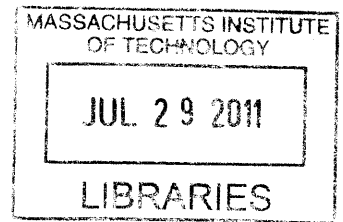


Localization and Tracking of Parameterized
Objects in Point Clouds

by

Robert Truax



Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

ARCHIVES

Master of Science in Mechanical Engineering

at the

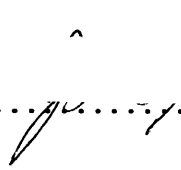
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2011

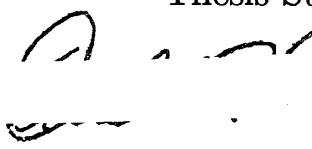
© Massachusetts Institute of Technology 2011. All rights reserved.

Author 

Department of Mechanical Engineering
May 19, 2011

Certified by 

John J. Leonard
Professor of Mechanical and Ocean Engineering
Thesis Supervisor

Accepted by 

David Hardt
Ralph E. and Eloise F. Cross Professor of Mechanical Engineering
Chairman, Department Committee on Graduate Theses

Localization and Tracking of Parameterized Objects in Point Clouds

by

Robert Truax

Submitted to the Department of Mechanical Engineering
on May 19, 2011, in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

Abstract

This thesis focuses on object recognition and tracking from three dimensional point cloud renderings of dense range and bearing data. Sensors like laser rangefinders and depth cameras have become increasingly popular in autonomous robotic applications. A common task is to locate and track specific objects of interest located somewhere in the point cloud. This often introduces a tedious network of heuristics to build objects from identified primitives or an intractable high dimensional search space. Through a parameterized object model and certain relaxation functions, a likelihood based view of the data can be used to accomplish these goals with increased performance and reliability. Improvements in mathematics and convergence properties have shown that this method can be realized in real time.

Thesis Supervisor: John J. Leonard

Title: Professor of Mechanical and Ocean Engineering

Acknowledgments

My wife Elizabeth for her steadfast support.

John Leonard for advice and help throughout the process.

Robert Platt for inspiration and help developing the idea of relaxation methods.

Garratt Gallagher for help understanding and using ROS software.

Contents

1	Introduction	9
2	Background	13
2.1	Vision Based Object Detection	14
2.2	Reconstruction from Range Data	15
2.3	Autonomous Manipulation and Perception	16
3	Parameterized Model Method for Point Clouds	19
3.1	Problem Statement	19
3.2	Relaxation Function	21
3.3	Other Relaxation Functions	24
3.4	Implementation Details	24
4	Experiments and Results	29
4.1	Planar Illustration	29
4.2	Static Three Dimensional Case	31
4.3	Dynamic Case	35
5	Conclusions	39

Chapter 1

Introduction

A key component of robotic systems is sensing and interpreting the world. Industrial robotics has shown extraordinary ability to manipulate objects once knowledge of their position and orientation has been determined. This is usually accomplished by tightly structuring the environment. In many circumstances robots have no sensing of their world. This forces robots to be inflexible and unable to handle unexpected events or objects. Current research focuses on giving robots more flexibility to deal with more general situations and unstructured environments. The major challenge for robotics is to take noisy, uncertain measurements of the world and translate them into a model sufficient to carry out a task.

An example of this can be seen in a robotic porter built for Agile Robotics at MIT seen in figure 1-1. This robot's task was to locate a stack of boxes on a provided pallet, select and grasp those available, and stack them in another location without human intervention. It used a nodding laser scanner and force sensors in its arms to accomplish this task. While existing methods performed well for sparse cases, it was found that occlusions and closely stack boxes presented difficulties. For this reason, the ideas presented in this thesis were developed to increase the performance and reliability of the robot[32, 35].

