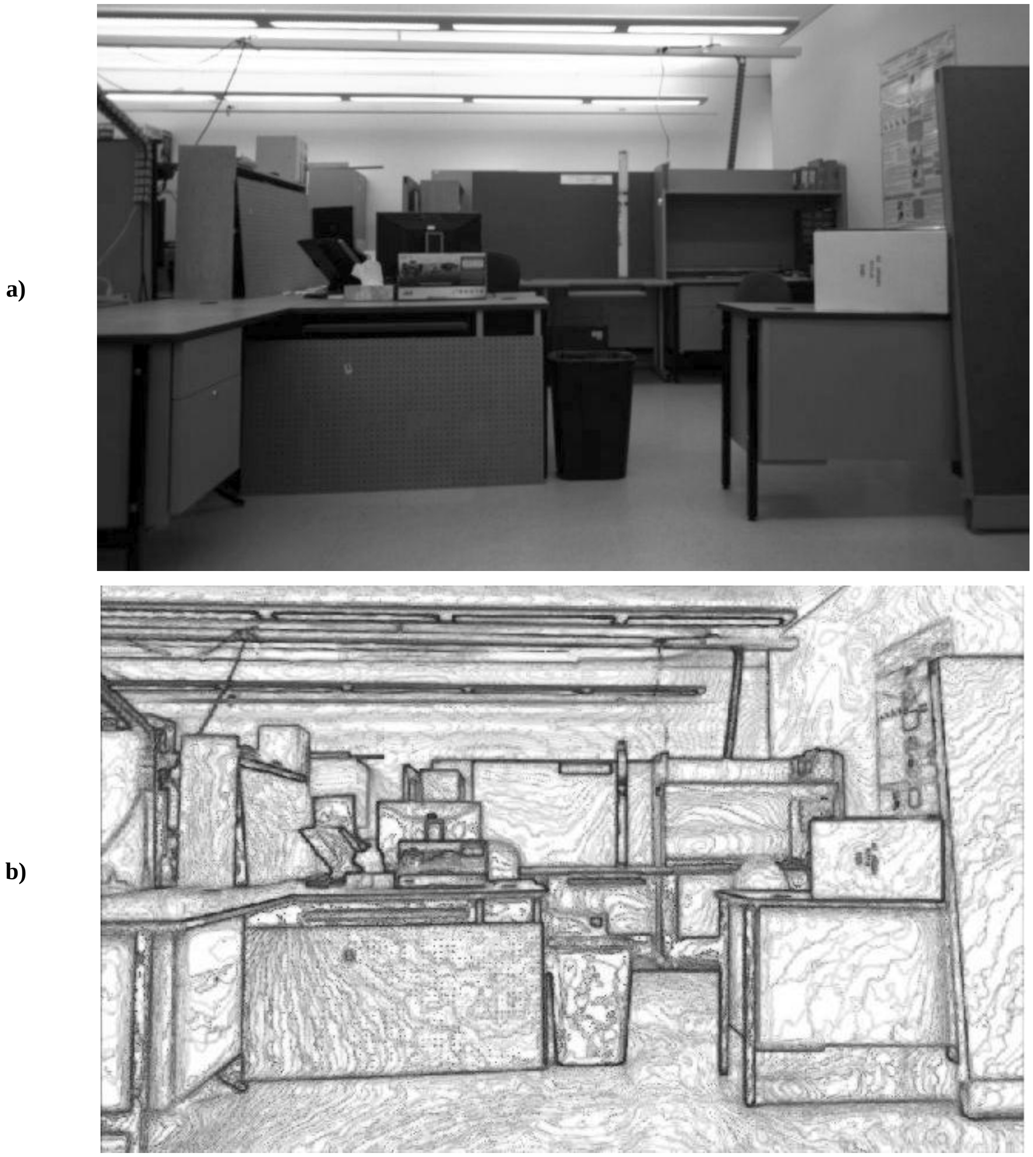


Figure 4-14: Dataflow of the SDP algorithm

a) The camera image of the scene. b) The dataflow image: whiter pixels show an easy flow of information, whereas black pixels show resistance to data flow (which appear at edges). The edge pixels are distinctively identified with dark pixels. This means that SDP segments the scene very well, and does not propagate information across object edges.

4.2.3.3 Subpixel Resolution

Once the cost-volume is refined through filtering, the upsampled depth image is created based on the new cost-volume using a winner-takes-all method: at each pixel, the depth with the highest probability is selected:

$$v_i = \arg\max_d \left(C_i(d)\right) \qquad (4\text{-}16)$$

where v_i is the depth candidate with highest probability.

In order to improve depth resolution and remove staircase artefacts introduced by depth discretization, a quadratic subpixel resolution equation is used, similar to [30]:

$$D_i = v_i - \frac{C_i(v_i+1) - C_i(v_i-1)}{2\left(C_i(v_i+1) + C_i(v_i-1) - 2\,C_i(v_i)\right)} \qquad (4\text{-}17)$$

where *D* is the upsampled depth image. A median filter can then be applied to the depth image to remove the occasional noisy estimates.

To summarize, the following pseudo-code describes the SDP algorithm:

- ***Projection:*** Project the laser measurements onto the camera plane, using Equations (4-7) and (4-8).
- ***Rejection:***
    - Identify the occluded pixels using the process in Section 4.2.2.2.
    - Perform Canny edge detection on the laser depth image to find the edge pixels.
    - Identify foreground edge pixels using Equation (4-10).
    - Based on the identified occluded and foreground pixels, calculate the weights of laser pixels using Equation (4-11).
    - ***Propagation:***
        - Calculate a cost-volume based on laser depth pixels, using Equation (4-13).
        - Calculate a bilateral filter based on the camera image, using Equation (4-5).
        - Iteratively apply the bilateral filter to the cost-volume, using Equation (4-14), until convergence or maximum iterations is reached.

- Use the subpixel resolution method – Equations (4-16) and (4-17) – to estimate SDP's resulting depth image based on the filtered cost-volume.

4.3 Results

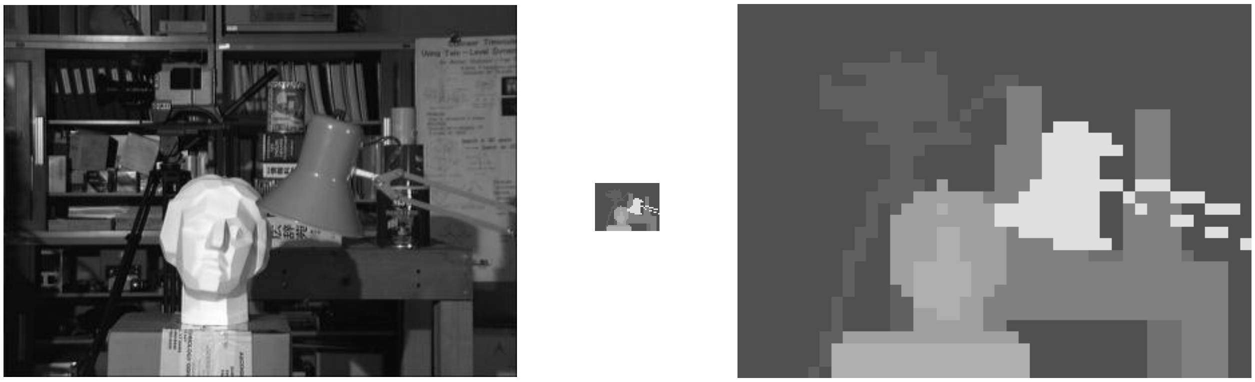
In [34], Kopf et al.'s JBU [32] is shown to provide similar results to the MRF method in [31]. Andreasson et al. [35] also compare a number of algorithms to the MRF, and at best obtain similar results. Chan et al.'s NAFDU [34] produces slightly better results than the MRF. The Yang algorithm, on the other hand, is a considerable improvement. Therefore, the MRF method and the Yang algorithm are used in this work to compare the upsampling results against. Two sets of experiments were performed: first using the Middlebury benchmark [78], and then using scenes captured by a laser scanner and a camera.

4.3.1 Middlebury Benchmark

Middlebury stereo datasets have been used extensively in the literature as a benchmark for stereo algorithms [22], [78]. Since a number of active-passive super-resolution methods report their results on the Middlebury dataset, this work also reports experimental results on Middlebury to compare the relative accuracies of the three methods. To use it for depth super-resolution, the ground-truth images are downsampled by a factor of 2, 4 or 8 in each image dimension. The resulting downsampled image imitates an accurate low-resolution depth sensor such as a TOF camera. Using that along with the reference camera image from the stereo pair, depth super-resolution can be performed (see Figure 4-15). The results are compared against the original ground-truth image in the benchmark to evaluate the super-resolution algorithm.

Since both input images are in the same camera coordinates, SDP's projection and rejection steps are not required (i.e., no edge detection and occlusion removal). Thus, our Middlebury results only compare our filtering formulation (*propagation step*) to the other methods.

Figure 4-15 shows the reference camera image of the Middlebury's Tsukuba scene, as well as the ground-truth image downsampled by a factor of 8 in each dimension. The two images are used as inputs to our SDP algorithm to generate the upsampled image in Figure 4-16.



a) b) c)

Figure 4-15: Inputs of the Super-Resolution Algorithm

Tsukuba dataset from the Middlebury dataset. a) The camera image. b) The ground-truth image downsampled by a factor of 8x8. c) The ground-truth image enlarged. The camera image and the downsampled ground-truth image are inputs to our super-resolution algorithm.

Figure 4-17 compares the errors obtained by the SDP algorithm, the Yang method and the MRF algorithm. The reported errors represent the percentage of "bad pixels" [78] in the resulting depth images. *Bad pixels* are those that have a distance error greater than a threshold. The reported results in this work have a threshold of 1 disparity pixel. For the MRF and Yang methods, the errors reported in [30] have been used. For SDP, the following parameters have been used: 15 filtering iterations with a filter window size of 11x11, $\lambda_s = 15$, and $\lambda_c = 1$. The parameters have been selected by experimentation. The effect of parameter variation is shown in Figure 4-18, Figure 4-19 and Figure 4-20, which display a subset of the experiments performed for parameter selection. In each figure, the error performance of SDP is plotted against the variation in one of the three bilateral filter parameters (kernel size, λ_c and λ_s), while the other parameters are kept constant.

As shown in Figure 4-17, SDP presents significant improvement in upsampling error. Particularly, SDP shows better performance at higher upsampling rates. For the 8x8 upsampled results, on average SDP improves the results of Yang and MRF by factors of 4.0 and 6.8 respectively.

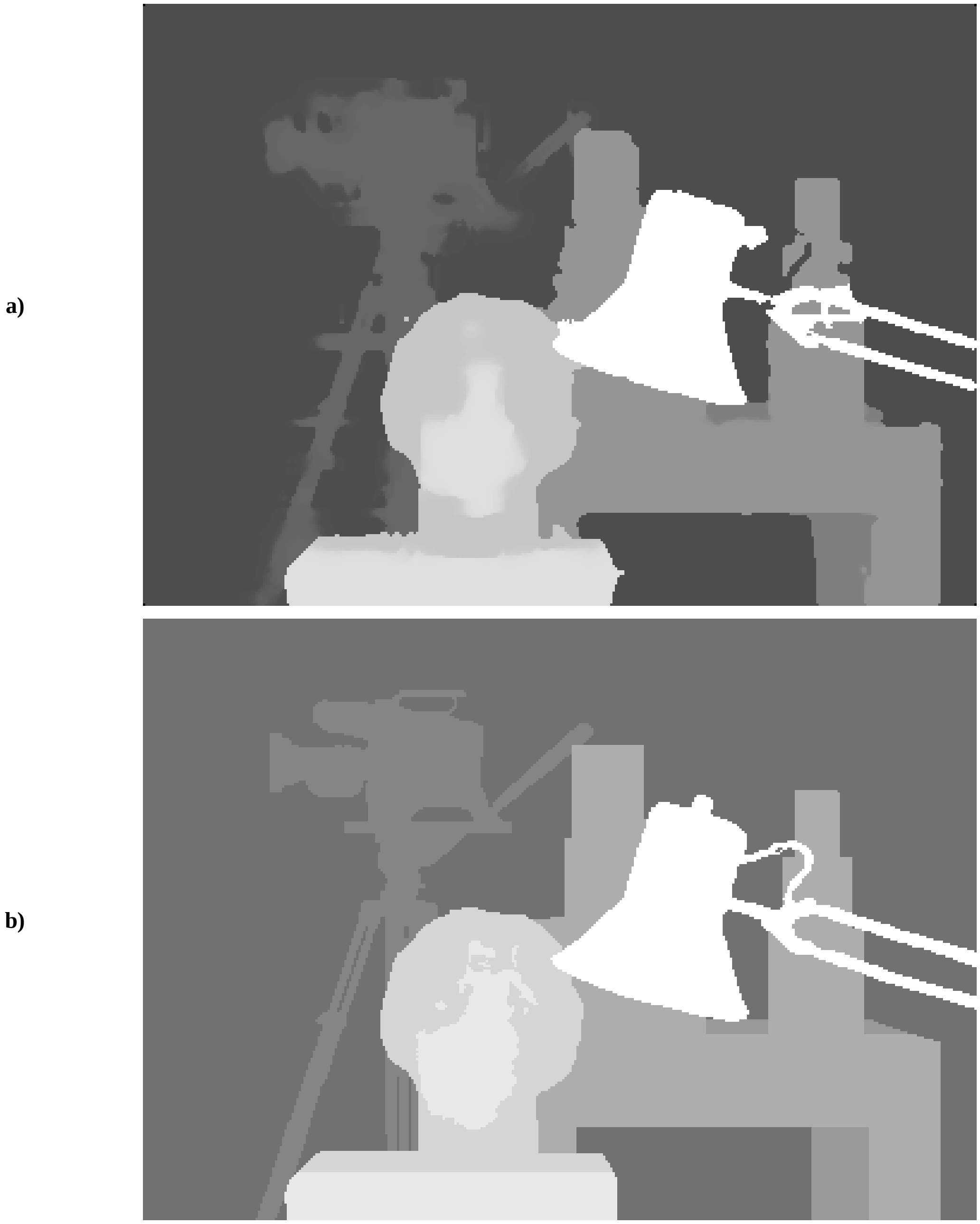
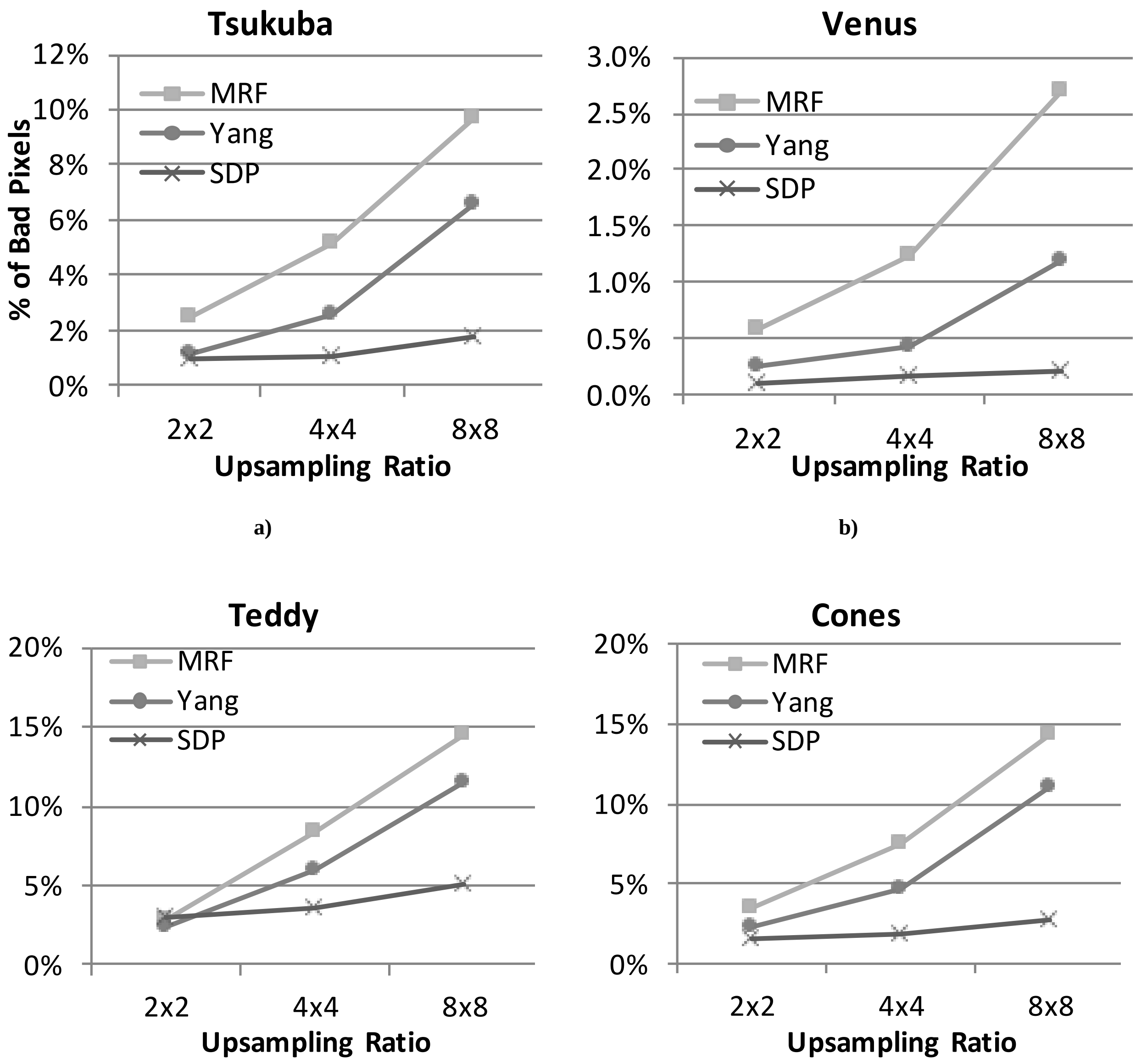


Figure 4-16: Upsampling Result of the SDP Algorithm Applied to the Tsukuba Image

a) The results of upsampling the Tsukuba images in Figure 4-15 using the SDP method. b) The ground-truth image is shown in (b).



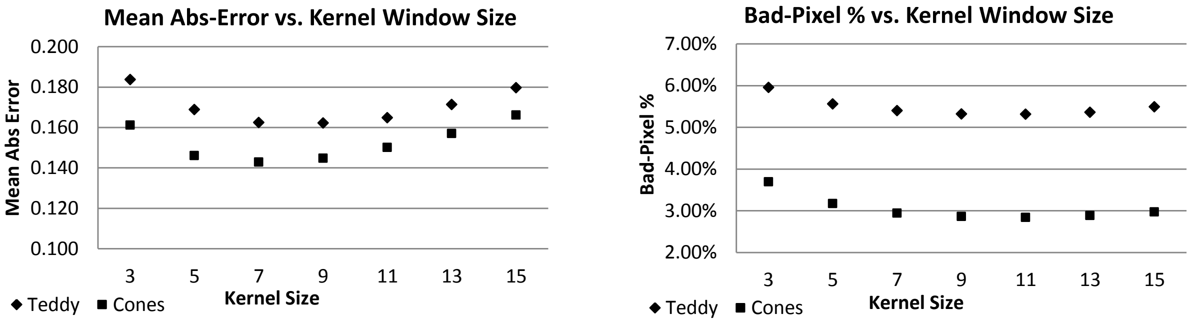
a)

b)

c)

d)

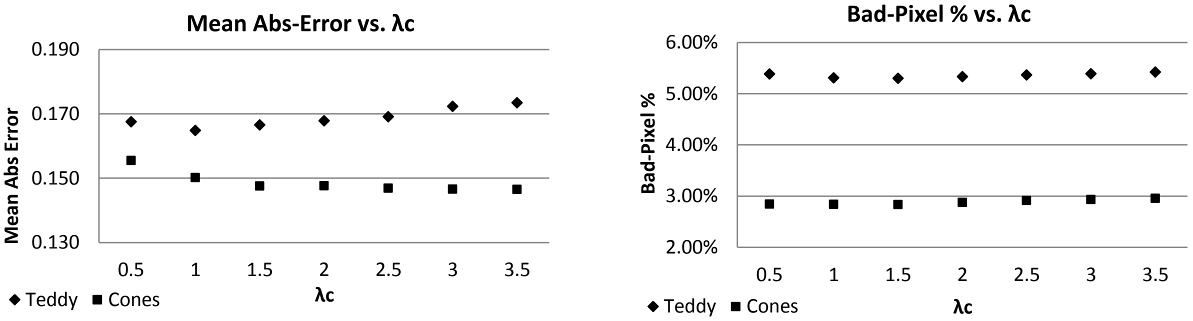
Figure 4-17: Results of the Super-Resolution Algorithms Applied to the Middlebury Benchmark Middlebury Benchmark: SDP vs. Yang vs. MRF [31]. The x-axis is the upsampling ratio, and the y-axis is the percentage of bad pixels (error > 1 disparity) as defined in [22]. a) Tsukuba dataset. b) Venus dataset. c) Teddy dataset. c) Cones dataset.



a) **b)**

Figure 4-18: Error Performance versus Kernel Size Variation in Teddy and Cones

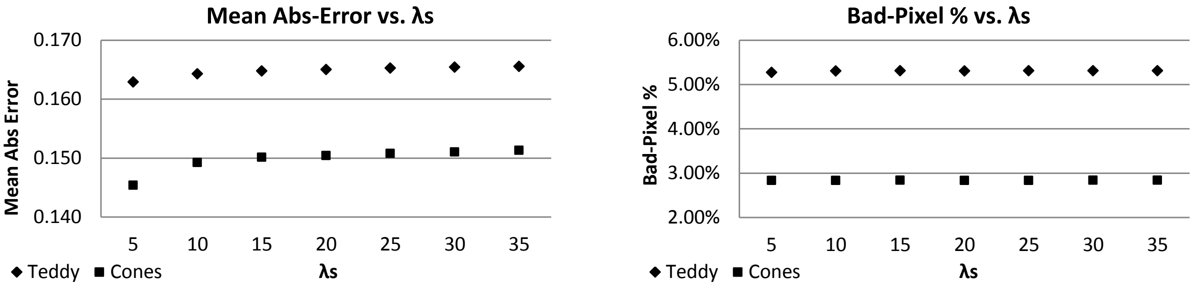
Mean absolute error in depth values (a) and bad-pixel percentage (b) are reported for Teddy and Cones, as the size of the bilateral-filter's kernel window is changed. Other parameters remain constant: λ_s=15 and λ_c=1.



a) **b)**

Figure 4-19: Error Performance versus λ_c Variation in Teddy and Cones

Mean absolute error in depth values (a) and bad-pixel percentage (b) are reported for Teddy and Cones, as the value of bilateral filter's λ_c – color sensitivity parameter in Equation (4-5) – is changed. kernel window size = 11x11 and λ_s=15.



a) **b)**

Figure 4-20: Error Performance versus λ_s Variation in Teddy and Cones

Mean absolute error in depth values (a) and bad-pixel percentage (b) are reported for Teddy and Cones, as the value of bilateral filter's λ_s – spatial sensitivity parameter in Equation (4-5) – is changed. kernel window size = 11x11 and λ_c=1.