Point clouds have become the major sensory input for perceiving complicated environments. Sensors that produce ranges and bearings to points throughout the environment can stitch together their data into a large representation of surfaces in the

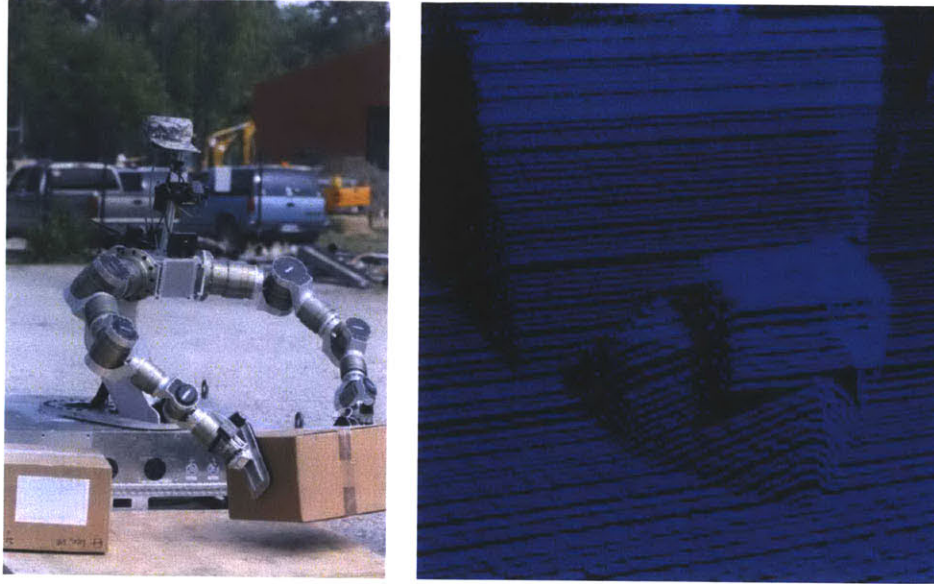


Figure 1-1: Scenario in which a robot must estimate the pose and shape of modeled objects with high confidence.

environment, called point clouds. By processing these data, robot can avoid obstacles, determine surface normals and curvature, and identify objects. Point clouds are commonly produced from scanning laser range finders, though recently the Microsoft Kinect sensor offers dense fast point clouds at a low cost. This suggests that point cloud based algorithms will become more important as this technology spreads.

Current methods for processing point cloud data focus on a bottom up approach. Since point clouds can contain tens of thousands of points, it is desirable to break up the problem of identification of objects into a hierarchy of primitives. The general approach is to first extract low level information in addition to range and bearing, including local surface normals and clusters. These are then formed into object primitives like planes, spheres, and cylinders. These primitives can then be assembled into simple objects, e.g. cuboids or tables, which can then be interpreted into the robotic task. This system of information processing gradually reduces the sensor complexity while increasing usefulness. Unfortunately, each of these steps requires tuning of special parameters and heuristics for proper function. If an algorithm fails to successfully locate boxes in a scene, it is difficult to determine if the issue lies with the computation of surface normals, over accommodating primitive fitting, or a low threshold on

how perpendicular planes must be to constitute a box. While this method is useful in its flexibility and generality, it is desirable to use another method when performing certain high level object identification and manipulation tasks.

Object identification and manipulation on a mobile robotic platform is desirable but difficult task for a robot, while comparatively easy for a human. The first thing to do would be to enter a room and locate all objects of interest. Once this information is in hand, many solutions exist to successfully manipulate the object to carry out a task. A bottom up approach may work, but it would be more desirable to use the knowledge of what the objects of interest look like to inform how to use the point cloud data. A good algorithm will also have a low number of parameters to tune and be robust to clutter and obstruction of other objects in the environment.

This thesis will demonstrate a method that will localize and track objects in interest inside point clouds. This method will have several key advantages over existing methods. It will use geometric knowledge of the objects of interest to inform its search, significantly reduce the number of parameters to tune, and be more robust to noise and occlusions in the environment. After localization and tracking, many solutions exist to manipulate objects or determine their dynamic characteristics. This will help allow robots to autonomously interact with their environment to perform desired tasks. This thesis will have the following structure:

Chapter 2 explores previous work and background in the object detection and robotic manipulation literature.