4.3.2 Laser Scanner Data

4.3.2.1 Setup

To test our algorithm with real sensor data, a 3D swinging laser scanner (manufactured by RoPro Design, using a SICK LMS 291) and a camera (Point Grey Research's Bumblebee XB3) were utilized to capture a number of indoor and outdoor scenes (see Figure 4-21). Scenes were captured on the campus of the University of British Columbia, and at a mine site. The 3D laser scanner has a scanning angular range of 180° horizontally, 60° vertically, a horizontal step-size of 0.5° and an average vertical step-size of 1.2°. The laser scanner captures 13k points/sec, and the camera provides color images with 800x600 resolution. It is recommended for the passive camera to have an infrared filter to make the laser light invisible. If visible, the active sensor's light can degrade SDP's performance by introducing texture edges that do not correlate to actual depth edges. The Bumblebee XB3 camera used in this work was not infrared sensitive.

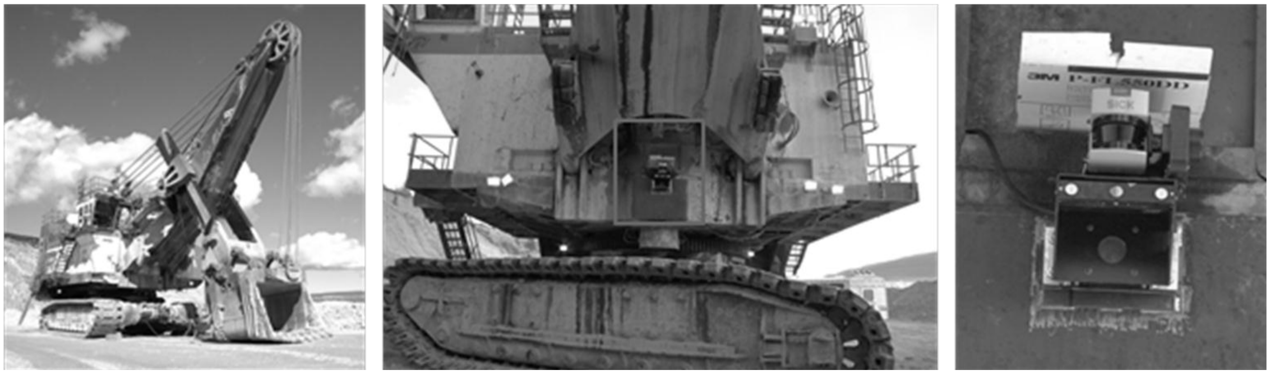


Figure 4-21: Laser Scanner and Camera Setup on a Mining Rope Shovel

A mining rope shovel is shown on the left. The sensor suit (a swinging laser scanner and a camera) is mounted underneath the boom, as shown in the center image. The sensor suit can be seen on the right image, which is a zoomed in image of the highlighted box in the center image.

To calibrate the two sensors, an image was constructed using the laser scanner's reflectivity readings for each depth measurement. The laser reflectivity image along with the camera image were then used in a calibration algorithm similar to [109] (see Appendix B).

At each scene, the laser scanner first captured a full 3D scan of the scene in an approximately one second sweep. This *sparse laser depth image* is used as the input to the super-resolution algorithm. Since in this experiment the laser scanner requires one second to capture the entire scene, the scenes must be static – no moving objects during the laser sweep. To handle dynamic scenes, faster depth image acquisition is required either by using better laser sensors, or utilizing temporal information (see Section 5.2 for a discussion about dynamic scene capturing).

A second scan of the scene was captured afterward at a slow sweep rate, in order to generate a *dense laser depth image* of the scene. The dense laser depth image provides laser measurements for more image pixels, and therefore can be used to evaluate the performance of the upsampling result.

To construct a ground-truth image, the dense laser depth image is projected onto the camera plane (see Section 4.2.1). The laser measurements that are deemed unreliable (e.g., the edge measurements) are then removed (see Section 4.2.2), since a ground-truth image needs to be an accurate representation of the scene. On average, ground-truth images that are constructed using dense depth laser images contain more than 5 times the number of laser pixels of a sparse laser depth image. Therefore, the proposed ground-truth images can evaluate the performance of the upsampling algorithm on the additional pixels for which they have laser measurement values.

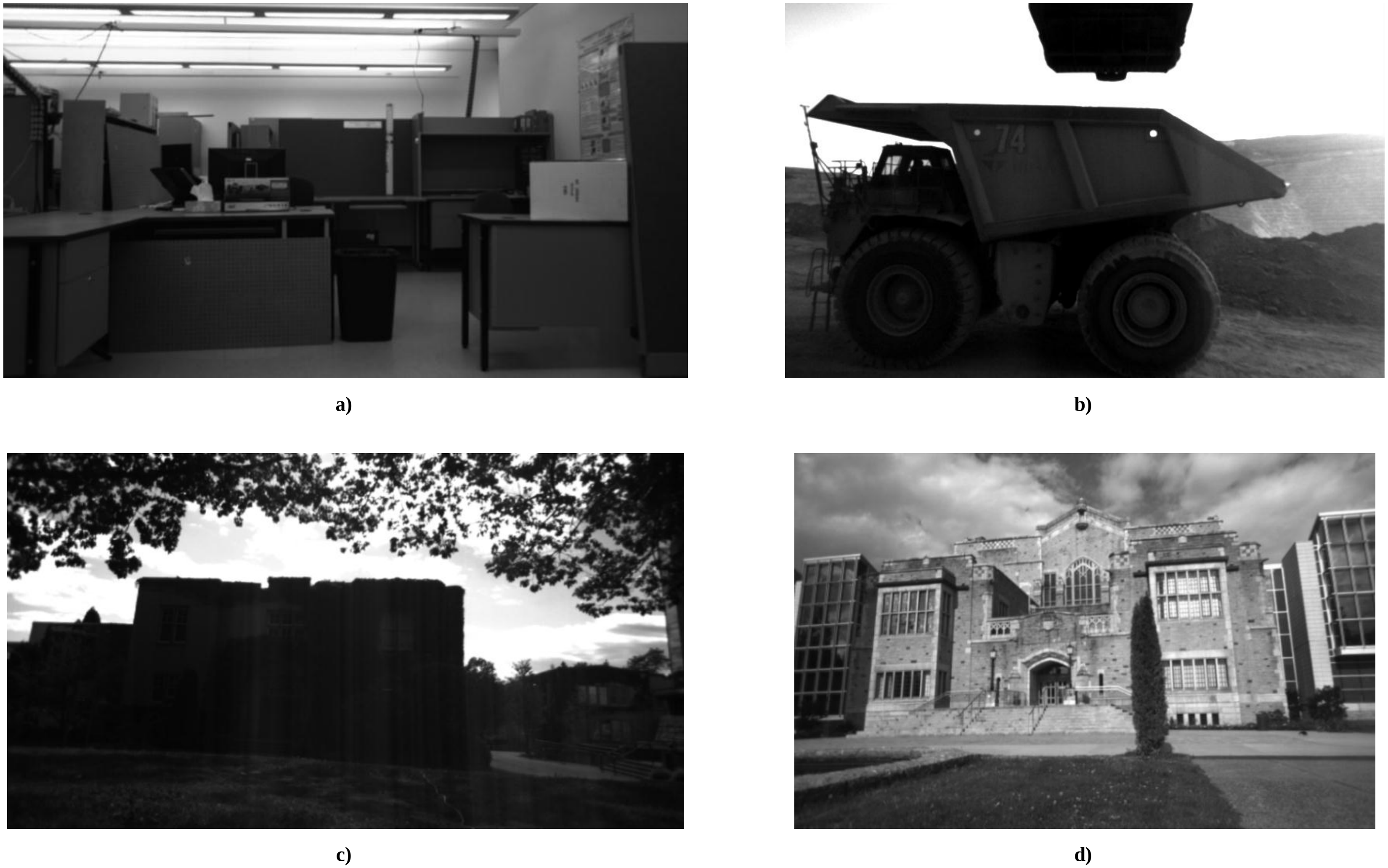
An upsampled image can thus be evaluated by comparing it against the laser pixels in the ground-truth image. The error in the upsampled image is calculated using the average of the absolute depth difference between the laser pixels in the ground-truth image and their corresponding pixels in the upsampled image. When comparing the sparse laser depth image – before upsampling – against the ground-truth image, a mean error of 3.0cm to 5.0cm is observed, depending on the scene. This is due to the mean systematic error of 5.0cm in the laser scanner used here.

Four of the captured laser scanned scenes are shown in Figure 4-22:

- *Office* (Figure 4-22.a) is a complex indoor scene with many high frequency details. The maximum range is 10m.
- *Mine* (Figure 4-22.b) is a haul-truck being loaded by a rope-shovel. The bucket can be seen above the truck. The camera image is dark due to sun light exposure. Figure 4-8.a is the same image, adjusted to appear brighter and clearer. The maximum range of this scene is 40m.
- *Building* (Figure 4-22.c) is for examining the super-resolution performance when dealing with poor camera images. The building structure and the bushes in front of it appear very dark and are hardly distinguishable. The camera image is heavily distorted with sunray artefacts. The maximum range of this scene is 30m.
- *Library* (Figure 4-22.d) is a long-range image of a complex multi-surface building structure with windows and reflective elements. This scene has a depth range of 55m.

For comparison, the MRF and the Yang algorithm have also been implemented. For the MRF and Yang methods, algorithm parameters were configured to obtain the best results in the laser scenes. This required running hundreds of experiments while varying one parameter at a time. No coupling of parameters was immediately observable, and as seen in Figure 4-19 and Figure 4-20, the output was relatively insensitive to parameter variations. Thus, the parameter combination that worked best in this experimentation was selected for each method.

The parameters for the MRF algorithm are set to $c = 0.1538$ and $k = 500$ [31]. Note that the RGB colors range from 0 to 255 and depth measurements are in meters. The parameters for the SDP algorithm and the Yang algorithm in this experiment are set as follows: 15 filtering iterations, 20 depth discretization levels (*Z*), $\lambda_s = 11$, $\lambda_c = 1$, and $\lambda_e = 1$. A filter window size of 7x7 is used for the outdoor scenes and a 13x13 window is used for the indoor scene. The filter sizes were chosen experimentally. Note that the window size and λ_s are linearly dependent on the size of the upsampling image, which in this case is 800x600.



a)

b)

c)

d)

Figure 4-22: Laser Scanned Scenes

Camera images of the four scenes: a) Office: complex indoor scene with a lot of high frequency details. b) Mine: A haul-truck being loaded by a bucket; the truck image is dark due to sun light exposure. c) Building: an extremely dark image heavily distorted with sunray artefacts and saturated by a bright sky. d) Library: a multi-surface building structure; this scene has up to 55m of depth range.

4.3.2.2 Results

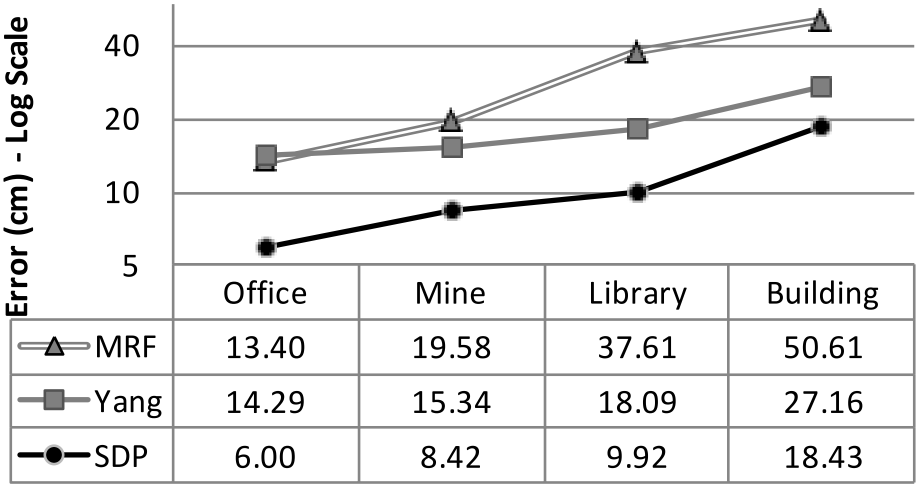
Figure 4-23 shows the average error in depth images produced by each algorithm in the scenes shown in Figure 4-22. The reported values are for a 50x upsampling factor: 7,000 laser measurements are turned into an image with more than 350k pixels.

Improved Accuracy: As shown, SDP's performance improves upon both the Yang and MRF method. On average, the mean error obtained by the Yang and MRF methods is reduced by a factor of 1.9 and 2.8 respectively. We should note that both the Yang and MRF methods were tuned to work as well as possible in the given scenes.

The laser scanner's 5.0cm mean error contributes to the reported upsampling errors. In the above four scenes, on average SDP reports a mean error of 10.7cm. The mean error is lowest in the short-range indoor Office scene (6.0cm) and highest in the Building scene with a highly distorted and dark camera image (18.4cm).

In MRF, information from a laser pixel propagates to its neighbouring pixels by minimizing depth discontinuities (i.e., smoothing the depth image). This process inadvertently smoothens the actual depth edges in the scene. An evidence of this issue can be observed in MRF's loss of accuracy in high-range images such as the Library scene, as shown in Figure 4-23. The two filtering-based methods (Yang and SDP) perform better in long-range scenes because they apply their smoothing filters to a cost-volume that represents probabilities, rather than applying it to the actual depth image itself.

Overall, we believe SDP presents a practical super-resolution approach for large upsampling factors (e.g., 50x), without a considerable loss of accuracy.



	Office	Mine	Library	Building
△ MRF	13.40	19.58	37.61	50.61
■ Yang	14.29	15.34	18.09	27.16
● SDP	6.00	8.42	9.92	18.43

Figure 4-23: Mean Absolute Error Obtained by SDP, Yang and MRF in Laser Scanned Scenes

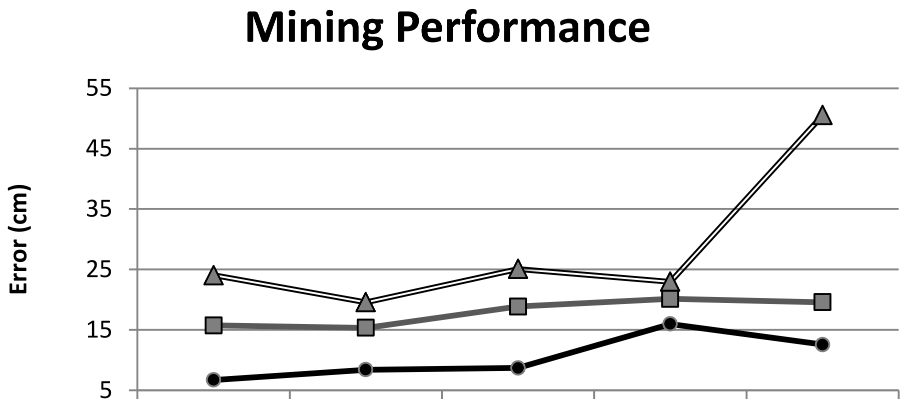
Comparison of mean of absolute errors in depth image results obtained by SDP, Yang, and MRF. The results are for an upsampling factor of 50x: 7,000 laser measurements upsampled to generate an image with 350k pixels. Note that the graph has a logarithmic scale on y-axis.

Better Edge Definition: Edge data have been removed from the ground-truth images constructed here because they are unreliable due to calibration inaccuracy (Section 4.3.2.1). Thus, the results shown in Figure 4-23 do not properly evaluate the performance of the algorithms in the difficult regions such as the object contours. It is possible, however, to visually evaluate and compare the algorithms' results.

As shown in Figure 4-25, the Yang method provides visibly better results than MRF. This can be observed on the contours of objects. However, both methods produce edges that are blurred, and object contours that are uneven. SDP, however, does not create edge-blur and shows contour definitions that are smoother and more realistic.

Better Handling of Poor Quality Camera Images: As seen in Figure 4-23, relative to other scenes the upsampling errors of all three algorithms are higher for the Building scene. This is because of the saturation and lack of contrast in the camera image caused by sun exposure. This makes image details such as edge features difficult to distinguish. Considering the poor quality of the input image, the SDP's error is within a reasonable range to make a practical case for its use in real world outdoor applications.

Mining Scenes: In order to evaluate the accuracy of the 3D images generated at a mine site, five static mining scenes were captured using the laser scanner and the camera mounted on a rope shovel (as discussed in Section 4.3.2.1). The scenes were captured during the shovel’s operation. Figure 4-24 shows the results of all five scenes. Similar to previous results, the SDP algorithm shows considerably better accuracy than the other two methods. The mean error of all algorithms increases, as the average depth of the scene increases from Scene #1 to Scene #5. On average, SDP obtains a mean depth error of 10.5cm for its mine site point-clouds. Note that the mine site images have a range of up to 40m.

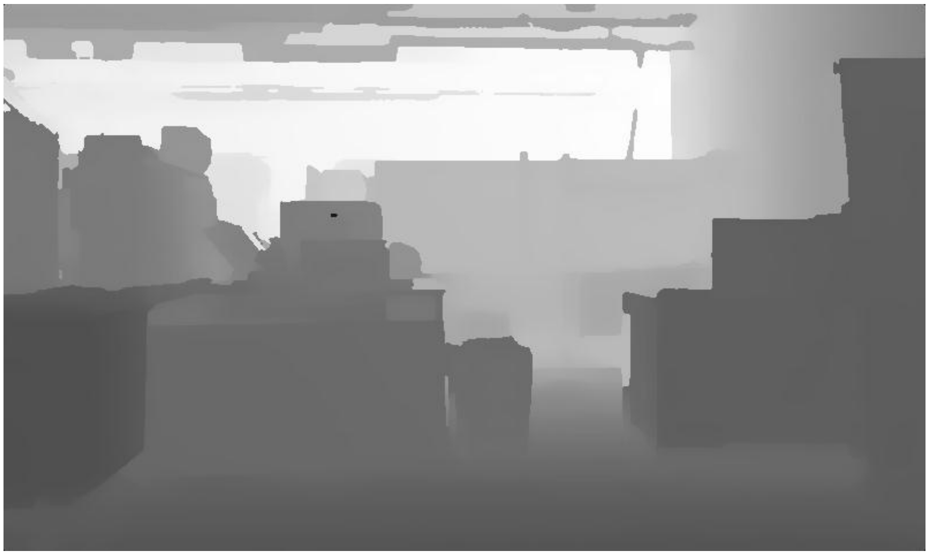


	Mine Scene #1	Mine Scene #2	Mine Scene #3	Mine Scene #4	Mine Scene #5
▲ MRF	24.07	19.58	25.08	22.98	50.54
■ Yang	15.73	15.34	18.88	20.15	19.58
● SDP	6.72	8.42	8.71	15.98	12.56

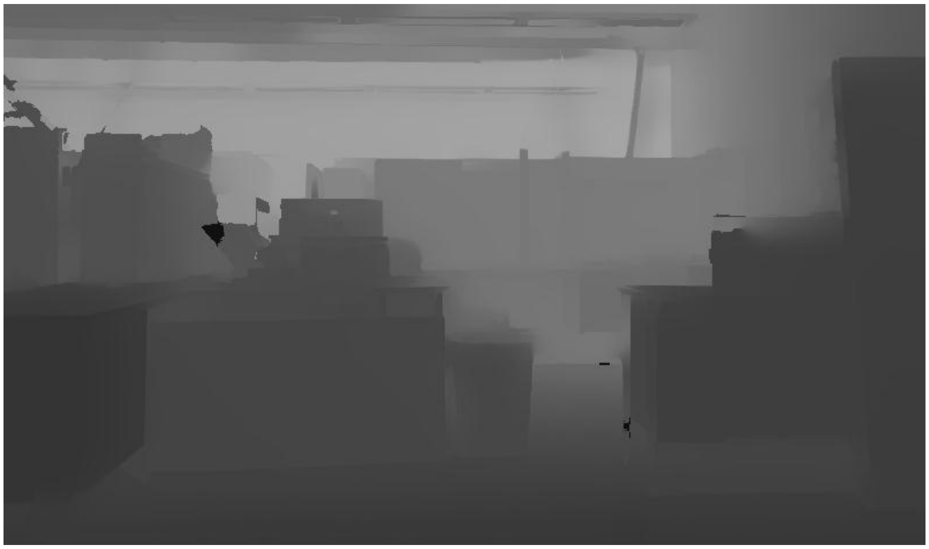
Figure 4-24: Mean Error Obtained by SDP, Yang and MRF in Laser Scanned Scenes of a Mine Site

Comparison of mean errors in depth image results obtained by SDP, Yang, and MRF in the mine site images. The results are for an upsampling factor of 50x: 7,000 laser measurements upsampled to generate an image with 350k pixels.

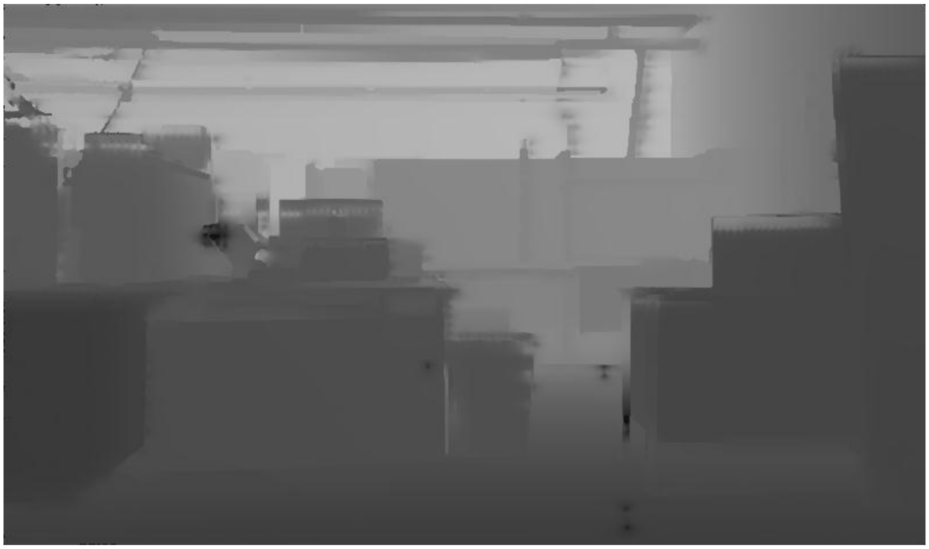
a) SDP – Office:



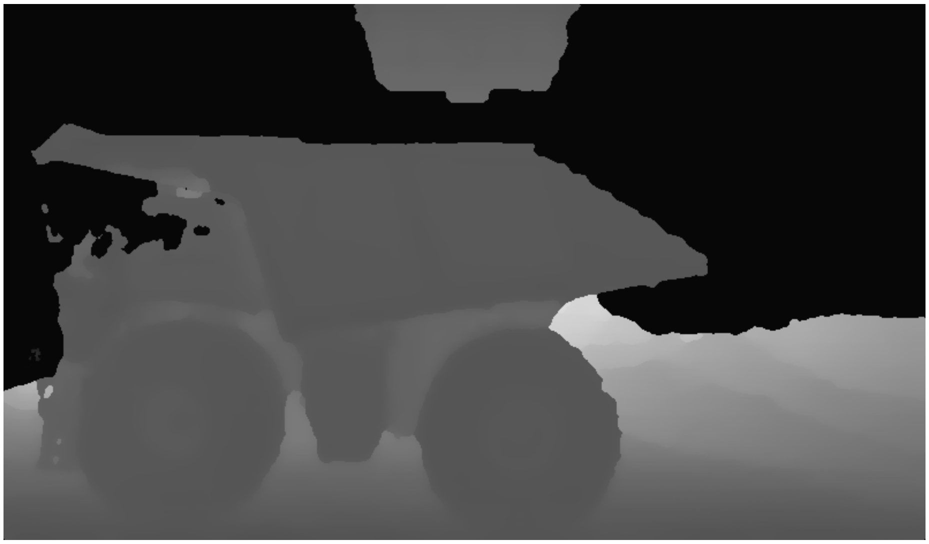
b) Yang – Office:



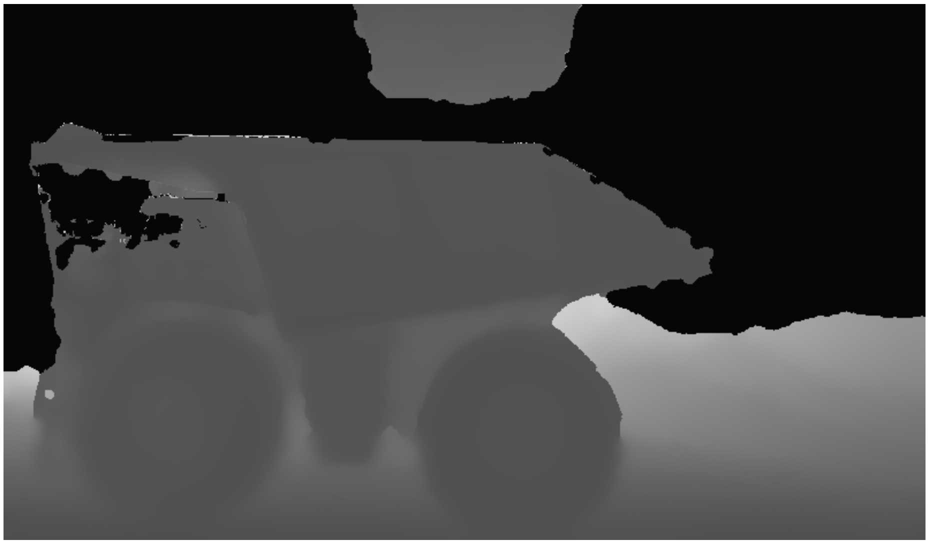
c) MRF – Office:



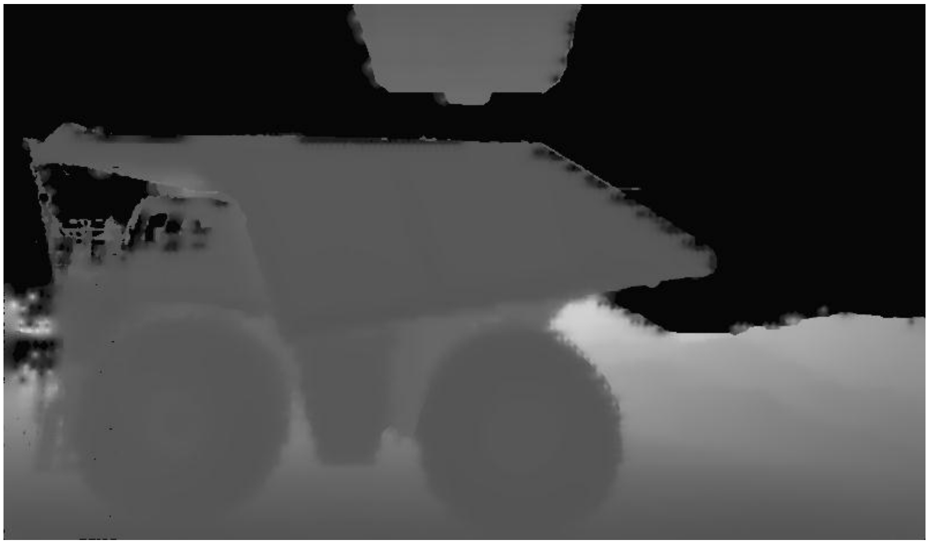
d) SDP – Mine:



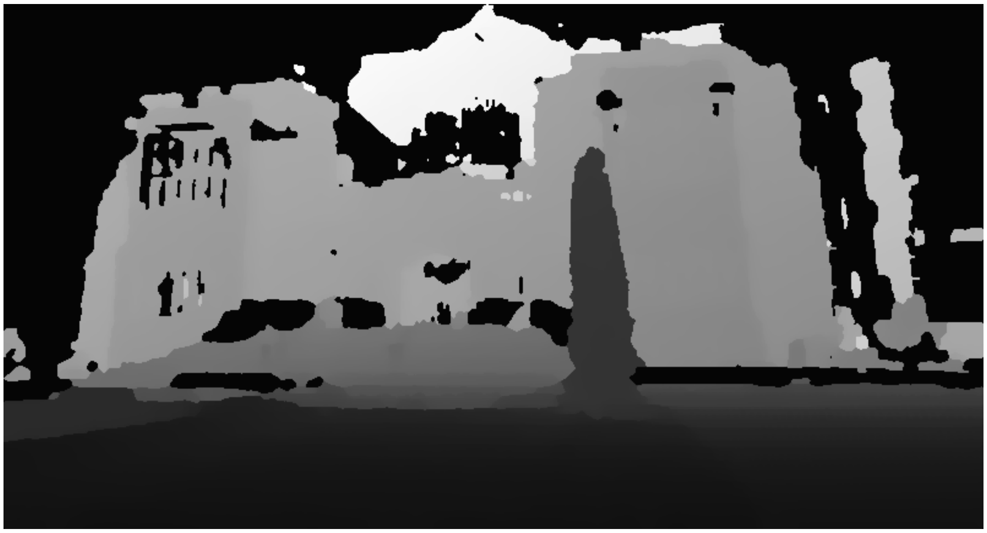
e) Yang – Mine:



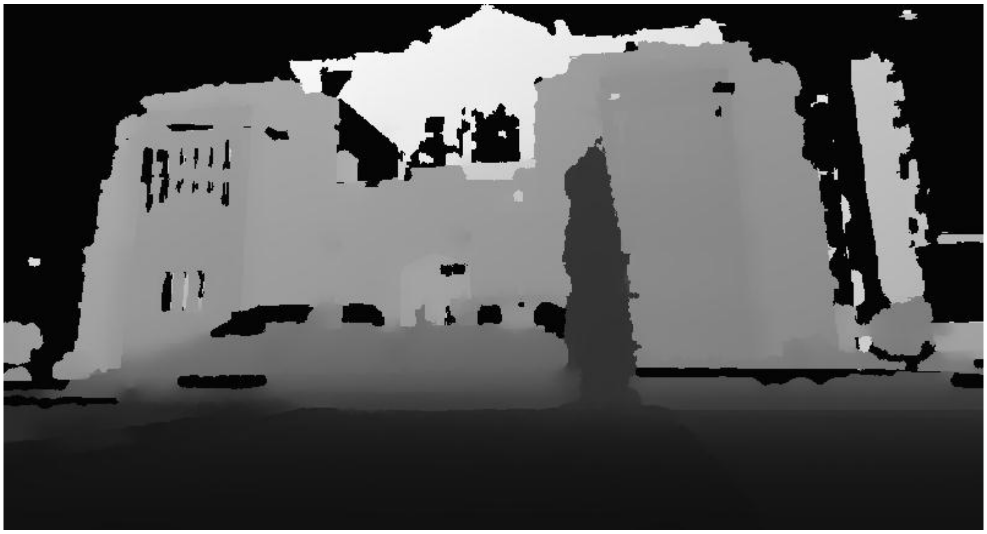
f) MRF – Mine:



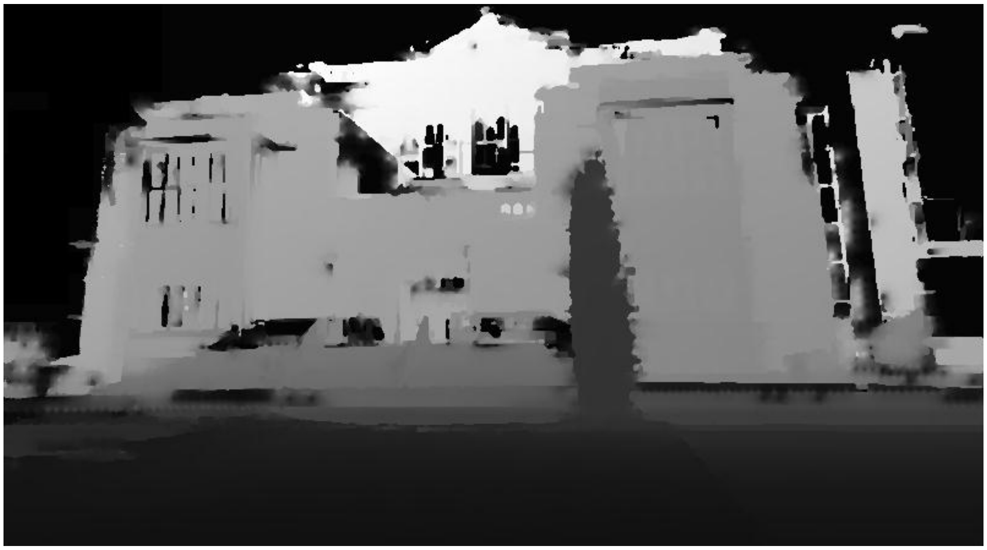
g) SDP – Library:

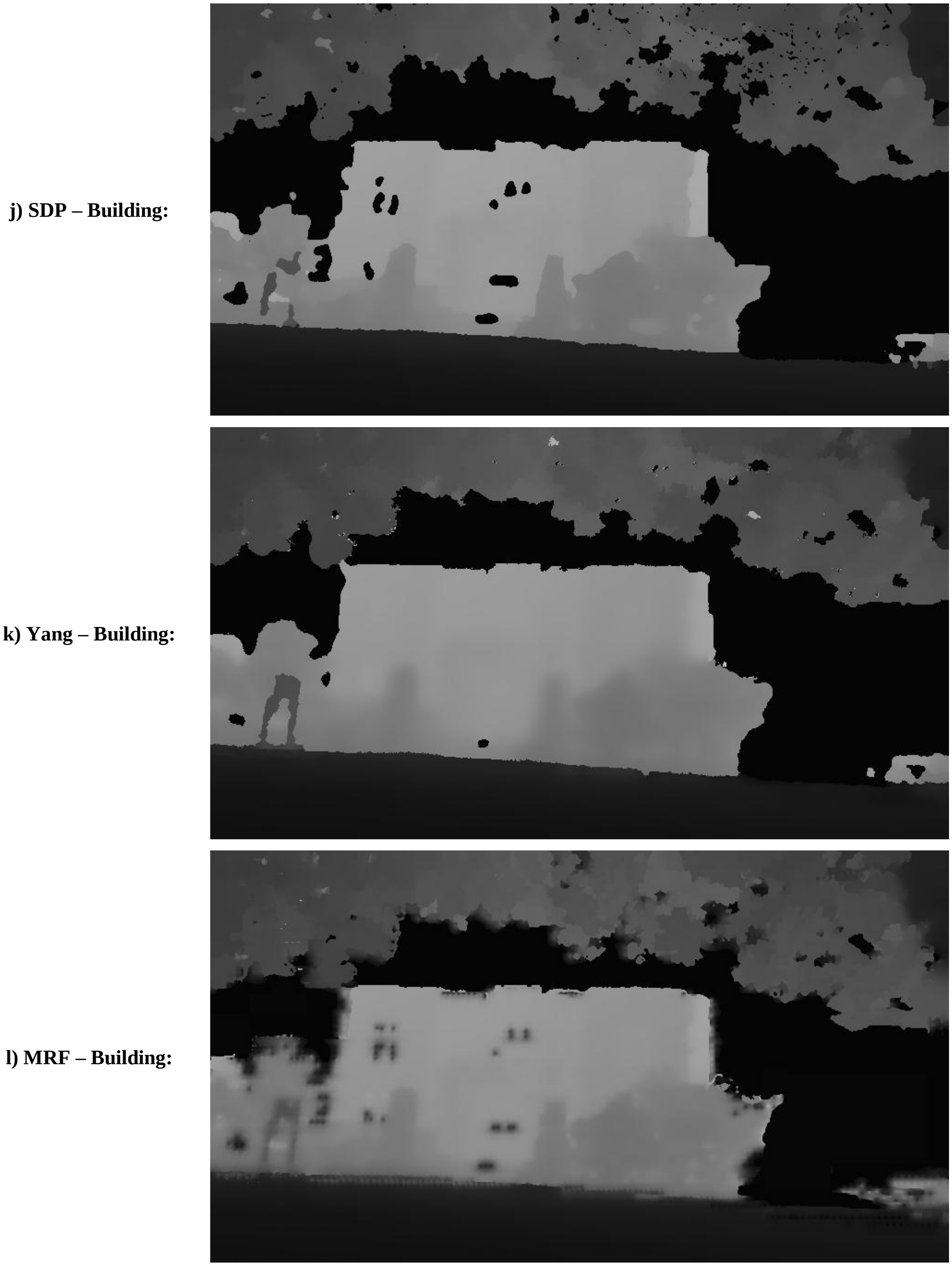


h) Yang – Library:



i) MRF – Library:





j) SDP – Building:

k) Yang – Building:

l) MRF – Building:

Figure 4-25: Results of the Super-Resolution Algorithms on Laser Scanned Scenes

Depth image results from SDP (top), the Yang method (centre), and MRF (bottom) in the Office scene (a, b, c), Mine scene #2 (d, e, f), the Library scene (g, h, i), and the Building scene (j, k, l). The Yang method shows better edge definition than MRF, and SDP shows better edge definition than the Yang method. This can be observed by sharpness of SDP edges as well as the smoothness of object contours.

4.4 Discussion

4.4.1 Improvement over the Yang Method

The calculation of cost-volume in SDP (Section 4.2.3) differs from the Yang method. As the Yang method iteratively refines its upsampled depth image by filtering its cost-volume, it reconstructs the cost-volume using the latest upsampled depth image at each iteration. Since the latest depth image contains error, this inevitably feeds that error into the system causing loss of accuracy. Meanwhile, SDP incrementally updates its probability volume at every iteration rather than constructing a new one as in the Yang method. In SDP, the depth probability distribution for a given pixel often contains multiple peaks representing more than one hypothesis. These peaks evolve as more iterations are applied and new information is introduced through neighbouring pixels. At the end of the process, the highest peak in the depth probability distribution determines the depth at that pixel. In the Yang method however, multiple hypotheses are not maintained since the cost-volume is reconstructed at every iteration based on information only from the best peak at that particular iteration (see Section 4.1.2). We believe this is a major factor that differentiates the performance of SDP and Yang.

When projected onto the camera image, laser pixels are sparse and unevenly distributed. In the Yang method, constructing a cost-volume requires an upsampled depth image, and thus in order to initialize the algorithm, an interpolation of the low-resolution laser depth image is required to match the size of the high-resolution camera image. Given the uneven distribution of the pixels, the interpolation step is not trivial. The choice of the interpolation algorithm influences the accuracy of the Yang method.

The proposed SDP algorithm, on the other hand, requires no interpolation. It constructs its cost-volume with a sparse laser depth image, and it is not sensitive to the spatial distribution of the sparse pixels. Thus, it is better suited for active depth sensors with low spatial resolution, such as lasers and TOF cameras.

4.4.2 Algorithm Performance in Edge Definition

The ground-truth images created in Section 4.3.2.1 do not evaluate the performance of the upsampling algorithms in defining object edges. That is because laser measurements of edges are unreliable due to calibration errors, and thus they are removed from the ground-truth image.

To evaluate the performance of the upsampling algorithms in edge definition, a visual inspection of the upsampled depth images can be done. In Figure 4-25, it is possible to observe blurring of depth edges in the results of the Yang and MRF methods. Figure 4-26 shows one example of a blurred edge on the Mine image: Notice the bottom half of the right tire of the truck.

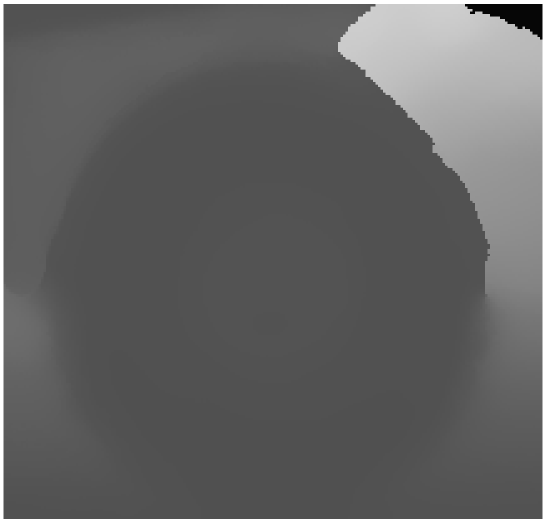


Figure 4-26: An Example of the Edge-Blur in a Depth Image Obtained by the Yang Method

The depth image of the right hand truck-tire obtained by the Yang method. The bottom half of the tire has blurry edges.

To better appreciate the problem with edge blur, Figure 4-27 shows the resulting 3D point-clouds by both SDP and the Yang method. In the result of the Yang method, the blur causes the tire to merge with the ground in one continuous surface. This is not the case for SDP. Blurring edges and thus merging objects have a negative impact on applications that require accurate object segmentation/classification.