Chapter 3 describes the mathematical framework of the object detection and localization problem. It will also develop the models and algorithms used to accomplish an object localization task.

Chapter 4 presents the results from several experiments and simulations demonstrating the effectiveness and advantages present in the method presented over the current state of the art method.

Chapter 5 concludes and summarizes the information presented in this thesis, as well as discuss applications and future work.

Chapter 2

Background

Point cloud depth maps have become the sensory input of choice for robotics in unstructured environments. Sensor systems including nodding LIDARs, time-of-flight cameras, and stereo cameras provide rich three dimensional data without the ambiguity that comes with bearing only systems like ordinary cameras. Using these data is the focus of current research in the robotics field. A common task for a robot is manipulation: locating objects of interest and interacting with them. As industrial robotics has shown over the past decades, the interaction step is well explored problem, and many methods exist to accomplish a manipulation task once the object's pose is known. However, determining where an object of interest is remains a difficult problem.

The problem associated with manipulation is slightly different than the general problem of perception. For manipulation, it is not necessary to characterize and label every part of a sensor scan. It is sufficient merely to identify the location and orientation, or pose, of particular objects of interest. Therefore, a good algorithm will find these objects while ignoring unmodeled aspects and be robust to noise and occluding obstacles. Once the object knowledge is in hand, other manipulation algorithms can take over. In addition, the goal of a good algorithm will be to rely as little as possible on many complicated parameters or heuristics.

This chapter will explore the literature for object detection and tracking. This can be broken down into three mains sections: approaches based on robot vision,

reconstruction from range data, and perception methods for autonomous robotics.

2.1 Vision Based Object Detection

Early object detection explored the more difficult problem of producing three dimensional reconstructions from camera based data. This addressed the more difficult problem of forming a representation of the environment from the bearing only data cameras provide. Horn and Brooks determined shape from light intensity values in camera images and building up continuous surfaces and using variational approaches[7, 18]. Horn's book *Robot Vision* also describes several methods for using cameras to describe the environment. He uses photometric methods for shape detection and surface determination. He also develops optical flow methods for developing structure from motion, building a picture of a static environment from a series of moving camera images. The main method used in this book for object detection and identification is using extended Gaussian images which use a kind of histogram of object surface normals to uniquely identify classes of objects.[17] A more recent survey of these methods and several comparative results can be found in a paper by Zhang et. al.[39]. Other methods determine shape from images through texture [21, 22, 15] or shape through changing the focal length [13, 9]. Many of these methods require sophisticated mathematical approaches, are slow to converge, and still result in output data equivalent to modern depth sensors. The desire to focus on a hierarchy of increasing data sophistication may come from work in this area, where simply arriving at a depth map required a significant amount of work. Now that these data are easily obtained, other algorithms become more feasible and useful.

Much work has explored identifying and tracking humans in visual and range data. The most visible result of this work has been motion capture technology employed by the entertainment industry to record the movements of actors and transmit them to animated characters. The literature in this area is exhaustive. Some good starting points include survey papers by Forsyth et. al. [14] and Moeslund et. al. [27]. Many examples in the area strongly supports the model specific approach. One approach

models people in images as two dimensional models, so called “cardboard” people which breaks each independently moving limb into a rectangle to track. Others use a more accurate three dimensional model of a human built from cylinders and spheroids connected with joints. Measured data are then used to determine the inverse kinematics of the human. Many of these methods focus on subtracting the background from an image and using the model to determine where edges and occlusions lie. Others use probabilistic models to track humans by building a shape database, initializing models with images, then simulating how most humans move to select the best guesses supported by subsequent data. More recent methods have used deformable models of humans to better refine their estimates.[1] The SCAPE (Shape Completion and Animation for PEople) system scans people under various poses to learn how muscles and joints change under different motions[2]. This gives strong support for an object of interest model fitting approach to detection and tracking. While this closely resembles the main idea of this thesis, these particular approaches are meant to be used offline with careful human supervision. The issues necessary to integrate these methods into an autonomous robotic situation have not been explored. This thesis will adapt these ideas for mobile robots to locate and interact with objects if interest.