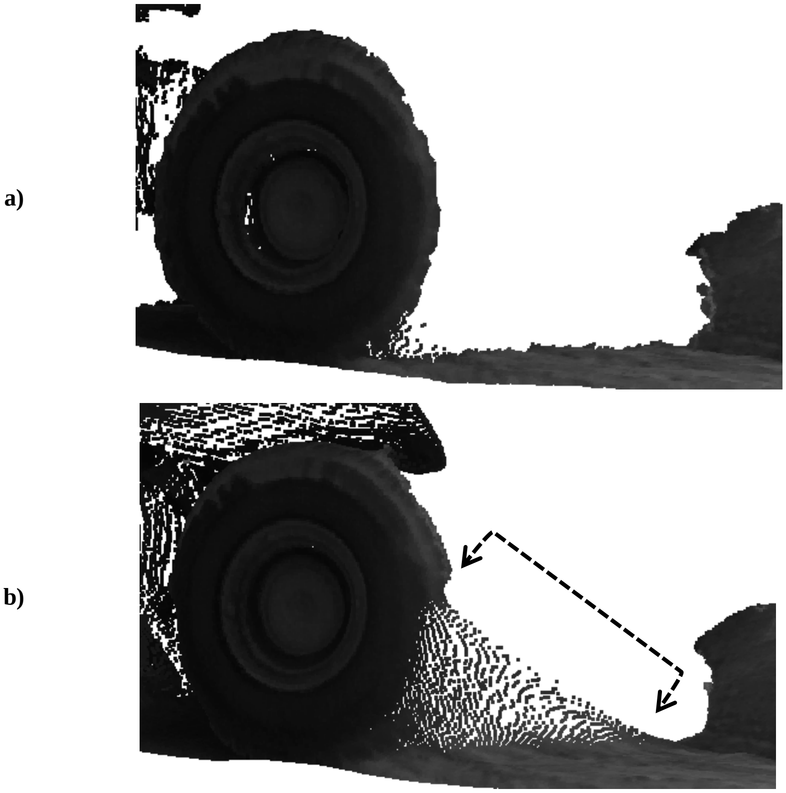
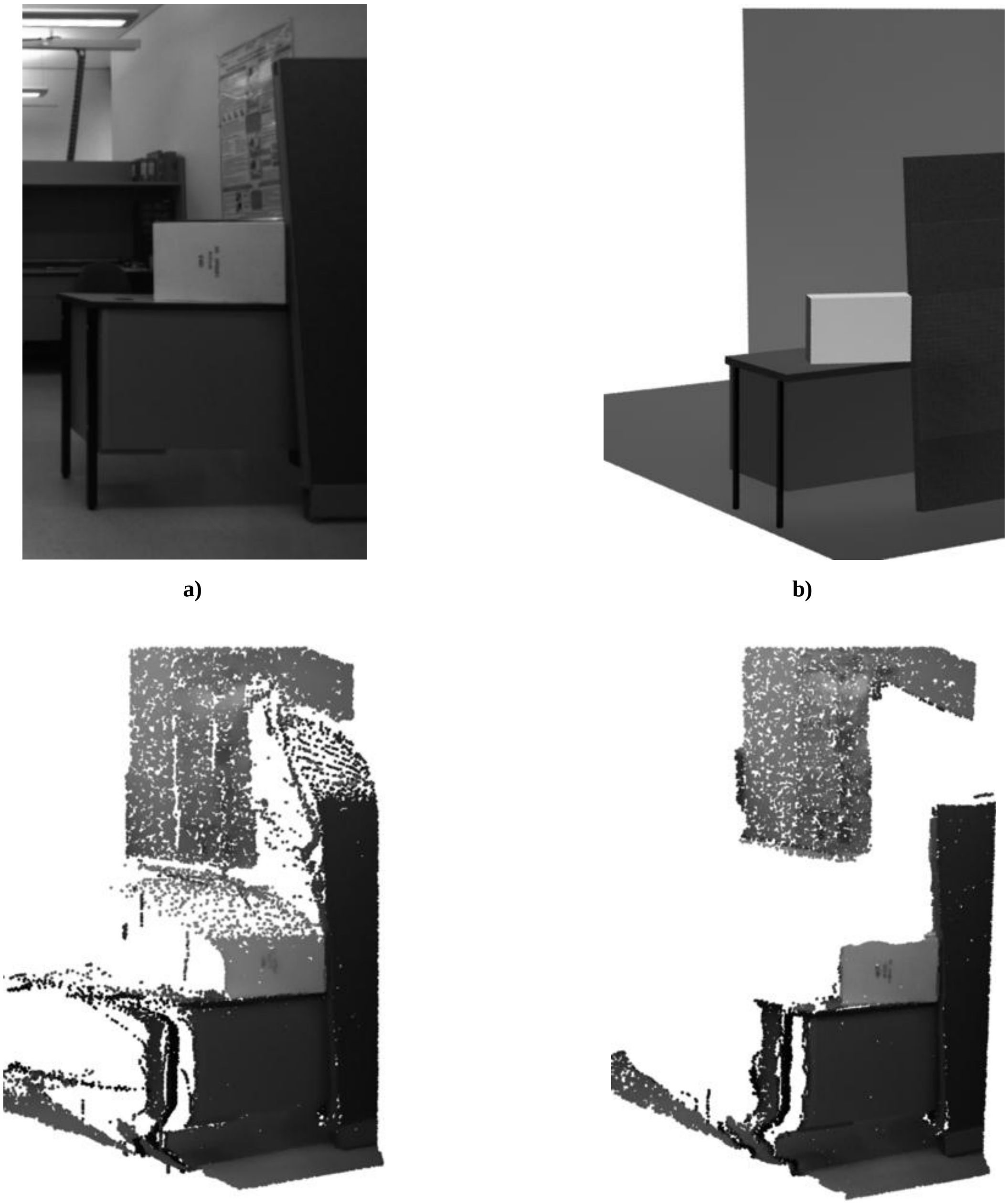


Figure 4-27: Comparison of the Edge Definition in 3D Point-Clouds of the Mine by SDP and Yang
a) The point-cloud of the truck tire, obtained by SDP, is viewed from the right-side of the scene. Note how well the tire is separated from the ground. b) The same angled-view of the point-cloud of the truck tire is shown, this time obtained by the Yang method. Due to the blur in edge definition of its upsampled depth image, the tire and the ground merge in one continuous surface (see the section highlighted by arrows).

The white box seen on the right hand table in the Office image (Figure 4-22.a) can be seen in the point-clouds obtained by the MRF and the SDP methods shown in Figure 4-28. In the MRF result, the box merges with the background wall, and the MRF algorithm fails to separate the two objects. In the SDP result, on the other hand, the objects are separated properly. Thus, the visual inspection of the 3D results suggests that SDP performs better in edge definition.



a)

b)

c)

d)

Figure 4-28: Comparison of the Edge Definition in Point-Clouds of the Office by SDP and MRF

a) The white box on the right hand table, in the Office scene. b) A synthetic geometric model of that segment of the scene is shown; the scene is viewed from an upper-left angle. c) The same angled view of the point-cloud obtained by the MRF algorithm, d) as well as the point-cloud obtained by the SDP algorithm. A separation is expected between the wall in the background, and the foreground objects: the box and the partition panel on the right. The MRF result merges the foreground objects with the wall, while the SDP result offers the expected separation. Please note that the white areas represent regions that are occluded in the laser view, and hence no laser measurements are available.

It must be noted that the ground-truth images in the Middlebury datasets quantitatively evaluate the upsampling performance on both the continuous surfaces as well as the object edges. Therefore, the upsampling results presented in Section 4.3.1 show that SDP is more accurate overall, when edge regions are included as well.

4.4.3 Importance of the Rejection Step

The numerical results in Figure 4-23 mainly evaluate the performance of the upsampling formulation in the propagation step (Section 4.2.3). It does not, however, evaluate the benefits of our rejection step (Section 4.2.2). This is because laser measurements of edge regions were removed from our ground-truth images (see Section 4.3.2.1) using the same edge-detection method proposed in SDP rejection step, since they were found to be unreliable due to calibration errors (Section 4.2.2). On the other hand, the edge regions in the upsampled image contain depth values that are generated by the data propagation step. The rejection step only improves these edge regions, and thus, its benefits are not evaluated using our ground-truth images (that do not contain edge regions). This also means that the numerical results presented in Figure 4-23 do not depend on whether the rejection step is performed.

Figure 4-29 shows the result of the SDP algorithm when the rejection step is not included. A visual comparison with Figure 4-25.d reveals the improvements introduced by the rejection step: in Figure 4-29, the edges are poorly defined and the object contours are not even and realistic. Particularly, observe the right tire, the left edge of the bucket (shown at the top), and the right edge of the canopy.

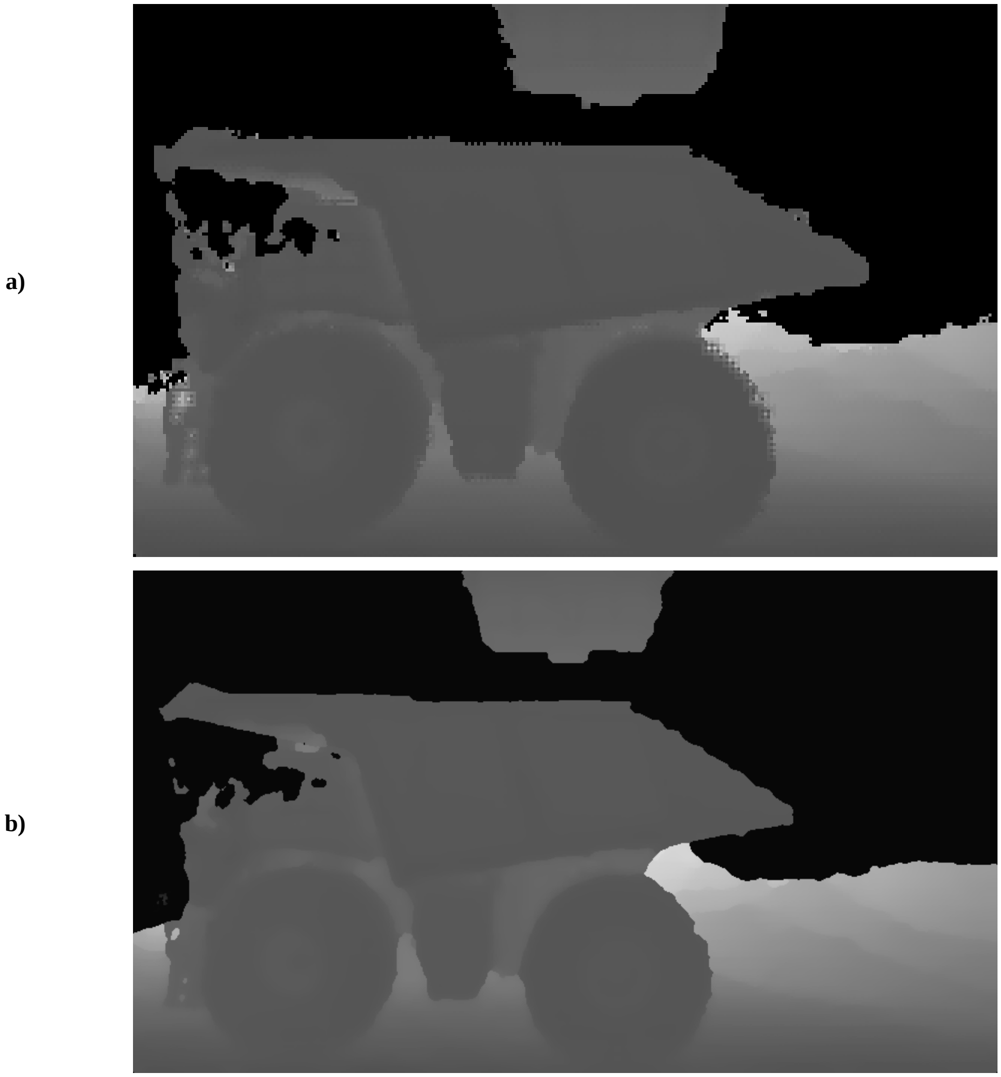


Figure 4-29: The Result of SDP on the Mine Scene, With and Without the Rejection Step

a) The result of the SDP algorithm without the rejection step. b) The result of the SDP algorithm with the rejection step. The edges on the right tire, the truck box and the bucket are not as well-defined in the top image.

Another way to demonstrate the importance of our rejection algorithm is by including it in another super-resolution method. Figure 4-30 shows the depth image result obtained by the Yang method when the rejection step is introduced. The edge definition improves considerably, compared to the same algorithm without the rejection step (see Figure 4-25.e). For instance, notice the improvement in the contour of the bucket shown at the top. As described above, the numerical results of the Yang method reported in Figure 4-23 would not improve if the rejection step is applied, even though the result is more visually similar to the SDP result in Figure 4-25.d.

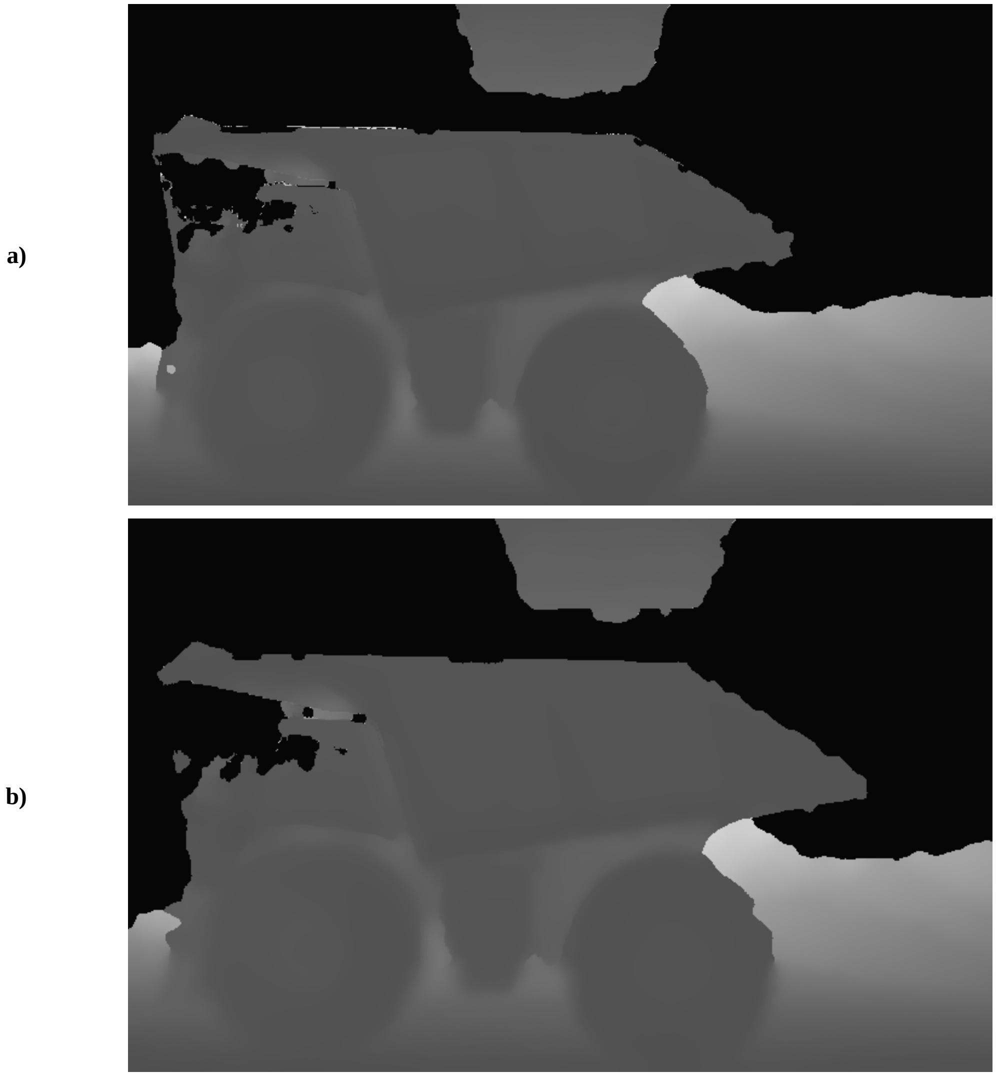


Figure 4-30: The Result of the Yang Method, With and Without the Rejection Step

a) The result of the Yang algorithm without using the rejection step. b) The result of the Yang algorithm when used in conjunction with the rejection step. The quality of the edges improves. For instance, notice the improvement in the contour of the bucket on top of the image.

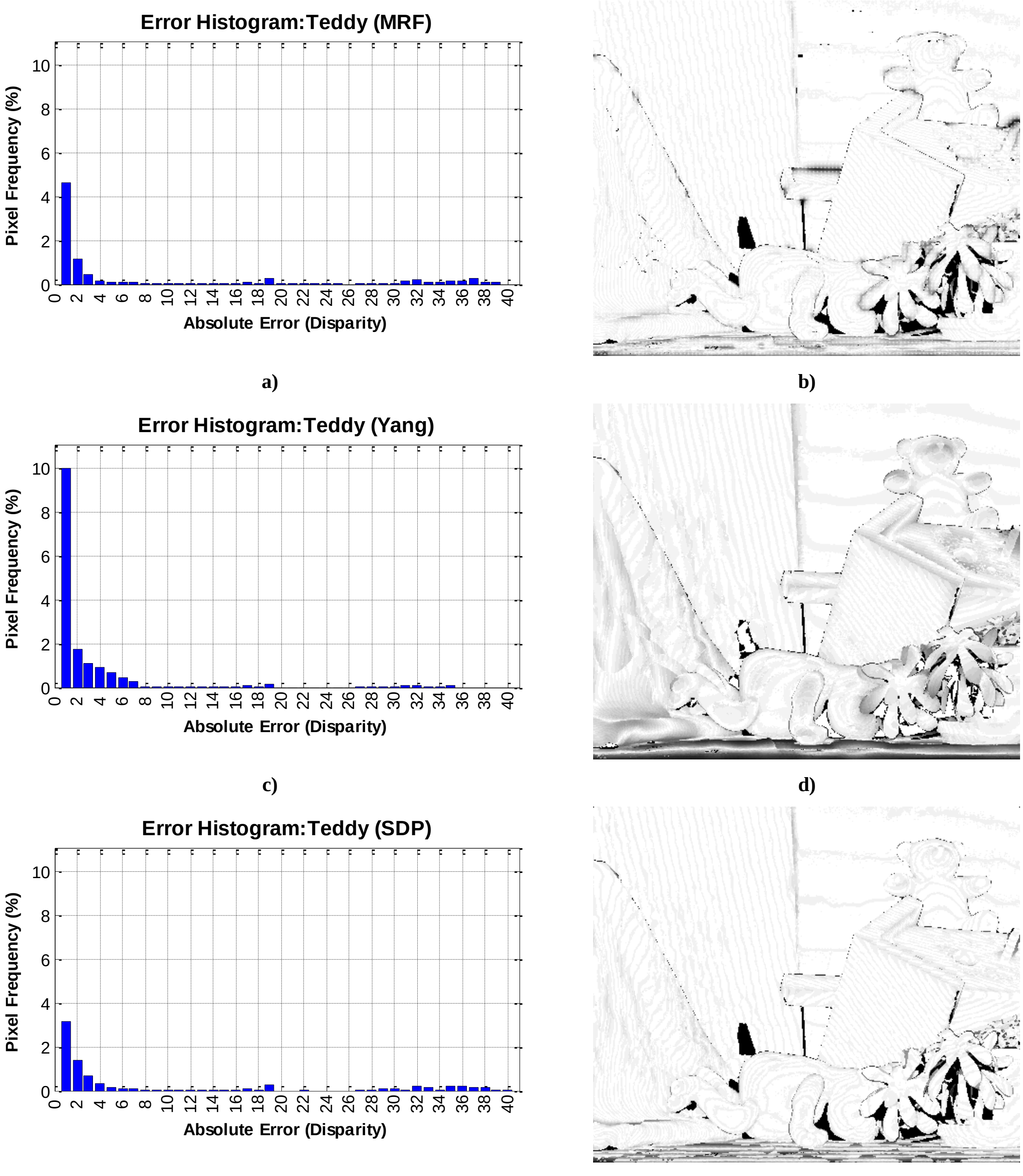
4.4.4 Error Distribution

To better evaluate the performance of each super-resolution algorithm discussed in this work, the histograms representing the error distribution for each algorithm are presented for a number of experiments. Figure 4-31 and Figure 4-32 show the histograms for Teddy and Cones images in Middlebury. The histograms plot percentage of depth pixels (*y* axis) having a given depth error (*x* axis).

Note that in Middlebury datasets, depth images represent disparity value rather than distance value, with disparity being inversely proportional with distance.

Along with the histograms, gray-scale error images are presented that show the amount of absolute depth error at each pixel. These images help show where the errors occur, spatially. Darker pixels have a higher error value. To make the spatial distribution of the errors in the images visually presentable, color adjustments have been made to each image separately. That means error images that on average contain less error had to be amplified (i.e., all values multiplied by a constant factor) to make the spatial distribution of their depth errors visible. Therefore, the overall intensity in each image is not representative of the overall algorithm performance, and must not be compared against error images from other algorithms. It can be seen from the images that, as expected, depth errors mostly occur near object edges for all algorithms.