2.2 Reconstruction from Range Data

There are many examples of using three dimensional range data in the literature. Key examples include work by Besl and Jain[3, 4], who developed techniques for object recognition from range images, creating a system that analyzed surface curvature and classified critical points; (2) Viola and Wells, who developed gradient descent techniques for alignment of 3D models with 2D and 3D data sets based on maximizing the mutual information; and (3) Besl and McKay, who developed a method for registration of 3-D shapes based on iterative closest point (ICP) matching[35].

Other research has focused on merging multiple sets of range data into a single coherent frame. Many approaches have sought to first find an underlying model, then fit the data provided from multiple vantage points onto their surfaces. Early

approaches focused on generalized cylinders [8, 24], superquadrics [33, 31], and physically based models [26, 11]. These references are much more relevant since they match range and bearing data to models of the environment. Many of these use iterative closest point to match two data frames or a single data frame to an estimated model, similar to the approach used in this thesis. However, these methods tackle the larger problem representing everything in the environment. They seek a total reconstruction through these simple models. In mobile robotics and robot manipulation, this step is wasteful and unnecessary. This thesis has a more focused approach, applying these ideas exclusively to objects of interest, seeking to reinforce those that match the object well and reject those that do not fit. This requires less processing and improved performance by throwing away irrelevant information.

2.3 Autonomous Manipulation and Perception

One of the most basic tasks for mobile robotics is obstacle avoidance. A great deal of research has been conducted determining what areas a robot may and may not enter without collision. Most recently, occupancy grids or maps like the OctoMap by Wurm *et al.* which label each volume piece of space as either occupied or unoccupied have become more efficient and useful[38]. They have proven to be effective in mapping and navigation of autonomous robots in several papers[10, 12]. These have a main advantage of being completely general and able to model arbitrary environments. However, they do not extract any meaning from filled and unfilled voxels. Alternate methods must be used on these occupancy grids to determine which pertain to objects of interest. The data from these maps are easily converted point clouds by extracting those points that lie on the surface between occupied and unoccupied cells.

Some of the most recent and extensive point cloud processing software has been designed with autonomous robotics in mind. The most common strategy that has been developed is a bottom-up approach. Planes or curves are determined from individual points in the data first, and then the planes or curves are fit to known models. For example, Vosselman *et al.* adopt this approach for reconstructing ground planes

based on aerial point cloud data. After extracting roof planes using a combination of a 3-D Hough transform and least squares plane estimation, the resulting planes are fit to *a priori* roof models [37, 36]. Liu *et al.* take a similar approach to model the objects encountered by a mobile robot: EM is used to locate planes in the scene and the planes are mapped onto a known set of polygons [20]. Rusu *et al.* apply a similar approach for the task of locating objects in a household kitchen environment. Planes in the scene are found using MLESAC [34]. Plane parameters are estimated using regression, neighboring planes are merged, and the resulting planes are matched to a given polyhedral model [28].

Rather than matching point data to models known *a priori*, an alternative approach is to reconstruct object models directly from point data. If only primitive planes or curves are of interest, then RANSAC-based methods have been demonstrated to be very effective [34, 30]. If more complex objects are to be modeled, RANSAC methods can be used to find the subset of points that match a desired object. Given these points, a superquadric can be used to model the geometry of the object surface [5]. Alternatively, Marton *et al.* propose using RANSAC methods to identify shape primitives and then modeling objects as arbitrary constellations of primitives [25]. The return to these simple primitives from the models listed above indicate the need for a simplified representation of the environment so that autonomy can be more easily programmed. Fitting approaches followed by heuristic building of objects allow for human-free operation while providing a wealth of parameters to tune to improve robotic performance. However, the complicated and resource intensive algorithms described earlier are ill-suited to the limited resources and time available for autonomous robotic tasks.