Figure 4-33 and Figure 4-34 show histograms and error images for SDP, Yang and MRF in the laser scanned scenes. It can be observed from the histograms that MRF tends to have more pixels with larger depth errors, which is why its mean error presented in Figure 4-17 and Figure 4-24 is higher than Yang. Yang, on the other hand, has a higher frequency of pixels with lower error value, compared to MRF. Finally, SDP’s error distribution shows an overall improvement over both Yang and MRF.



a) b)

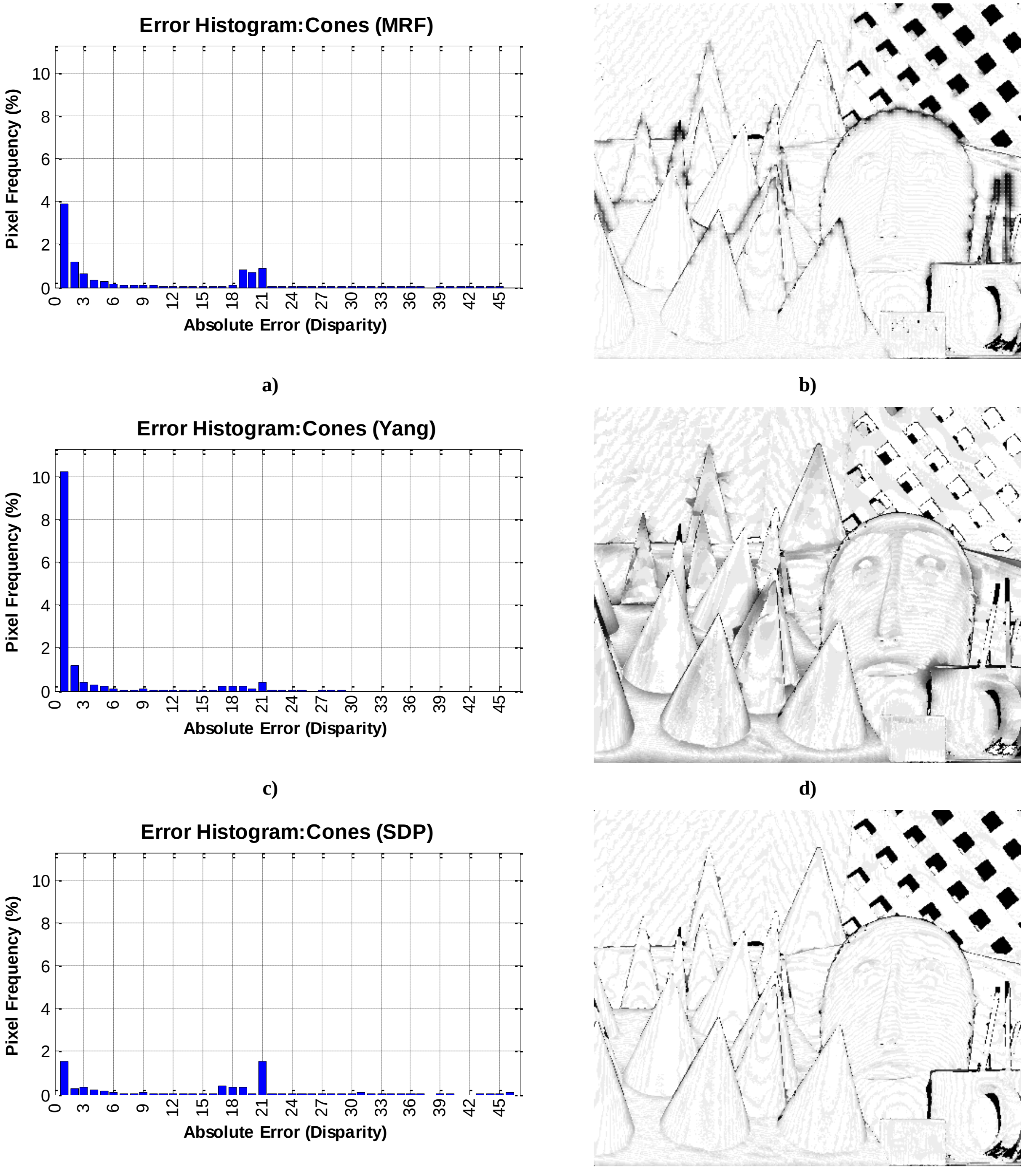
c) d)

e) f)

Figure 4-31: Error Images and Histograms in Middlebury's Teddy Image

a,c,e) Error histograms for MRF, Yang and SDP respectively. The histograms plot percentage of depth pixels (*y* axis) having a given depth error (*x* axis).

b,d,f) "Error images" for MRF, Yang and SDP respectively. Error images show the absolute depth error at each pixel – darker pixels have higher errors. Images have been adjusted make the spatial distribution of depth errors better visible in print. Therefore, the images must not be compared with each other for evaluating error performance of the algorithms.

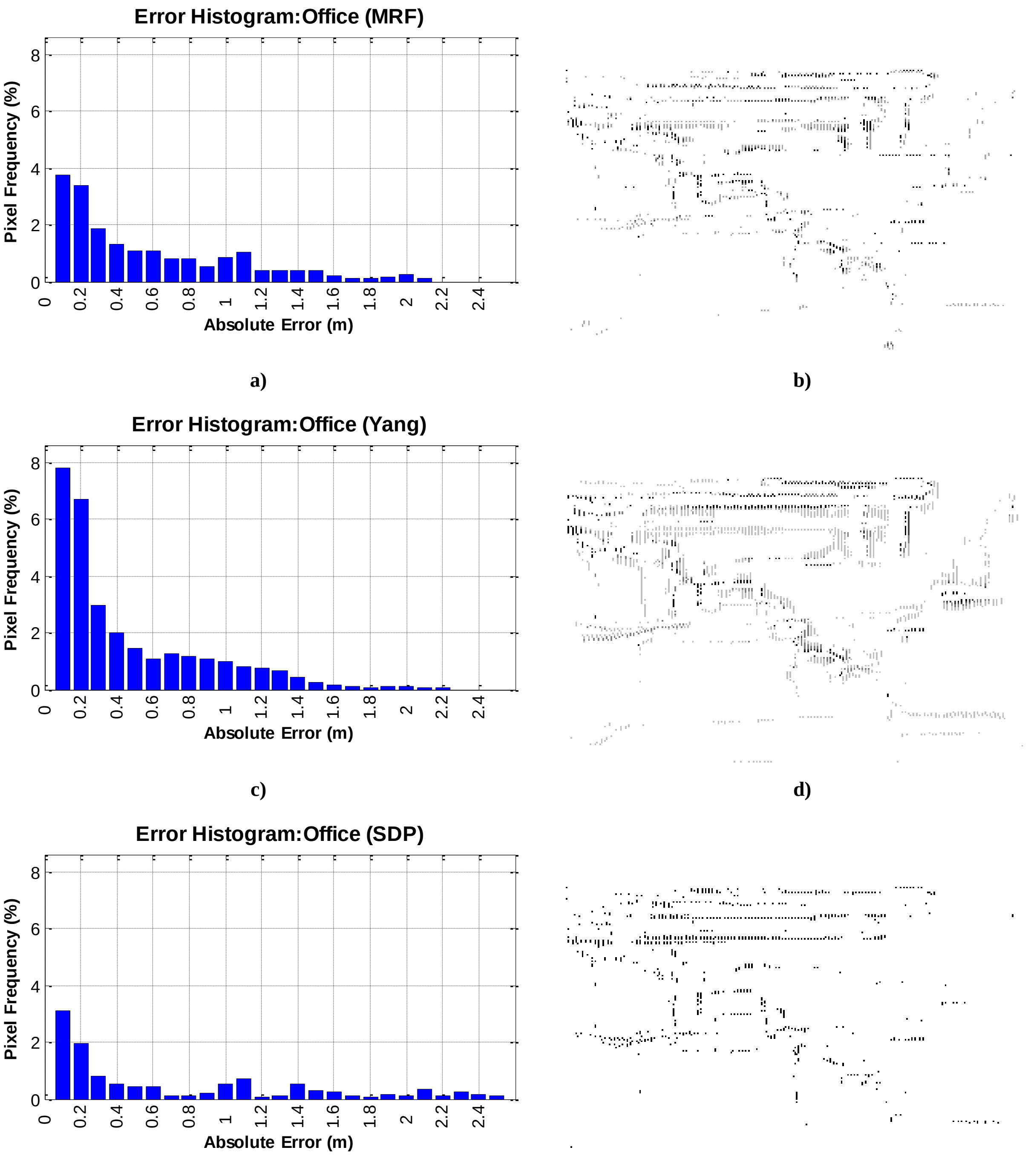


a) b) c) d) e) f)

Figure 4-32: Error Images and Histograms in Middlebury's Cones Image

a,c,e) Error histograms for MRF, Yang and SDP respectively. The histograms plot percentage of depth pixels (*y* axis) having a given depth error (*x* axis).

b,d,f) "Error images" for MRF, Yang and SDP respectively. Error images show the absolute depth error at each pixel – darker pixels have higher errors. Images have been adjusted make the spatial distribution of depth errors better visible in print. Therefore, the images must not be compared with each other for evaluating error performance of the algorithms.



a) b)

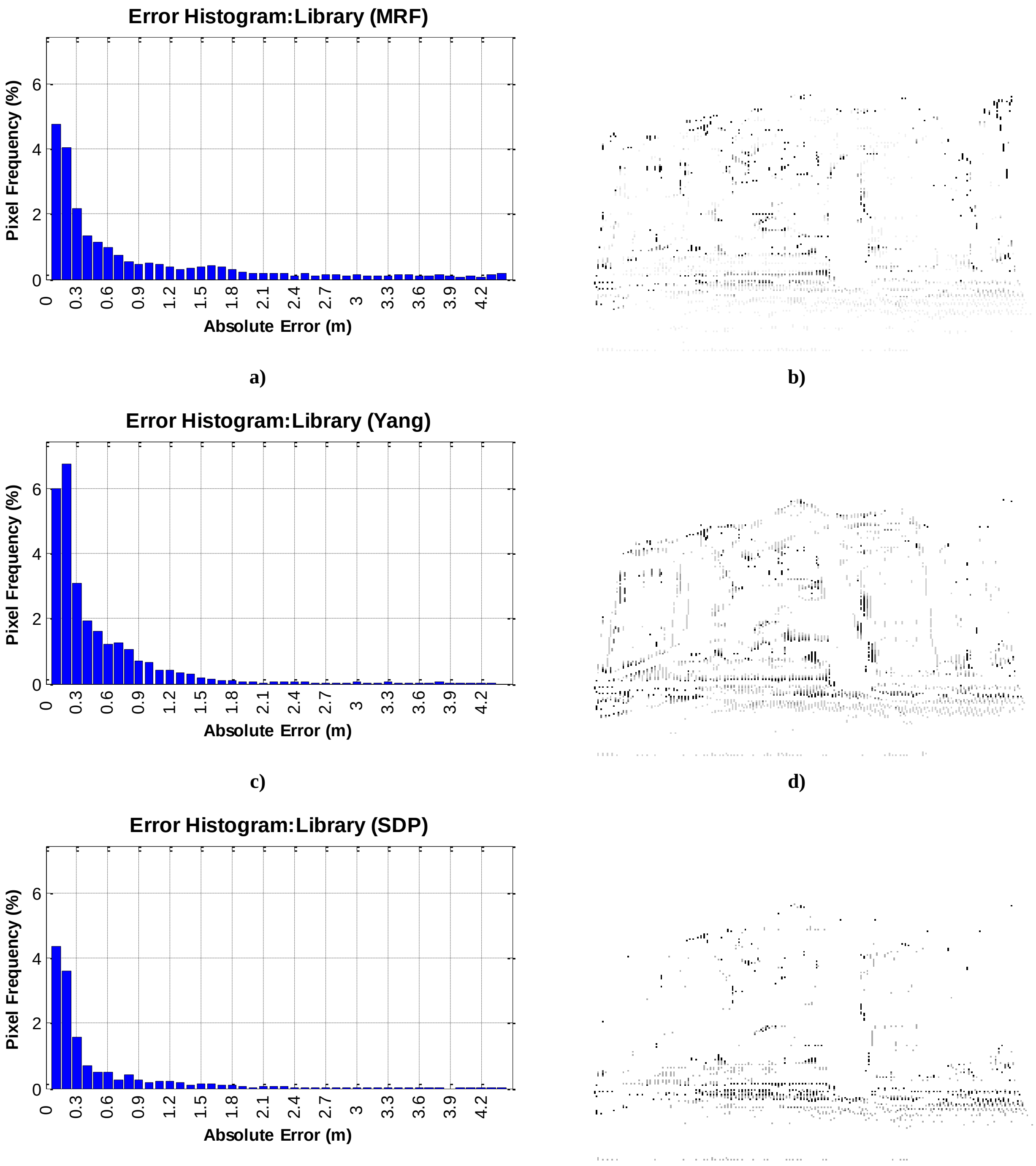
c) d)

e) f)

Figure 4-33: Error Images and Histograms for the Office Scene

a,c,e) Error histograms for MRF, Yang and SDP respectively. The histograms plot percentage of depth pixels (*y* axis) having a given depth error (*x* axis).

b,d,f) "Error images" for MRF, Yang and SDP respectively. Error images show the absolute depth error at each pixel – darker pixels have higher errors. Images have been adjusted make the spatial distribution of depth errors better visible in print. Therefore, the images must not be compared with each other for evaluating error performance of the algorithms.



Error Histogram:Library (MRF)

Error Histogram:Library (Yang)

Error Histogram:Library (SDP)

a)

b)

c)

d)

e)

f)

Figure 4-34: Error Images and Histograms for the Library Scene

a,c,e) Error histograms for MRF, Yang and SDP respectively. The histograms plot percentage of depth pixels (*y* axis) having a given depth error (*x* axis).

b,d,f) “Error images” for MRF, Yang and SDP respectively. Error images show the absolute depth error at each pixel – darker pixels have higher errors. Images have been adjusted make the spatial distribution of depth errors better visible in print. Therefore, the images must not be compared with each other for evaluating error performance of the algorithms.

4.4.5 Upscaling Ratio

In Figure 4-17, we showed using the Middlebury datasets how SDP is capable of maintaining its accuracy when the upsampling ratio increases. Figure 4-35 shows the increase in error as the upsampling ratio changes from 5x to 50x. It can be observed once again that the increase in SDP error is small, even though the upsampling ratio increases significantly.

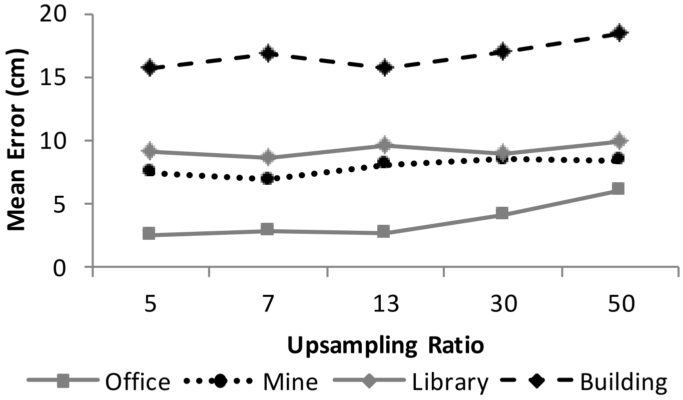


Figure 4-35: SDP's Mean Error vs. Upsampling Ratio

The SDP's upsampling error for five different upsampling ratios varying from 5x and 50x. The increase in error is reasonably small given the increase in upsampling ratio.

4.4.6 Computational Requirement

The number of iterations required for SDP to converge depends on the application at hand. The overall accuracy is obtained in 10 to 15 iterations, while fine-tuning the edge definition in complex scenes may require up to 50. Figure 4-36 shows the error in the Building scene decrease as more iterations are applied. The error does not change significantly after 10 iterations.

Other than the number of iterations, another parameter that affects the computational requirement of SDP is the number of depth discretization levels. Fewer depth discretization levels means fewer filtering operations. It was found through experimentation that 20 levels of discretization are sufficient for obtaining best accuracy (see Figure 4-37).

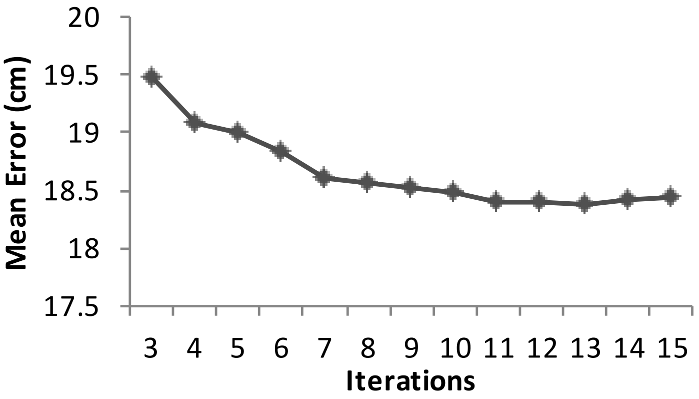


Figure 4-36: SDP's Mean Error vs. the Number of Iterations

Decrease in error for the Building scene is shown as more iterations of SDP are performed. After 10 iterations, the change in error is minimal.

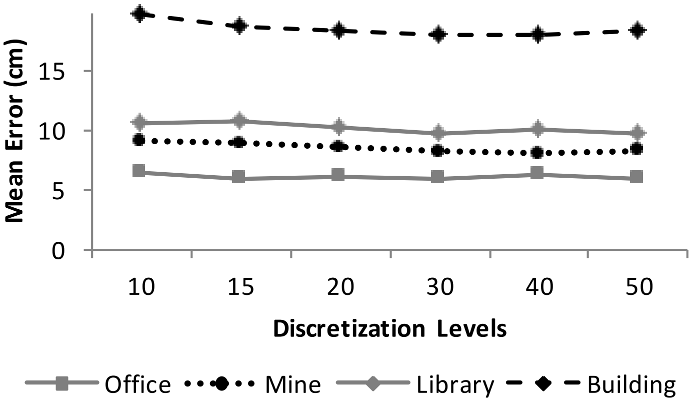


Figure 4-37: SDP's Mean Error vs. Number of Depth Discretization

Mean error vs. the number of depth discretizations levels, for all four scenes. It can be seen that the change in mean error is marginal above 20 discretization levels; so no more than 20 levels is required.

The processing cost of SDP increases linearly with respect to the number of iterations, the number of discretization levels, and the upscaling ratio defined as the number of upsampled image pixels over the number of laser depth pixels. This is because the computational cost for calculating the SDP filter for each pixel stays constant with respect to the mentioned three parameters.

Figure 4-38 shows the average processing time for a MATLAB implementation of SDP along with some C++ (MEX) optimization. The processing times are plotted against the upsampling ratio, to show the increase in time as the size of the output depth image increases. The processing times along with the size of the upsampled depth image are listed in Table 4-2 for a number of the experiments.

Due to the small size of the input depth image from the laser scanner, the processing cost of the SDP's rejection step is marginal compared to its propagation step. The computational cost of SDP's propagation step is similar to the Yang method given the same number of iterations, kernel size, and number of discretization levels in this work. This is because both algorithms run a bilateral filter iteratively, even though each algorithm uses a different set of equations to initialize and update its cost-volume.

A number of super-resolution methods are currently capable of performing in real-time [28], [32], [35], but in terms of accuracy, it is shown in this work that SDP improves upon the current state-of-the-art algorithms. Optimization methods such as parallelization have been suggested to improve efficiency of bilateral filtering [28], [108]. While we believe it is possible to obtain real-time performance for smaller upsampling ratios, further work needs to be done to find ways for real-time computation of higher ratio upsampling.

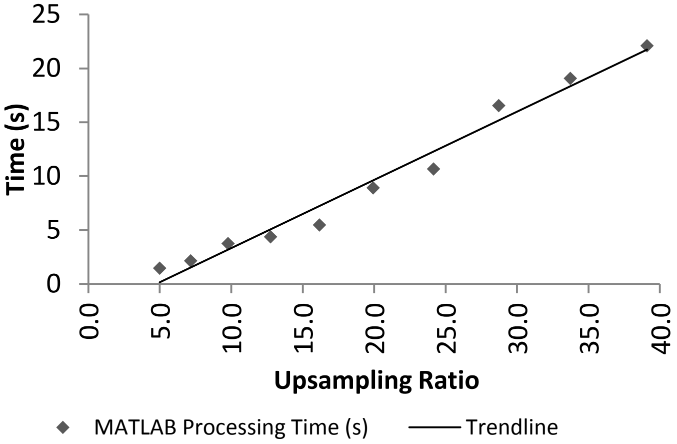


Figure 4-38: Processing Time vs. SDP Upsampling Ratio

The processing time (in seconds) required by a MATLAB implementation of SDP, plotted against the upsampling ratio.

Table 4-2: Average Processing Time of SDP With 10 Iterations and 20 Discretization Levels.

Upsampling Ratio	Image Size (pixels)	Time (seconds)
5x	250x188	1.46
9.8x	350x263	3.74
19.9x	500x375	8.90
33.7x	650x488	19.04
51.1x	800x600	38.38

4.5 Conclusions

The probability-distribution-based super-resolution algorithm proposed in this paper (SDP) offers more accurate results compared to previous works: less than ¼ of the error obtained by the MRF and Yang methods in the Middlebury datasets (Section 4.3.1), and less than ½ the error obtained by the same methods in the laser-scanned datasets gathered for this thesis and reported in Section 4.3.2. Furthermore, as shown in Figure 4-35, SDP is capable of maintaining its accuracy with larger upsampling ratios. These are achieved by the handling of sensor calibration error using the proposed rejection step (i.e., foreground edge removal and occlusion rejection), and a new formulation of the cost-volume initialization and data propagation.

The proposed SDP algorithm is a practical method for real-world applications of depth imaging since it requires minimal parameter adjustment across varying scenes (i.e., larger filter size needed in indoor environments). Moreover, compared to existing algorithms, it obtained sharper and more realistic edge definitions making it better suited for applications such as object segmentation and classification (as shown in Figure 4-25). It is also visually superior to other methods, which presents an advantage for 3D modelling and visualization applications. Additionally, it is especially suited for fusion of a camera image with an active depth sensor, because it can initialize itself using a sparse and unevenly distributed depth image.

Using the sensors mounted only at the boom of the shovel and the SDP algorithm, it is possible – although not actually demonstrated in this thesis – to obtain all necessary workspace variables outlined in Appendix A: the shovel's arm geometry, the location of the truck [112], the location of other obstacles, and the swing angle of the shovel. Moreover, depending on the position of the sensor, other information such as the truck load profile and the dig-surface profile could also be extracted from the high-resolution depth images obtained by the SDP algorithm. With an obtained average depth error of less than 10.5cm in mining scenes, and the *x*-*y* spatial resolution of 4.3cm above, the SDP generates depth images that are within required accuracies for each of the workspace parameters (e.g., 30cm for dipper tracking, 25cm for truck localization).

Further work needs to be done to obtain real-time performance for higher upsampling ratios. Moreover, the addition of video information (i.e., temporal dimension) is another way to improve this technology and compensate for the slow laser scan-rate. In this case, older depth image frames can be used in the construction of a new depth image. Thus, the system would not need a time-consuming full 3D laser scan to generate each frame of its upsampling result: it can utilize new partial laser measurements of the scene along with the past laser measurements for the rest of the scene to generate a new upsampled image. For instance, while the laser is busy scanning the truck tires, older depth information of the rest of the truck can be utilized in constructing a new depth image of the scene. Motions in the scene can be monitored using the camera images, and the previously obtained depth information can be updated accordingly to reflect the scene changes. Thus, the addition of temporal information could present a solution for capturing dynamic scenes.

Chapter 5

Conclusions

5.1 Summary of Contributions, and Conclusions

This work was motivated by the need to introduce collision detection to mining operations, in order to improve their safety and efficiency. The needs of this application led to the development of techniques to extract arm geometry variables and enhance 3D imaging for workspace monitoring.

Non-Contact Arm Geometry Extraction

Based on bucket pose measurements using a planar laser scanner and inverse arm kinematics, arm geometry variables were estimated and compared with joint sensor measurements. A preliminary bucket-tracking algorithm was first tested on a hydraulic mini-excavator. The results were further developed and made applicable to an electric rope shovel with a larger workspace area and less distinctive bucket geometry. The results showed that the laser scanner-based method produced worst case errors of 1.4˚ degrees for joint angle estimations on the excavator and 1.6˚ on the rope shovel, compared to the Motion Metrics joint angle sensors mounted on the machines. The Motion Metrics joint angle sensors themselves were specified to have a worst-case error of 1.6˚ degrees.

The developed ICP-Aided BPF algorithm had a mean error of 6.7cm on the rope shovel (i.e., less than 0.3% of the dipper's effective reach) and a maximum error of 21.5cm. It tracks the excavator end-effector and calculates its arm geometry variables with a low computational cost suitable for real-time performance. The ICP-Aided BPF offers solutions to deal with tracking problems such as occlusion and shape variations.

The three main contributions of the proposed ICP-Aided BPF include:

- The development of a retrofitable cab-mounted laser-based arm geometry extraction system.
- A novel integration of BPF and ICP algorithms that locates the shovel end-effector within 0.3% of its range. To achieve this, the proposed ICP-Aided BPF algorithm utilizes three sources of information: temporal data, the dipper's kinematic constraints, as well as the dipper's geometrical model.
- Achieving real-time performance – processing 19.5 scans per second – by utilizing Distance Transformation in the BPF algorithm.

3D Workspace Monitoring

A 2D laser scanner proved suitable for arm geometry extraction, but it is unable to provide full 3D images of the scene which is needed for workspace monitoring applications such as truck pose estimation and dig surface profiling. An accurate, slow and sparse laser scanner can be complemented by a fast high-resolution camera to create images suitable for dynamic scene monitoring in outdoor environments. This motivated the development of a 3D imaging technology that utilizes a swinging laser scanner and a camera. The objective of the proposed sensor fusion approach is to obtain 3D images with accuracy similar to the laser scanner and spatial-resolution similar to the camera.

The proposed probability distribution-based super-resolution algorithm (SDP) achieves accurate results and improves upon previous super-resolution algorithms in the literature [30], [31], producing less than ¼ of the error obtained by the MRF and Yang methods in the Middlebury benchmark datasets (Section 4.3.1), and less than ½ the error obtained by the same methods in the laser-scanned datasets collected in field tests for this thesis (reported in Section 4.3.2). Moreover, the SDP method is shown to be capable of maintaining its depth accuracy with larger upsampling ratios (e.g., up to 50x ratio as shown in Figure 4-35) while increasing the spatial x-y resolution of the laser scanner from 26.2 cm to 4.3 cm (as shown in Section 4.5). Compared to the literature, SDP's improvement in edge definition makes it visually superior and better suited for applications such as object segmentation and classification.

The three major contributions of the proposed SDP algorithm are:

- Introduction of a foreground edge removal method.
- Proposing the rejection step to deal with sensor calibration error, which was shown to improve the quality of the resulting depth image (see Figure 4-29 and Figure 4-30).
- Improved formulation of cost-volume calculation and iterative bilateral filtering, which on average produced depth images that were 4.0 times more accurate than Yang's alternative formulation when tested using the Middlebury benchmark.

Other Sensors for SDP

While a 3D laser scanner was used in the fusion approach in this work for capturing 3D depth images, any active sensing technology that produces depth imagery can be utilized instead. The choice of the depth sensor depends on the application in hand. For instance, indoor applications could utilize a TOF camera instead. Structured lighting could also be utilized for faster depth image acquisition. Independent of the choice of the depth sensor, integration with a camera is a valuable technique for obtaining high resolution 3D images, given that passive cameras produce higher resolution images than active depth sensors such as laser scanners, TOF cameras and structure lighting.

Promise of SDP for Future Mining Applications

The proposed super-resolution algorithm has been tested on images obtained on a rope shovel. The algorithm constructs high-resolution 3D images of the scene with an average depth error of less than 10.5cm. The high-resolution 3D images produced by SDP can potentially be used to extract all necessary workspace variables for a rope shovel collision avoidance system (see Appendix A):

- The shovel's arm geometry can be extracted by tracking the dipper (see Chapter 3).

- The truck can be located using the algorithm in [112]. Point-clouds from a stereo-camera are used in [112] to locate the truck. It is reasonable to assume that replacing the stereo-camera with a considerably more accurate laser-based system could improve the accuracy of the algorithm.
- The rope shovel swing angle could be measured using a point-cloud matching algorithm to track the sensor motion by registering consecutive point-clouds of scene. *Scan-matching* is an algorithm used for this purpose in [16], [113].

Furthermore, other workspace information such as obstacle locations, load profile and dig-surface profile could also be extracted from the high-resolution 3D images produced by the SDP algorithm. Using a single cab-mounted sensor package to extract all necessary workspace variables can be very beneficial to mining operations. It makes the system easily retrofittable with minimal down-time required for installation and maintenance. The proposed SDP algorithm demonstrates potential for this purpose.

5.2 Limitations and Future Work

More field experiments are required to evaluate the durability of the proposed systems on a rope shovel, and the effects of environmental factors such as fog and dust on the algorithm performance. It should be noted that currently the rope shovels are not operated when the operator has limited visibility (e.g., severe weather conditions), and therefore the proposed system is not required to operate in the most severe conditions. Moreover, this work was applied to two machines – a hydraulic mini-excavator, and a rope shovel; further work would be required to evaluate the algorithm on other excavators such as a hydraulic excavator or a dragline.

The work presented in this thesis shows promising results for proximal workspace monitoring of mining machines. However, additional work is required before the proposed systems can be utilized on a mine site. In addition to the need for reliable extrinsic sensor calibration procedures discussed in Section 3.7, periodic calibration procedures are also needed to adjust for gradual changes in sensor position and orientation caused by machine vibrations. One way to address this issue is to record and observe sensor images over time, in order to monitor for such changes by identifying reliable and constant features on the machine. Furthermore, risk mitigation techniques need to be investigated, including the use of two or more laser sensors at different locations on the machine to address potential problems caused by direct sun exposure or occlusion. The dipper tracking uncertainty estimation methods discussed in Section 3.6.3 can be used to evaluate risk and inform the operator when the dipper tracking confidence is low. These methods include the use of particle distribution as well as the ICP registration error to estimate dipper tracking uncertainty.

The SDP algorithm was shown to improve upon edge definition compared to other works in the literature, though some artefacts are still visible in object outlines (see Figure 4-25). Further investigation is required to identify the cause and find ways to remove these artefacts. In addition, one drawback of the proposed approach is that laser measurements of edge regions are eliminated during the rejection step, while such laser measurements can present valuable high-frequency details of the scene. Thus, removing them can cause loss of important details in the scene. It has been shown that the removal of edge laser measurements has improved the quality of the SDP depth images compared to the other works in the literature. However, in the future, alternative methods can be sought to replace the current rejection step, so that information from edge laser measurements can be made useful and incorporated into the resulting depth image.

Another limitation of the SDP algorithm is that it is not performing in real-time (1.5 – 40 sec, depending on upsampling ratio, as seen in Table 4-2). Many applications of depth-imaging require real-time performance (e.g., obstacle avoidance). Using parallelization to achieve real-time SDP performance is an appropriate step to follow this work. Parallelization has been shown to significantly improve the time efficiency of bilateral filtering [28], [108]. Beyond parallelization, other optimizations can also be investigated for reducing the computational complexity of the algorithm and obtaining real-time performance for large upscaling ratios.

The slow scan-rate of the laser sensor used in this work means that a 3D laser point-cloud of the scene cannot be acquired in a real-time frame rate. In this work, each 3D laser point-cloud was obtained in approximately one second. Thus, with this particular configuration, dynamic scenes that contain fast moving objects cannot be captured accurately. One way to address the slow laser scan-rate is to use faster (and more expensive) 3D laser scanners, such as a 3D Flash LIDAR, that obtain up to 500,000 measurements/sec compared to the 13,000 measurements/sec obtained by the laser sensor in this work. A high performance 3D Flash LIDAR generates 128x128 images at 30Hz, thus using a one megapixel camera image, SDP can upsample the Flash LIDAR images by a factor of 60x. Alternatively, the addition of video information (i.e., temporal dimension) could be used to compensate for the slow laser scan-rate and broaden the range of potential applications for this technique.

Finally, while this work discussed the potential applications for the SDP algorithm in rope shovel collision avoidance, the implementation of the proposed applications is yet to be done. Algorithms have been proposed that can be used for dipper tracking and arm geometry extraction (Chapter 3), load profiling and truck localization [112], as well as swing angle measurement [16], [113]. Further work is required to implement and evaluate the above methods. The workspace monitoring instrumentation described in Appendix A can be used to provide ground-truth data for validation and performance evaluation of the resulting technology.

References

[1] N. P. Himmelman et al., “Rope Shovel Collision Avoidance System,” in *Proceedings of Applications of Computers and Operations Research in the Mineral Industry (APCOM) Conference*, Vancouver, Canada, 2009.

[2] A. Bozorgebrahimi, R. A. Hall, and M. A. Morin, “Equipment size effects on open pit mining performance,” *International Journal of Mining, Reclamation and Environment*, vol. 19, no. 1, p. 41, 2005.

[3] E. Widzyk-Capehart, G. Brooker, S. Scheding, R. Hennessy, A. Maclean, and C. Lobsey, “Application of Millimetre Wave Radar Sensor to Environment Mapping in Surface Mining,” in *International Conference on Control, Automation, Robotics and Vision*, pp. 1-6, 2006.

[4] Mine Safety and Health Administration - MSHA, “MSHA’s Data Retrieval System,” Mar-2009. [Online]. Available: http://www.msha.gov/drs/drshome.htm. [Accessed: 01-Mar-2009].

[5] A. Stentz, J. Bares, S. Singh, and P. Rowe, “A Robotic Excavator for Autonomous Truck Loading,” *Autonomous Robots*, vol. 7, no. 2, pp. 175-186, 1999.

[6] G. J. Winstanley, K. Usher, P. I. Corke, M. Dunbabin, and J. M. Roberts, “Dragline automation: a decade of development: shared autonomy for improving mining equipment productivity,” *IEEE Robotics & Automation Magazine*, vol. 14, no. 3, pp. 52-64, 2007.

[7] M. Dunbabin and P. Corke, “Autonomous excavation using a rope shovel,” *Journal of Field Robotics*, vol. 23, no. 6-7, pp. 379-394, 2006.

[8] D. L. Wallis, R. Jefferey, P. M. Siegrist, and P. R. McAree, “Control systems modelling of a 2100BLE electric mining shovel,” in *Australian Mining Technology Conference*, Hunter Valley, Australia, 2006.

[9] E. Duff, “Accurate Guidance and Measurement for Excavators using a Laser Scanner,” Technical Report C14043, ACARP, Oct. 2006.

[10] A. H. Kashani, W. S. Owen, P. D. Lawrence, and R. A. Hall, “Real-Time Robot Joint Variable Extraction from a Laser Scanner,” in *IEEE International Conference on Automation and Logistics*, pp. 2994-2999, 2007.

[11] E. Duff, K. Usher, and P. Ridley, “Swing loader traffic control,” CSIRO ICT Centre, Brisbane, Australia, 06/192, Jan. 2006.

[12] J. M. Roberts, P. I. Corke, and G. J. Winstanley, “Development of a 3500-Tonne Field Robot,” *The International Journal of Robotics Research*, vol. 18, no. 7, pp. 739-752, Jul. 1999.

[13] J. Roberts, G. Winstanley, and P. Corke, “Three-Dimensional Imaging for a Very Large Excavator,” *The International Journal of Robotics Research*, vol. 22, no. 7-8, pp. 467-477, Jul. 2003.

[14] G. Brooker, R. Hennessey, C. Lobsey, M. Bishop, and E. Widzyk-Capehart, “Seeing through dust and water vapor: Millimeter wave radar sensors for mining applications,” *Journal of Field Robotics*, vol. 24, no. 7, pp. 527-557, 2007.

[15] M. Adams, S. Zhang, and L. Xie, “Particle filter based outdoor robot localization using natural features extracted from laser scanners,” in *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1493-1498 Vol.2, 2004.

[16] F. Lu and E. Milios, “Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans,” *Journal of Intelligent and Robotic Systems*, vol. 18, no. 3, pp. 249-275, Mar. 1997.

[17] R. Labayrade, C. Royere, D. Gruyer, and D. Aubert, “Cooperative Fusion for Multi-Obstacles Detection With Use of Stereovision and Laser Scanner,” *Autonomous Robots*, vol. 19, no. 2, pp. 117-140, 2005.

[18] W. Boehler and A. Marbs, “Investigating Laser Scanner Accuracy,” in *XIXth CIPA Symposium,* Antalya, Turkey, 2003.

[19] W. C. Stone et al., “Performance Analysis of Next-Generation LADAR for Manufacturing, Construction, and Mobility,” NIST Technical Report, 2004.

[20] A. S. Hall and P. R. McAree, “Robust bucket position tracking for a large hydraulic excavator,” *Mechanism and Machine Theory*, vol. 40, no. 1, pp. 1-16, Jan. 2005.

[21] D. W. Hainsworth, P. I. Corke, and G. J. Winstanley, “Location of a dragline bucket in space using machine vision techniques,” in *IEEE International Conference on Acoustics, Speech, and Signal*, vol. vi, p. VI/161-VI/164 vol.6, 1994.

[22] 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, pp. 7-42, Apr. 2002.

[23] M. Z. Brown, D. Burschka, and G. D. Hager, “Advances in computational stereo,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 8, pp. 993-1008, 2003.

[24] K. J. Yoon and I. S. Kweon, “Adaptive Support-Weight Approach for Correspondence Search,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, 2006.

[25] J. Sun, N.-N. Zheng, and H.-Y. Shum, “Stereo Matching Using Belief Propagation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 7, pp. 787-800, 2003.

[26] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister, “Stereo Matching with Color-Weighted Correlation, Hierarchical Belief Propagation, and Occlusion Handling,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 3, pp. 492-504, 2009.

[27] A. Klaus, M. Sormann, and K. Karner, “Segment-Based Stereo Matching Using Belief Propagation and a Self-Adapting Dissimilarity Measure,” in *International Conference on Pattern Recognition*, vol. 3, pp. 15-18, 2006.

[28] J. Dolson, J. Baek, C. Plagemann, and S. Thrun, “Upsampling Range Data in Dynamic Environments,” *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[29] S. Fuchs and G. Hirzinger, “Extrinsic and depth calibration of ToF-cameras,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-6, 2008.

[30] Q. Yang, R. Yang, J. Davis, and D. Nister, “Spatial-Depth Super Resolution for Range Images,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2007.

[31] J. Diebel and S. Thrun, "An Application of Markov Random Fields to Range Sensing," in *Proceedings of Conference on Neural Information Processing Systems (NIPS)*, Cambridge, MA, USA, p. 291--298, 2006.

[32] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint Bilateral Upsampling," *ACM Transactions on Graphics*, vol. 26, no. 3, p. 96, 2007.

[33] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," in *International Conference on Computer Vision*, Bombay, India, p. 839, 1998.

[34] D. Chan, H. Buisman, C. Theobalt, and S. Thrun, "A Noise-Aware Filter for Real-Time Depth Upsampling," in *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, Marseille, France, 2008.

[35] H. Andreasson, R. Triebel, and A. Lilienthal, "Non-Iterative Vision-Based Interpolation of 3D Laser Scans," in *Autonomous Robots and Agents*, pp. 83-90, 2007.

[36] S. Gould, P. Baumstarck, M. Quigley, A. Y. Ng, and D. Koller, "Integrating Visual and Range Data for Robotic Object Detection," in *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, Marseille, France, 2008.

[37] A. Riemens, O. Gangwal, B. Barenbrug, and R. P. . Berretty, "Multi-step Joint Bilateral Depth Upsampling," in *International Society for Optical Engineering*, 2009.

[38] P. J. Besl and H. D. McKay, "A method for registration of 3-D shapes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, no. 2, pp. 239-256, 1992.

[39] P. Biber, S. Fleck, and W. Strasser, "A probabilistic framework for robust and accurate matching of point clouds," *26th Pattern Recognition Symposium*, 2004.

[40] B. Rosenhahn, C. Perwass, and G. Sommer, "Pose Estimation of Free-Form Surface Models," in *Pattern Recognition*, pp. 574-581, 2003.

[41] Y. Zhu and A. C. F. Colchester, "Plane curve matching under affine transformations," *Vision, Image and Signal Processing, IEE Proceedings* -, vol. 151, no. 1, pp. 9-19, 2004.

[42] J. Salvi, C. Matabosch, D. Fofi, and J. Forest, “A review of recent range image registration methods with accuracy evaluation,” *Image and Vision Computing*, vol. 25, no. 5, pp. 578-596, May 2007.

[43] Z. Xie, S. Xu, and X. Li, “A high-accuracy method for fine registration of overlapping point clouds,” *Image and Vision Computing*, vol. 28, no. 4, pp. 563-570, Apr. 2010.

[44] R. Sandhu, S. Dambreville, and A. Tannenbaum, “Particle filtering for registration of 2D and 3D point sets with stochastic dynamics,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.

[45] J. Minguez, L. Montesano, and F. Lamiraux, “Metric-based iterative closest point scan matching for sensor displacement estimation,” *Robotics, IEEE Transactions on*, vol. 22, no. 5, pp. 1047-1054, 2006.

[46] Y. Chen and G. Medioni, “Object modeling by registration of multiple range images,” in *IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2724-2729, 1991.

[47] Z. Zhang, “Iterative point matching for registration of free-form curves and surfaces,” *International Journal of Computer Vision*, vol. 13, no. 2, pp. 119-152, Oct. 1994.

[48] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla, “Shape context and chamfer matching in cluttered scenes,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, p. I-127-I-133 vol.1, 2003.

[49] G. Turk and M. Levoy, “Zippered polygon meshes from range images,” in *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pp. 311-318, 1994.

[50] S. Rusinkiewicz and M. Levoy, “Efficient variants of the ICP algorithm,” in *Third International Conference on 3-D Digital Imaging and Modeling*, pp. 145-152, 2001.

[51] R. S. J. Estepar, A. Brun, and C.-F. Westin, “Robust generalized total least squares Iterative Closest Point registration,” in *International Conference on Medical Image Computing and Computer Assisted Intervention*, Saint-Malo, France, 2004.

[52] M. P. Dubuisson and A. K. Jain, “A modified Hausdorff distance for object matching,” in *Proceedings of the 12th International Conference on Pattern Recognition*, vol. 1, pp. 566-568, 1994.

[53] S. Tafazoli, “Identification of frictional effects and structural dynamics for improved control of hydraulic manipulators,” PhD Thesis, University of British Columbia, 1997.

[54] F. Ghassemi, S. Tafazoli, P. D. Lawrence, and K. Hashtrudi-Zaad, “An accelerometer-based joint angle sensor for heavy-duty manipulators,” in *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1771-1776 vol.2, 2002.

[55] A. H. Kashani, W. S. Owen, N. Himmelman, P. D. Lawrence, and R. A. Hall, “Laser Scanner-based End-effector Tracking and Joint Variable Extraction for Heavy Machinery,” *The International Journal of Robotics Research*, Sep. 2010.

[56] B. Ma and R. E. Ellis, “Surface-Based Registration with a Particle Filter,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2004*, pp. 566-573, 2004.