A key difficulty with the compositional methods described above is that while shape primitives are used to locate complex objects, there is no flow of information in the other direction: object information does not inform the search for primitives. In manipulation, we are often interested in deciding which grasp action has the greatest chance of succeeding. If object models are given, then we must decide which modeled object in the scene is *most likely*: our objective is to find the object model(s) that

maximize the likelihood of the point cloud data. However, in order to maximize the likelihood of matching points in the cloud to object surfaces, it is necessary to associate each hypothesized object with a subset of points. For example, Liu *et al.* solve the data association and the optimization problems simultaneously using expectation maximization (EM) [20]. The need to associate points with objects makes object localization more complex because a bad association will cause localization to fail[35].

The algorithms developed by Rusu are the most easily accessible due to his recent development of the Point Cloud Library[29]. The wide distribution, open source format, and good documentation suggest that this will become a popular toolbox for research robotics. This has allowed for easy comparison and integration with the software written for this thesis. Therefore, this class of algorithms based on compositional methods will be the primary point of comparison for performance, since it represents the state of the art.

This chapter has shown the previous work in the object detection literature, including early vision approaches, three dimensional reconstruction with range data, and perception methods for use in autonomous robotic platforms. The next chapter will outline the method proposed in this thesis to solve the object detection and tracking problem.

Chapter 3

Parameterized Model Method for Point Clouds

This chapter will outline the algorithm proposed in this thesis. It will first define the problem mathematically, then develop the relaxation function used, and end with a discussion of implementation details.

3.1 Problem Statement

This algorithm will localize and track objects of interest in range and bearing data from common robotics sensors. These include scanning laser range finders, or LIDARs, time-of-flight cameras, or stereo camera platforms including the Microsoft Kinect. A common method is to assemble collections of the measurements into point clouds.

$$Z = \{z_0, z_1, z_2, \dots, z_N\} \quad (3.1)$$

$$z_i = \{x_i, y_i, z_i\} \quad (3.2)$$

A point cloud Z is a collection of individual range and bearing measurements z_i which are usually represented in a Cartesian coordinate system $\{x, y, z\}$. Each of these points are measurements of surfaces in the environment, a subset of which are

objects of interest, though many are environmental clutter. For the moment, consider only those points which result from objects of interest: $Z_o \subseteq Z$. From these points, the goal is to maximize the likelihood of a set of objects with flexible parameters given this point cloud.

$$C_{max} = \arg \max_C \Pr(C|Z_o) \quad (3.3)$$

$$C = \{o_1, o_2, \dots\} \quad (3.4)$$

$$o_i = O(P) \quad (3.5)$$

C is a collection of objects o_i which can be described from a parameter list P through an object generation function O . For example, a cuboid can be represented from a parameter list with a translation in \mathbb{R}^3 , a rotation in $SO(3)$, and the lengths of its three sides in \mathbb{R}^3 , resulting in nine parameters. Unfortunately, the preprocessing step to label each measurement point is nearly as difficult as finding all objects of interest, and therefore does not help. A way to address this issue is to represent the environment generalized as a collection composed entirely of objects of interest so every measurement can be used:

$$H_{max} = \arg \max_H \Pr(H|Z) \quad (3.6)$$

where H is a larger collection of objects representing the whole environment up to a maximum number. This is also an unsatisfactory solution because the dimensionality of the problem has grown significantly. To adequately represent the environment, it is likely many more objects will be necessary. This significantly increases the difficulty in optimization. One might hope that H could be broken down into individual objects and each object could be optimized separately. First, it can be shown using Bayes' rule that solving equation 3.6 is equivalent to solving:

$$H_{max} = \arg \max_H \Pr(Z|H) \quad (3.7)$$

Assuming each measurement is independent of each other, this can now be broken into a form using individual objects.

$$\Pr(Z|H) = \prod_i^{|Z|} \Pr(z_i \in S(H)) \quad (3.8)$$

$$= \prod_i^{|Z|} [1 - \Pr(z_i \notin S(H))] \quad (3.9)$$

$$= \prod_i^{|Z|} [1 - \prod_{o \in H} \Pr(z_i \notin S(o))] \quad (3.10)$$

This shows that the probability of a measurement being on a surface in H is directly related to the measurement *not* being on any of the surfaces. Therefore, a straightforward separation of the problem is not possible.