[57] D. Kim and D. Kim, “A novel fitting algorithm using the ICP and the particle filters for robust 3d human body motion tracking,” in *Proceeding of the 1st ACM workshop on Vision networks for behavior analysis*, Vancouver, British Columbia, Canada, pp. 69-76, 2008.

[58] E. D. Dickmanns and V. Graefe, “Dynamic monocular machine vision,” *Machine vision and Applications*, vol. 1, no. 4, pp. 223–240, 1988.

[59] D. B. Gennery, “Visual tracking of known three-dimensional objects,” *International Journal of Computer Vision*, vol. 7, no. 3, pp. 243–270, 1992.

[60] C. Harris, “Tracking with rigid models,” in *Active vision*, pp. 59-74, 1993.

[61] J. Rehg and T. Kanade, “Visual tracking of high dof articulated structures: an application to human hand tracking,” *Computer Vision—ECCV’94*, pp. 35–46, 1994.

[62] G. Welch and G. Bishop, “An introduction to the Kalman filter,” *University of North Carolina at Chapel Hill, Chapel Hill, NC*, 1995.

[63] M. Isard and A. Blake, “CONDENSATION—Conditional Density Propagation for Visual Tracking,” *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5-28, 1998.

[64] A. Doucet, N. De Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice*. New York: Springer Verlag, 2001.

[65] T. Yaqub and J. Katupitiya, “Laser scan matching for measurement update in a particle filter,” in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1-6, 2007.

[66] A. Almeida and Rui Araujo, “Real-Time Tracking of Moving Objects Using Particle Filters,” in *IEEE International Symposium on Industrial Electronics*, vol. 4, pp. 1327-1332, 2005.

[67] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House Publishers, 2004.

[68] J. V. Candy, “Bootstrap Particle Filtering,” *Signal Processing Magazine, IEEE*, vol. 24, no. 4, pp. 73-85, 2007.

[69] Tao Wu, Xiaoqing Ding, and Shengjin Wang, “Video Tracking Using Improved Chamfer Matching and Particle Filter,” in *International Conference on Conference on Computational Intelligence and Multimedia Applications*, vol. 3, pp. 169-173, 2007.

[70] G. Borgefors, “Hierarchical chamfer matching: a parametric edge matching algorithm,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 849-865, 1988.

[71] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*, Corrected. New York: Springer Verlag, 2002.

[72] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf, “Parametric correspondence and Chamfer matching: two new techniques for image matching,” in *International Joint Conference on Artificial Intelligence*, 1977.

[73] J. Denavit and R. Hartenberg, “A kinematic notation for lower-pair mechanisms based on matrices,” *ASME Journal of Applied Mechanics*, vol. 23, pp. 221, 215, 1955.

[74] L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators*, 2nd ed. Springer, 2001.

[75] A. H. Kashani, P. D. Lawrence, R. A. Hall, and N. P. Himmelman, “Active and Passive Sensor Fusion Using Bilateral Filtering on Depth Probability,” *(submitted: Oct, 2010).*

[76] Q. Yang, K. H. Tan, B. Culbertson, and J. Apostolopoulos, “Fusion of Active and Passive Sensors for Fast 3D Capture,” in *IEEE International Workshop on Multimedia Signal Processing*, Saint-Malo, France, 2010.

[77] J. Zhu, L. Wang, R. Yang, and J. Davis, “Fusion of Time-of-Flight Depth and Stereo for High Accuracy Depth Maps,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.

[78] D. Scharstein and R. Szeliski, “Middlebury Stereo Vision Page,” http://vision.middlebury.edu/stereo, 2010.

[79] C. Stiller, J. Hipp, C. Rössig, and A. Ewald, “Multisensor obstacle detection and tracking,” *Image and Vision Computing*, vol. 18, no. 5, pp. 389-396, Apr. 2000.

[80] S. Takahashi and B. K. Ghosh, “Motion and shape identification with vision and range,” *Automatic Control, IEEE Transactions on*, vol. 47, no. 8, pp. 1392-1396, 2002.

[81] L. Romero, A. Núñez, S. Bravo, and L. E. Gamboa, “Fusing a Laser Range Finder and a Stereo Vision System to Detect Obstacles in 3D,” in *Advances in Artificial Intelligence*, pp. 555-561, 2004.

[82] J. Miura, Y. Negishi, and Y. Shirai, “Mobile robot map generation by integrating omnidirectional stereo and laser range finder,” in *IEEE/RSJ International Conference on Intelligent Robots and System*, vol. 1, pp. 250-255 vol.1, 2002.

[83] Y.-G. Wu, J.-Y. Yang, and K. Liu, “Obstacle detection and environment modeling based on multisensor fusion for robot navigation,” *Artificial Intelligence in Engineering*, vol. 10, no. 4, pp. 323-333, Nov. 1996.

[84] H. Baltzakis, A. Argyros, and P. Trahanias, “Fusion of laser and visual data for robot motion planning and collision avoidance,” *Machine Vision and Applications*, vol. 15, pp. 92–100, 2003.

[85] V. Sequeira, E. Wolfart, E. Bovisio, E. Biotti, and J. G. M. Goncalves, “Hybrid 3D reconstruction and image-based rendering techniques for reality modeling,” in *Videometrics and Optical Methods for 3D Shape Measurement*, San Jose, CA, USA, vol. 4309, pp. 126-136, 2000.

[86] F. Boughorbal, D. L. Page, and M. A. Abidi, “Automatic reconstruction of large 3D models of real environments from unregistered data-sets,” in *Three-Dimensional Image Capture and Applications III*, San Jose, CA, USA, vol. 3958, pp. 234-243, 2000.

[87] S. F. El-Hakim, C. Brenner, and G. Roth, “A multi-sensor approach to creating accurate virtual environments,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 53, no. 6, pp. 379-391, Dec. 1998.

[88] M. D. Elstrom, “A stereo-based technique for the registration of color and LADAR images,” Master’s thesis, The University of Tennessee, Knoxville, 1998.

[89] P. W. Smith and M. D. Elstrom, “Stereo-based registration of range and projective imagery for data fusion and visualization,” *Optical Engineering*, vol. 40, no. 3, pp. 352-361, Mar. 2001.

[90] I. Stamos and P. E. Allen, “3-D model construction using range and image data,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 531-536 vol.1, 2000.

[91] P. Dias, V. Sequeira, J. G. M. Gonçalves, and F. Vaz, “Automatic registration of laser reflectance and colour intensity images for 3D reconstruction,” *Robotics and Autonomous Systems*, vol. 39, no. 3-4, pp. 157-168, Jun. 2002.

[92] V. Sequeira, K. Ng, E. Wolfart, J. G. M. Gonçalves, and D. Hogg, “Automated reconstruction of 3D models from real environments,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 54, no. 1, pp. 1-22, Feb. 1999.

[93] M. Lindner, M. Lambers, and A. Kolb, “Sub-Pixel Data Fusion and Edge-Enhanced Distance Refinement for 2D/3D Images,” *International Journal of Intelligent Systems Technologies and Applications*, vol. 5, no. 3, pp. 344–354, 2008.

[94] V. Garro, P. Zanuttigh, and G. M. Cortelazzo, “A New Super Resolution Technique for Range Data,” in *Associazione Gruppo Telecomunicazioni e Tecnologie dell’Informazione*, 2009.

[95] W. Hannemann, A. Linarth, B. Liu, G. Kokai, and O. Jesorsky, “Increasing Depth Lateral Resolution Based on Sensor Fusion,” *International Journal of Intelligent Systems Technologies and Applications*, vol. 5, no. 3-4, pp. 393 - 401, 2008.

[96] Q. Yang, “3D Reconstruction from Stereo/Range Images,” Master’s thesis, University of Kentucky, Lexington, 2007.

[97] U. Hahne and M. Alexa, “Combining Time-of-Flight Depth and Stereo Images without Accurate Extrinsic Calibration,” *International Journal of Intelligent Systems Technologies and Applications*, vol. 5, no. 3, pp. 325–333, 2008.

[98] J. Zhu, L. Wang, J. Gao, and R. Yang, “Spatial-Temporal Fusion for High Accuracy Depth Maps Using Dynamic MRFs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 899-909, 2010.

[99] S. A. Gudmundsson, H. Aanaes, and R. Larsen, “Fusion of Stereo Vision and Time-Of-Flight Imaging for Improved 3D Estimation,” *International Journal of Intelligent Systems Technologies and Applications*, vol. 5, no. 3/4, pp. 425-433, 2008.

[100] B. Bartczak and R. Koch, “Dense Depth Maps from Low Resolution Time-of-Flight Depth and High Resolution Color Views,” in *Advances in Visual Computing*, pp. 228-239, 2009.

[101] Y. M. Kim, C. Theobalt, J. Diebel, J. Kosecka, B. Micusik, and S. Thrun, “Multi-view Image and ToF Sensor Fusion for Dense 3D Reconstruction,” in *IEEE Workshop on 3-D Digital Imaging and Modeling (3DIM)*, 2009.

[102] Y. M. Kim and K. Zhu, “Super-resolution 3D Multiview Reconstruction Using Time-Of-Flight Depth Sensors,” Technical Report, Stanford University, 2008.

[103] B. Huhle, S. Fleck, and A. Schilling, “Integrating 3D Time-of-Flight Camera Data and High Resolution Images for 3DTV Applications,” in *3DTV Conference*, pp. 1-4, 2007.

[104] A. Harrison and P. Newman, “Image and Sparse Laser Fusion for Dense Scene Reconstruction,” in *Proceedings of the 7th Int’l Conference on Field and Service Robotics*, Cambridge, MA, 2009.

[105] P. Dias, V. Sequeira, F. Vaz, and J. G. M. Goncalves, “Registration and fusion of intensity and range data for 3D modelling of real world scenes,” in *International Conference on 3-D Digital Imaging and Modeling*, pp. 418-425, 2003.

[106] B. Huhle, T. Schairer, P. Jenke, and W. Straer, “Realistic Depth Blur for Images with Range Data,” in *LNCS Lecture Notes in Computer Science -- Proc. Dynamic 3D Imaging Workshop*, 2009.

[107] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical recipes in C: the art of scientific programming*, vol. 5. Cambridge University Press, 1992.

[108] Q. Yang, K.-H. Tan, and N. Ahuja, “Real-time O(1) Bilateral Filtering,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 557-564, 2009.

[109] J.-Y. Bouguet, *Camera Calibration Toolbox for Matlab*. California Institute of Technology.

[110] J. Canny, “A Computational Approach to Edge Detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 8, no. 6, pp. 679-698, 1986.

[111] H. Breu, J. Gil, D. Kirkpatrick, and M. Werman, “Linear time Euclidean Distance Transform Algorithms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 5, pp. 529–533, 1995.

[112] J. R. Borthwick, P. D. Lawrence, and R. H. Hall, “Mining Haul Truck Localization Using Stereo Vision,” in *Proceedings of the Robotics and Applications Conference*, vol. 664, p. 9, 2009.

[113] J. S. Gutmann and C. Schlegel, “Comparison of scan matching approaches for self-localization in indoor environments,” in *European Workshop on Advanced Mobile Robot*, pp. 61–67, 2002.

[114] S. G. van der Tas, “Data acquisition for an electric mining shovel pose estimator,” Technical Report DCT 2008.025, Eindhoven University of Technology, Jan. 2008.

[115] E. Duff, “Tracking a vehicle from a rotating platform with a scanning range laser,” in *Australasian Conference on Robotics and Automation*, University of Auckland, 2006.

[116] M. E. Green, I. A. Williams, and P. R. McAree, “A Framework for Relative Equipment Localisation,” in *Australian Mining Technology Conference*, 2007.

[117] F. Ghassemi, S. Tafazoli, P. D. Lawrence, and K. Hashtrudi-Zaad, “Design and Calibration of an Integration-Free Accelerometer-Based Joint-Angle Sensor,” *Instrumentation and Measurement, IEEE Transactions on*, vol. 57, no. 1, pp. 150–159, 2007.

[118] L. H. Lin, “Enhanced stereo vision SLAM for outdoor heavy machine rotation sensing,” Master’s thesis, University of British Columbia, 2010.

[119] Z. Zhang, “A flexible new technique for camera calibration,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330-1334, 2000.

Appendices

Appendix A: Rope Shovel Collision Avoidance System[7,8]

Introduction

Rope shovels are used extensively in open pit mining to extract material from the earth and load it into haul trucks. The rate at which they are able to extract and load the material is often the limiting factor in a mine's output, and as such, the shovels need to be run continuously in order to meet production targets. Unfortunately, the truck loading process is not without risk, as the dipper can collide with the haul truck during loading and self-collisions are possible between the dipper and its caterpillar tracks. Although collisions do not typically occur on a daily or even weekly basis, when they do occur they can result in serious injury to the truck driver and expensive downtime and machine repairs for the mine. A system that is capable of warning the shovel operator when the dipper is on a collision course with a haul truck or with the shovel tracks could significantly reduce the likelihood of collisions, and therefore be of significant value to a mine.

System Overview

During the loading process, we assume that the truck is parked beside the shovel and that the shovel's tracks are stationary. In order to avoid collisions during loading, we must determine the exact state of the shovel and the exact pose of the truck. The task of determining the configuration of the shovel and the truck has been divided into three subsystems: shovel arm geometry, shovel swing angle, and truck location. Figure 5-1 illustrates the shovel components such as the boom and the dipper handle, the swing

[7] A version of this section has been published in [1].

[8] This appendix presents an overview of the project that motivated this thesis. The shovel collision avoidance system discussed here consists of a number of other components that are not in the scope of this thesis.

angle (θ_1), and the arm geometry variables consisting of the dipper handle angle (θ_2), and the dipper handle extension (d).

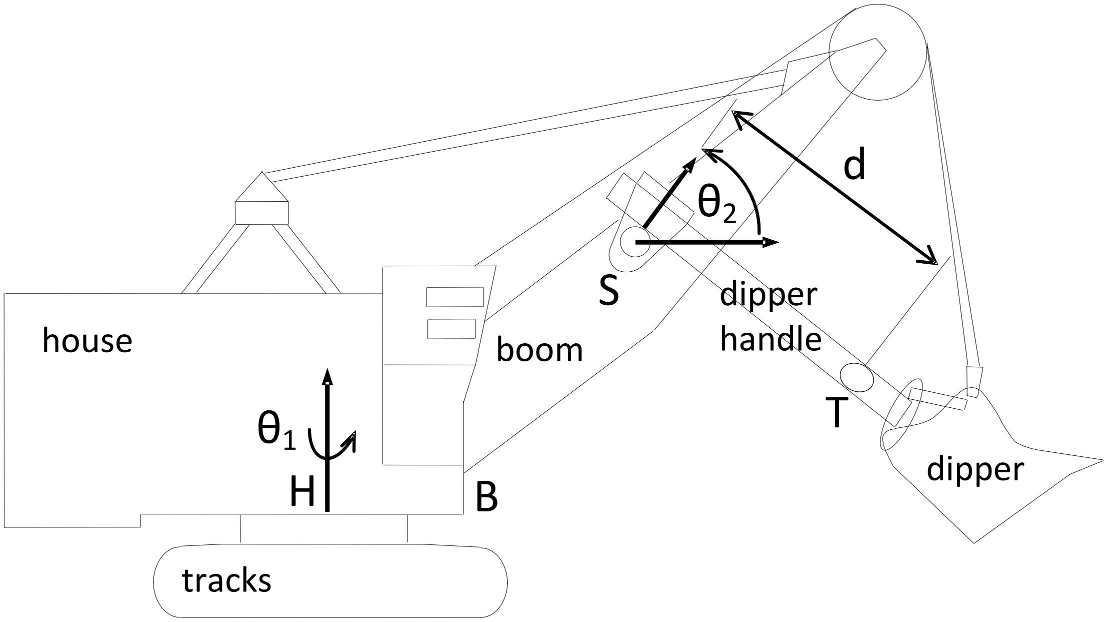


Figure 5-1: Rope Shovel Geometry Variables

The electric rope shovel's swing angle (θ_1), dipper handle angle (θ_2), and dipper handle extension (d) are outlined. Variables θ_2 and d are referred to in this work as the *arm geometry variables*.

This is not the first attempt to instrument an excavator or mining shovel [5], [7], [114] or to locate a haul truck [5], [115], [116]. It is, however, to the best of our knowledge, the first published work describing a sensing system for a real-time collision avoidance system for a full-scale electric rope shovel.

In order to reduce the costs associated with installation and maintenance, the collision avoidance system is designed such that all the subsystems are installed on a shovel. This allows the data collected by each subsystem to be easily gathered by a central computer where collisions will be predicted. The information collected could be used to warn the operator of an impending collision, and if no corrective action is taken, briefly override the operator's controls to prevent a collision. The following sections describe the hardware components and the data collected by each of the three subsystems. An overview of the hardware layout comprising all subsystems is shown in Figure 5-2.

Arm Geometry

Obtaining arm geometry variables is an important step towards building the collision avoidance system. The arm geometry of the shovel is determined by the angle of the boom, the extension of the dipper handle, and the angle of the dipper handle. The boom angle is set by adjusting the length of the boom suspension ropes whose length is kept constant during operation but can change slightly when the boom is lowered for maintenance and raised (joint *B* in Figure 5-1). The angle at which the dipper is attached to its handle is also adjustable, but stays constant during operation (joint *T* in Figure 5-1).

During typical operation the arm geometry is controlled by the crowd and hoist functions. The arm geometry can be directly measured in two ways. The hoist rope length and the dipper handle extension can be used to determine the location of the dipper. One difficulty that arises when measuring the hoist rope length is estimating the stretch in the hoist rope as it depends on the load and arm geometry. Alternatively, the angle of the dipper handle with respect to the boom can be used with the dipper handle extension to locate the dipper (θ_2 in Figure 5-1). By measuring the angle of the dipper handle with respect to the boom the uncertainty associated with stretch in the hoist rope can be avoided.

The angle of the dipper handle with respect to the boom can be measured between the saddle block and the boom. The saddle block (*S* in Figure 5-1) encapsulates the rack and pinion drive that connects the dipper handle to the boom, and keeps the dipper handle on the shipper shaft. As the dipper is moved, the saddle block pivots with the dipper handle, making it ideal for measuring the angle of the dipper. One problem related to measuring the dipper handle angle on the saddle block is that the saddle block can twist back and forth on the dipper handle as the hoisting direction changes or as the direction of the torque applied to the shipper shaft changes. The way the saddle block moves on the dipper handle is shown in Figure 5-3.

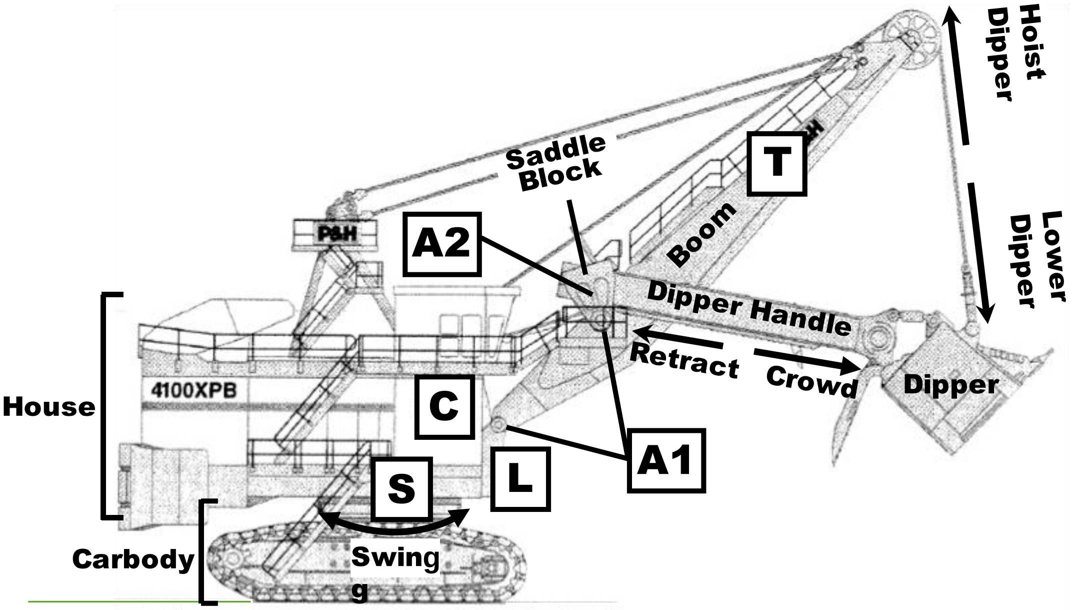


Figure 5-2: Shovel Movements and Hardware Locations

A1) accelerometers placed on the rotary joints, A2) rangefinder placed on the saddle block prismatic joint, C) central computing system located in house underneath operator cab, L) laser scanners placed below boom foot; one planar laser scanner for arm geometry extraction, and one 3D laser scanner along with camera for workspace monitoring, S) stereo camera placed underneath house looking inwards, and T) truck localization stereo camera mounted on boom.

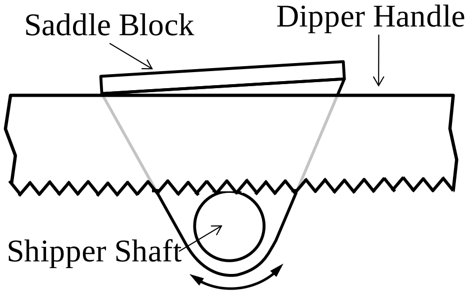


Figure 5-3: The Saddle Block

The saddle blocks are not rigidly connected to the dipper handle and can rotate back and forth depending on how much space is left between the dipper handle and the saddle block slide bars.