3.2 Relaxation Function

To make the problem more tractable, it is desirable to use a relaxation function to approximate 3.6 while requiring few parameters and does not require a preprocessing step. This will lead to a function whose maxima are near the maxima of equation 3.6, but may have spurious local maxima that do not correspond.

A key fact to notice is that when H has reached its maximum with respect to the measurements in equation 3.6, each object inside H will be maximized with respect to the points around it:

$$o_{max} = \arg \max_o \Pr(Z_o|o) \quad (3.11)$$

where o is a candidate object and Z_o is the subset of points in Z which result from measurements on the real object. In the case of laser scanners, depth cameras, or other range and bearing measurements, it is common to model sensor errors with a normal Gaussian distribution. While errors in range exhibit different characteristics from errors in bearing, this thesis will assume a simplified measurement model:

$$\Pr(z_i|o) = \mathcal{N}(d, \sigma) \quad (3.12)$$

The probability of an individual measurement z_i given an object o is a zero mean Gaussian distribution with error d and standard deviation σ . d is the distance from the measurement point in Cartesian coordinates to the object surfaces visible from the sensors origin. This assumes that errors are equally likely in all Cartesian directions, and show no difference in the range or bearing directions. By using this model for equation 3.11, each point z_i will be assigned to its closest object. However, the relaxation function should be separable into individual objects so the entire collection need not be considered at once and no preprocessing assignment is required. The function should match equation 3.12 when measurement points are nearby, but little or no effect otherwise:

$$\Pr(z_i|o) = \begin{cases} \mathcal{N}(d, \sigma) & \text{for } d < d_{max} \\ \mathcal{N}(d_{max}, \sigma) & \text{for } d \geq d_{max} \end{cases} \quad (3.13)$$

This successfully integrates the requirements necessary for the relaxation function. Points nearby a candidate object have an ordinary measurement effect, and points farther than d_{max} have a uniform low probability and will have little effect on the object probability. This function requires no preprocessing of assignments of points and can be separated for every candidate object considered. d_{max} can be chosen based on properties of the sensor noise such that an actual measurement outside d_{max} is highly unlikely. 3σ is a common choice, but could change depending on the quality of the sensors. Unfortunately, this function does not correspond to an actual probability distribution since the integral is infinite instead of one. However, this functions will find the configurations sought for H , and similar assumptions for the measurement model have been successfully used by Blake and Zisserman.[6]

There is a special case for the sensors used. Points that are far away from the candidate object, but inside the shadow cast by the object from the sensor origin, are highly unlikely and should be somehow penalized in equation 3.13. In this case, the distance d to the object should instead be distance to the shadow region: $d_s = D(z_i, o, s)$ where s is the location of the sensor. An example can be seen in figure 3-1.

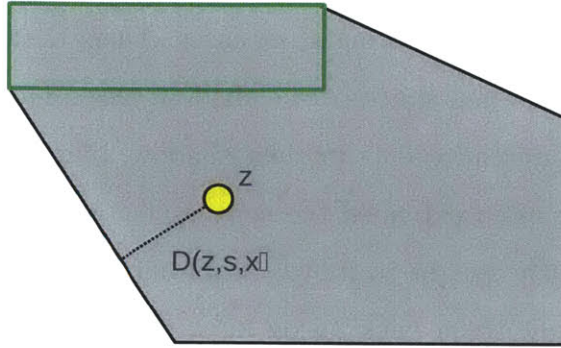


Figure 3-1: Calculation of the distance, $D(z_i, o, s)$, of point z_i from the region not occluded by the object. The object hypothesis is illustrated in green; the region occluded by the object with respect to the source is shaded.