We estimate that to prevent a bucket from hitting the truck, a bucket-positioning safety zone of 30cm would be required (see Section 3.2.1). This is consistent with Brooker et al. [14] who used 1% of the maximum range as the required accuracy.

Traditionally in mining machines, joint angle or extension sensors are used to measure joint variables, which in turn are used for forward kinematics to determine the workspace location of an end-effector [20]. In this work, the arm geometry was measured both directly and indirectly: Directly by placing joint angle and extension sensors at each joint, and indirectly by tracking the position of the dipper with a depth imaging sensor (e.g. a laser scanner) and using inverse kinematics to determine the dipper handle angle, dipper handle extension, and hoist rope length. Advantages of using joint sensors are:

- Complex algorithms are not required for obtaining the arm geometry.
- The processing simplicity makes the required computing power minimal.
- Visibility of the dipper is not required as it is with an imaging sensor where occlusion can degrade the arm geometry measurements. Occlusion can occur, for instance, when the dipper is lowered into the truck canopy during dumping.
- Joint sensors may be less sensitive to environmental factors such as dust and lighting, compared to non-contact imaging sensors that often require clear line of sight.

Advantages of using a cab-mounted non-contact imaging sensor are:

- Cab-mounted non-contact sensors are less vulnerable to mechanical damage [6].
- Linkage singularities cannot cause numerical instabilities in forward kinematics calculations when using a laser scanner, as they can for joint sensors [20].
- In some excavators, the dipper is not rigidly connected to the body, making forward kinematics impossible [21].

Joint Sensor-Based

The first method for measuring the arm geometry used sensors placed at each of the joints. Pairs of biaxial accelerometers were used to measure the angles of the rotary joints (boom and dipper handle angles) and a laser rangefinder was used to measure the extension of the prismatic joint. The accelerometer-based angle measurement system used is part of a payload measurement system for hydraulic shovels called LoadMetrics, designed and manufactured by Motion Metrics International [117]. To measure the joint angle one biaxial accelerometer is placed on the proximal link of the joint and the other is placed on the distal link of the joint. Figure 5-4 shows the accelerometers installed on the boom joint. A second pair of accelerometers was installed on the saddle block joint to measure the dipper handle angle. The accelerometers are connected to a Motion Metrics International LoadMetrics computer that runs an algorithm which determines the difference in angle between the two sensors with a worst case error of 1.6°. An offset is used to adjust the measurement according to the placement of the sensors.

The laser rangefinder used to measure the extension of the dipper handle was a SICK DT 500. It has a worst case error of 3mm between 0.2m and 18m on a surface with 6% reflectivity (black). The rangefinder was mounted on the saddle block aiming along the dipper handle, away from the dipper. A target was mounted on the end of the dipper handle opposite the dipper. The rangefinder was connected to the same computer used for the angle sensors via an RS422 serial connection. An offset was used to adjust the distance measured by the rangefinder to make it correspond to the dipper handle extension. Figure 5-5 shows the installation of the rangefinder and target on a shovel.

Even though the measurement error in the joint sensors as well as the twisting issue in the saddle block degrade the quality of the joint measurements, the results can still be used effectively to verify and evaluate the novel non-contact arm geometry extraction system proposed in the next subsection.

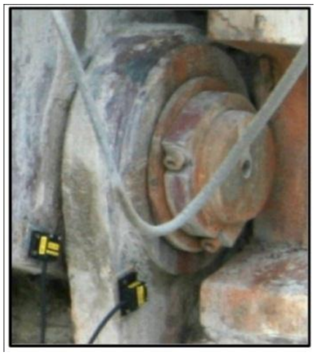


Figure 5-4: Accelerometers

The biaxial accelerometers mounted to boom joint.

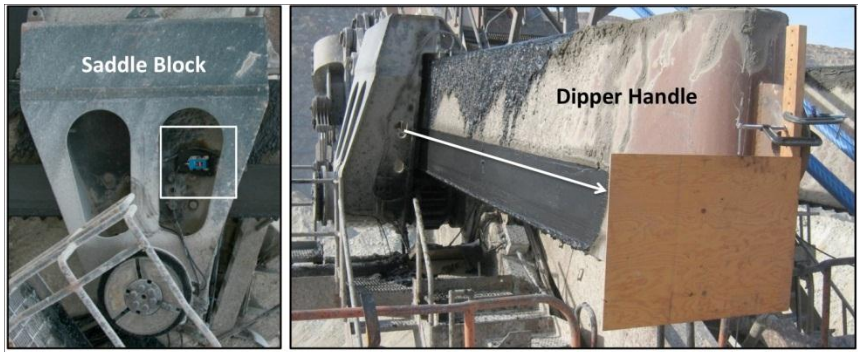


Figure 5-5: Dipper Extension Sensor

The left image shows the installation of the rangefinder (white bounding box) on the saddle block. The right image shows a target used on the end of the dipper handle with a white arrow depicting the laser beam path.

Non-Contact Laser Scanner-Based

Rather than using joint sensors, a cab-mounted imaging sensor can be used to track the dipper. Laser scanner-based arm geometry extraction is not affected by problems such as the stretch in the hoist rope or the twist in the saddle block. This subsystem focuses on the development and implementation of a method for estimating the dipper location using a planar laser scanner. To be operational on a mining shovel and be applicable to the problem at hand, the sensor must be: reliable with a Mean Time Between Failures (MTBF) of greater than 1 year [14], accurate within 1% of the measurement range [14], and capable of range measurements up to 50m [3].

Computer vision, radar, and laser scanning have all been used for object tracking as well as dipper localization. While radars can provide the reliability and range, they are often slower (i.e., capture fewer measurements per second) or more expensive than laser scanners [14], [15]. Winstanley et al. [6] successfully utilized a laser scanner for their research after finding cameras to be less reliable than laser scanners when faced with environmental factors such as rain, dust, and shadows.

The laser scanner was mounted vertically underneath the boom facing forward (i.e. towards the dipper). In this position, the laser scanner maintained a consistent view of the dipper and occlusions were minimal. A typical scan plane of the laser scanner at the given position is provided in Figure 5-6.b) where the shovel boom, dipper, ground, and truck are evident in the point-cloud. The laser scanner provides 40 readings per second via an RS422 serial connection with a maximum viewing range of 80m, resolution of 1cm, and a mean (μ) accuracy of 5.0cm with a standard deviation (σ) of 1.0cm.

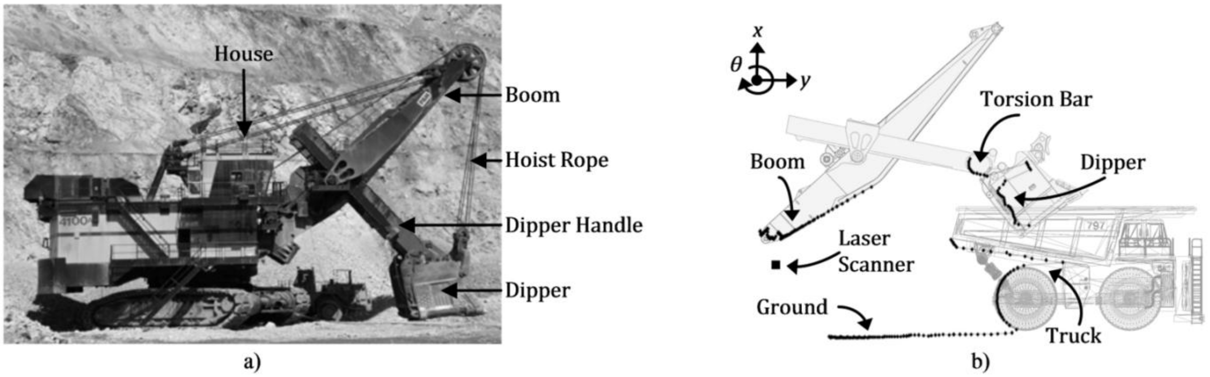


Figure 5-6: P&H Electric Rope Shovel

a) A P&H 2800 Electric Rope Shovel. b) A sample laser point-cloud superimposed on equipment diagrams. The laser scanner is mounted vertically underneath the shovel boom. The laser scanner's point-cloud, shown by a series of dots, represents a contour of the environment. Note that the above point-cloud is not taken during a normal digging cycle. Here, the laser sees the back of the truck, whereas during a normal loading sequence, the truck is located sideways and the laser only sees its side.

Swing Measurement

Swing angle, the angle between the house and the lower carbody (shown in Figure 5-2), is one of the measurements required for the computation of the dipper's 3D position in the workspace. Without swing angle, the collision avoidance system cannot determine the dipper's position and cannot predict common collisions such as the collision of the dipper with the excavator tracks or external objects such as a truck. Unfortunately, many shovels do not have a swing angle measurement system in place and one must be designed and retrofitted for this project. The desired swing angle accuracy of the system is 0.25 degrees (worst-case) which corresponds to approximately 10cm horizontal error in dipper localization [118].

Encoder-Based Swing Measurement

One method for obtaining the swing angle is to use an encoder to measure the change in rotation angle that the swing motor shaft makes. For this purpose, a 100 line BEI HS35F encoder was attached to the shaft of one of the swing motors. In this configuration each quadrature increment on the encoder corresponds to a house rotation of 0.002°. The encoder was mounted to a stub shaft which was attached to the motor shaft, rising through the brake assembly. The brake assembly was removed to allow the stub shaft to be mounted to the motor shaft. The stub shaft had to be trued to the motor shaft to minimize wobble when the motor was spinning to prevent damage to the encoder. This laborious process would have to be repeated any time motor maintenance was required. A flexible coupling could be used to connect the encoder to the motor shaft but this would require a more complex mounting assembly which would in itself impede maintenance.

Camera-Based Swing Measurement

Given the drawbacks of using an encoder, a novel camera-based swing angle sensor which can be easily retrofitted and does not get in the way of regular maintenance was investigated. The proposed swing angle sensor consists of a Point Grey Bumblebee 2 stereo camera mounted on the bottom skirt of the excavator housing, looking down, toward the center of the carbody (Figure 5-7). As the house rotates, the camera will rotate with the house and revolve around the stationary carbody, seeing differing views of the carbody (Figure 5-8). An Ethernet cable carries digital video images from the camera to a computer in the house. The computer analyzes the images in real-time and determines the angle from which the camera is viewing the carbody.

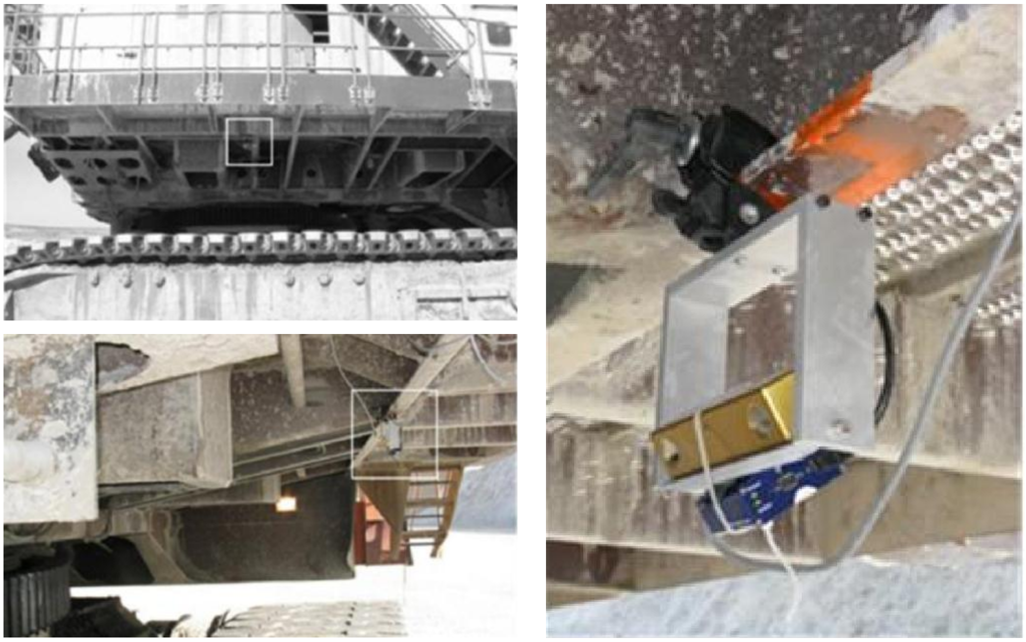


Figure 5-7: Swing Camera Installation

The left image shows the camera attached to bottom skirt of the house, its position indicated by white bounding box. The middle image shows how the camera is aimed downwards, toward the carbody centre. The right image shows a closeup of the camera.

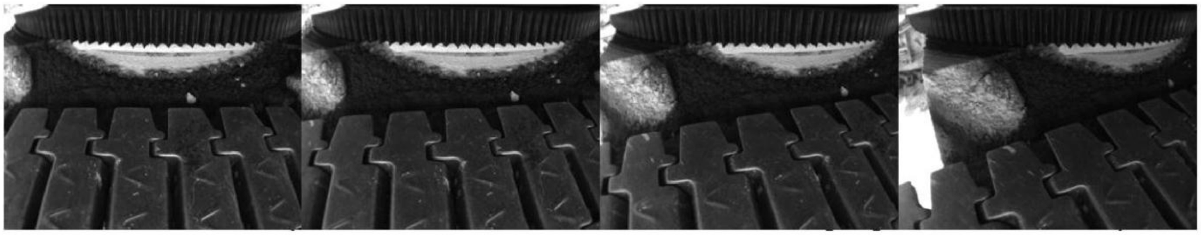


Figure 5-8: Sample Images Taken by the Camera as the House Rotates Clockwise

For easy retrofitting, we are designing a system such that the camera need not be exactly positioned or angled when mounted to the excavator. As long as the lower carbody covers most of the view of the camera, the system will function properly. Further, the swing sensor should not need a prior model of the excavator as there are many variations of shovels. Thus the swing angle sensor automatically learns the 3D features on the carbody and calibrates the camera position and orientation with respect to the swing rotation axis.

Truck Localization

Once the shovel is fully instrumented, the task still remains of precisely locating the haul truck to avoid contact. Like the system developed by Stentz et al. [5], we wish to determine the truck's exact position and orientation (i.e., pose), which can be fully described by six variables – three for translation and three for rotation. However unlike [5], we require the system to work in real-time without imposing new requirements or restrictions on how the shovel operator digs or loads a truck. The system must work in the background to quickly and accurately determine the six pose variables. The truck pose estimation must be accurate to within 25cm in worst case [112].

As stated previously, a goal was to place all equipment for the collision avoidance system on the shovel. This requirement restricts us from installing beacons or specialized GPS equipment on the trucks. As such, the use of a shovel-mounted 3D imaging sensor was seen as the best solution. Several 3D imaging sensors, namely stereo cameras, laser scanners and millimeter-wave radar units, were considered. Laser scanners and radar units were attractive because they deliver highly accurate 3D measurements, but unfortunately, also have slow image acquisition rates and low resolution images [3], [5]. Stereo cameras, meanwhile, deliver high-resolution 3D images with a high acquisition speed. However, stereo cameras suffer from the fact that their depth accuracy falls off exponentially with the distance between the camera and the measured object. This stems from the fact that stereo cameras work using triangulation.

This subsystem was developed using point-clouds obtained from a stereo-camera sensor, but the proposed algorithm can be applied to other depth-sensing technologies as well. Once more accurate depth sensors are available that acquire full 3D images in real-time, the stereo-camera system can be replaced to obtain more accurate truck localization results.

We chose to mount the camera high on the side of the shovel boom, far from potential sources of dust and damage. As the camera is not intended for use in outdoor or rough environments, we constructed a waterproof, heated camera enclosure and secured the camera on shock mounts. Figure 5-9 shows the camera's location on the boom, its enclosure, and its view of a haul truck.

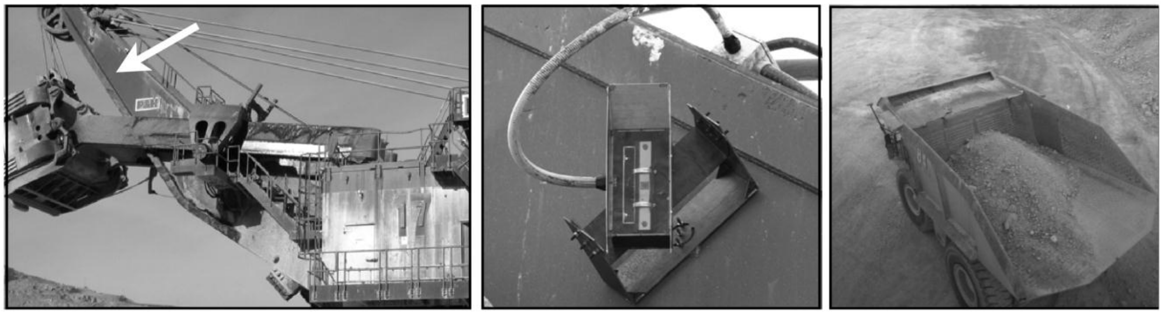


Figure 5-9: The Truck Localization Stereo-Camera Installation

The left image shows a white arrow pointing at the location on the boom where the stereo camera is mounted. The middle image shows the stereo camera, mounted on the boom, in its protective enclosure. The right image shows the view of a haul truck delivered by the stereo camera.

The data produced by a stereo-camera is called a "point-cloud", which is the same as a regular image except that for each pixel the precise (x; y; z) location relative to the camera (and hence the shovel) is known. For the system to function, it must know which areas of the cloud represent the truck, as they must be avoided, and which represent the payload of the truck, as they may be gently brushed against by the dipper. Additionally, the (x; y; z) measurements of the point-cloud will contain errors which must not confuse or overwhelm the system.

What we wish to achieve is to be able to use this data to locate the truck from distances on the order of fifteen meters with a worst-case accuracy of 25cm. Furthermore, this must be accomplished quickly enough to operate within the real-time time constraints of a collision avoidance system. We believe that the described hardware platform and resultant data will provide the basis for such a system.

3D Workspace Monitoring

3D imaging of the machine workspace can provide valuable information such as obstacle positioning, and dig surface profiling. A depth-measurement sensor is required that can provide accurate 3D images of the entire scene in real-time. A 3D laser scanner can provide accurate information, but it is slow in capturing the entire 3D scene and it creates very sparse point-clouds. A fusion of a laser scanner and a camera, however, has the potential for creating an accurate system that provides spatially dense results in real-time [31].

A laser scanner and a camera were mounted under the shovel boom. The objective of this subsystem is to fuse the two sensors in real-time to create 3D images that can be used for obstacle positioning and dig surface profiling.

Discussion and Conclusions

Four measurement subsystems have been described for a system to prevent collisions between the shovel's dipper and a haul truck, or between the shovel's dipper and its own tracks. Together, the four subsystems can provide to a collision avoidance system the position of the dipper, the position of the haul truck with respect to the shovel's carbody, and the position of other obstacles. The sensing subsystems are for:

- *Arm Geometry*, which measures the dipper pose relative to the revolving frame. Two different approaches to obtain this information have been developed: a joint sensor based method which can be compared to the results from a laser scanner based method.
- *Swing Angle*, which measures the angle between the revolving frame and the fixed frame. This system relates the dipper pose as found from the Arm Geometry subsystem, to the track body frame. We have also developed two approaches here: an encoder-based angle measurement subsystem which can be compared to the camera-based swing angle measurement subsystem.

- *Haul Truck Localization*, which measures the pose of a haul truck with respect to the revolving frame. A point-cloud-based approach to localize the truck has been developed.
- *Workspace Monitoring*, which uses real-time 3D imaging of the workspace to locate obstacles and monitor changes in the digging surface. The sensors used for this subsystem include a swinging laser scanner and a camera.

All the sensors have been developed so that they could be easily retrofitted. They are all attached externally to the shovel without requiring the shovel to be disassembled or extensively modified. A practised team could install all the hardware components in one 12 hour shift.

These sensors have been installed and tested on a shovel at the Highland Valley Copper mine. The current installation described here forms the test bed for determining the most appropriate set of sensors, and for the development of the collision avoidance system itself.

Appendix B: Laser Scanner and Camera Calibration

The MATLAB Calibration Toolbox [109] was used in this work to obtain the alignment parameters between the laser scanner and the camera. The steps of the algorithm are as follows:

1. A checkerboard is presented at different locations and orientations in the scene, and images from the camera and the laser scanner are recorded.

   The 3D laser scanner is configured to scan the scene at small angular increments of 0.1°. Along with its depth image, the laser scanner generates a grayscale reflectivity image which is used in this work for the calibration algorithm.
2. For all images, the outer corners of the checkerboard are manually selected. This guides the calibration algorithm in identifying the corresponding feature-points between the two sensor images.
3. Once the corresponding feature-points are detected, the extrinsic calibration parameters can be estimated using [119].

The above calibration method has been used by others in the literature to register laser scanners and cameras [77], [97], [103]. The variance in the results from each image set shows that the method is not very accurate [97]; the low spatial resolution of the laser scanner, and its unreliability in measuring depth edge (i.e., the beam-splitting effect on edges causing depth averaging), contribute to the lack of accuracy. Thus, in this work, 30 different image sets were captured with the checkerboard at varying positions and orientations, in order to improve the resulting calibration accuracy.

As shown in Chapter 4, without a highly accurate calibration it is still possible to integrate the camera and laser data and obtain accurate high-resolution 3D images [97]. The SDP's *rejection* step proposed in Section 4.2.2 addresses the calibration error problem to minimize its impact on the resulting high-resolution 3D images.