Therefore, equation 3.13 can be expanded to:

$$\Pr(z_i|o) = \begin{cases} \mathcal{N}(d_s, \sigma) & \text{if } z_i \text{ in shadow} \\ \mathcal{N}(d, \sigma) & \text{if outside shadow and } d < d_{max} \\ \mathcal{N}(d_{max}, \sigma) & \text{otherwise} \end{cases} \quad (3.14)$$

The final probability of a candidate object for all measurement points is therefore:

$$\Pr(Z|o) = \prod_{z_i \in Z} \Pr(z_i|o) \quad (3.15)$$

By sampling the space around the sensor with candidate objects, then ascending the gradient of equation 3.14 for each object for a given measurement set, each candidate will find a local maximum. Several of these will be spurious, but many should correspond to correct configurations of objects in the space. This is the main algorithm of this thesis.

3.3 Other Relaxation Functions

Previous work on this problem shown in chapter 4 did not follow the same relaxation functions as above.[35] The functions used for these results are similar, but less efficient. Instead of combining the inside shadow case with outside shadow case as in equation 3.14, a hierarchy was used. First, candidate objects were evaluated based on the shadow case, ignoring other cases. Then, the outside shadow case would be projected into the null space of the shadow case. This put the first priority on objects having no measurements in their shadow, then slowly fitting points around it afterward. This often extended the convergence time necessary for objects, since candidates frequently sought to move points out of their shadow in one step, then immediately brought them back inside for a better fit the next step to repeat the process. This is why the new relaxation function shown in 3.14 was developed.

The last difference was in how the individual measurements are combined. In equation 3.14, the cases far away are handled by thresholding the probability distribution. This allows the joint probability to be calculated with an intuitive multiplication. Previous work sought to preserve a true probability function applying the Gaussian distribution measurement model to all points. To combat far points having a huge effect on the gradient of an object, individual measurement probabilities were not multiplied as in equation 3.15. Instead, the function maximized the *number* of high probability points on the object's surface, and therefore added them together.

Though these previous examples are different from the algorithm described in the previous section, they share similar characteristics, and help to illustrate the performance of this method, especially when compared to other solutions.

3.4 Implementation Details

Several details should be more closely examined to successfully implement this algorithm. A good parameterization of the object should be determined, recomputed terms should be recognized and preserved in future calculation, and a suitable opti-

mization method should be found.

The parameters chosen for the object can greatly affect the outcome of the convergence. An early parameterization of the cuboids used in this thesis was poorly chosen. A particular corner of the cuboid was selected as the origin. The translation, rotation, and side lengths of the cuboid were defined around this point. This had the advantage of simplifying several terms when calculating the derivatives of the distance function. If a measurement point was closest to the origin point instead of a point on the opposite side, the box rotation and side lengths had no effect since variations in these parameters did not move the origin point. Unfortunately, this also introduces an asymmetry in the object which skews the convergence properties when the cuboid is in a particular rotation. Since measurement points in certain areas had no effect on certain directions in the gradient, they did not affect the object as much as other points leading to poor convergence. A symmetric parameterization was then chosen, with the origin being the center of the cuboid. This ensured that no part of the cuboid was favored over another.

Rotation representation should also be chosen carefully. The parameterization should be continuous and differentiable throughout all rotations so the gradient can always be calculated, and should have only three degrees of freedom, so standard optimization tools may be utilized. Euler rotations are common, but are not differentiable at certain key points. This is commonly referred to as gimbal lock. Unit quaternion representations solve the gimbal lock problem and are continuously and differentiable, but generally have four degrees of freedom with one constraint of unit size. This could be used, but would require a constrained optimization algorithm to account for the superfluous degrees of freedom which is more difficult to implement. However, an axis-angle formulation, with an extension of an exponential map, can solve all these issues.

Rotations may be parameterized by three axis-angle variables: $\mathbf{v} = \{r_x, r_y, r_z\}$. The axis of rotation is a unit vector in the direction of \mathbf{v} and the rotation in radians is the size of \mathbf{v} . A size of zero corresponds to the identity rotation. This has the minimum parameters necessary for rotations, which makes it ideal for optimization,

Algorithm 1 Overview of implementation of thesis.

```
for  $\mathbf{b} \in B$  do  
  repeat  
    for  $\mathbf{p} \in Z$  do  
       $r \leftarrow D(\mathbf{p}, \mathbf{b})$   
       $\mathbf{j} \leftarrow \frac{\partial}{\partial \mathbf{b}} r$   
       $\mathbf{R} \leftarrow r$   
       $\mathbf{J} \leftarrow \mathbf{j}$   
    end for  
     $\mathbf{b} \leftarrow \mathbf{b} + [\mathbf{J}^T \mathbf{J} + \lambda \text{diag}(\mathbf{J}^T \mathbf{J})]^{-1} \mathbf{J}^T \mathbf{R}$   
  until converged  
end for
```

but is not useful in this form. The most useful representation for rotating vectors and differentiation is a 3×3 rotation matrix. This may be obtained by first converting to a unit quaternion from the axis-angle, then converting to a rotation matrix. Each step is easily computable and differentiable. The conversion from a quaternion to a rotation matrix is differentiable for all rotations, but redundant on variables. However, the first conversion solves the problem by using axis angle. The differentiation on this step is less straightforward, but through an appropriate exponential map described by Grassia[16] this too is easily determined. This combination allows for an good parameterization for rotation.

An outline of the implementation is provided for a reference in algorithm 1. The algorithm first goes through each box parameter vector \mathbf{b} in a collection of hypotheses B and processes the point cloud Z for every point \mathbf{p} until the parameter updates converge. A distance r to the box as described in equation 3.14 is first computed. Then the Jacobian of r is found with respect to the box parameters \mathbf{b} . These are appended into matrices \mathbf{R} and \mathbf{J} until each point has been processed. Then a Levenberg-Marquardt optimization is performed with an appropriately chosen λ . The parameter vector is updated and the process is repeated until the optimization reaches convergence. For this thesis, a trust region approach was chosen. If an update to a box parameter vector result in a better cost, λ was halved, resulting in a more quadratic Gauss-Newton optimization. Otherwise, λ was double, resulting in a more fixed step gradient descent[23, 19]. Notice that solving the maximum probability in equation

3.14 is equivalent to solving a least squares distance problem for the measurement points and the object surfaces. This simplifies the calculation necessary to determine r and \mathbf{j} .

It is important to note that the inner for loop will reuse a great deal of information about the parameters and derivatives relating to the box. Taking care to store these common terms appropriately can significantly reduce the computational load. Indeed, it reduced processing time from several minutes for the initial static cases shown to real time operation of moving boxes in dynamic cases.

For static cases, B was usually initialized by a uniform distribution of small, unrotated boxes throughout the area of interest. For dynamic cases, a static case was first performed to provide good initial estimates of objects in the environment. Each iteration of the algorithm performed for a new point cloud frame used the previous parameter estimates to initialize B . This was sufficient since frames were captured fast enough that the change in box parameters did not bring it outside of the area of convergence of the previous estimate. Though it was not explored in this thesis, a motion model updated with a simple Kalman filter could use successive box estimates as positional measurements and better predict where the box will be in the next time frame. This would improve convergence time and be more robust to fast moving objects. In fact, once successive pose estimates have been found, any dynamic property of interest can be found through well known methods.

This chapter has described the algorithm proposed in this thesis. The next chapter will demonstrate the effectiveness and performance of this algorithm on several experiments and simulations.

Chapter 4

Experiments and Results

The algorithms described in chapter 3 have been implemented for several example and test cases. Early examples derive from a previous implementations of an earlier algorithm, while later examples demonstrate the current algorithm on more difficult problems. While these first examples are not implementations of the proposed algorithm, they exhibit nearly the same properties and are therefore useful in understanding the algorithm’s mechanics and performance.

4.1 Planar Illustration

Figure 4-1 illustrates the localization process in the plane. The point cloud (the green circles in Figure 4-1) was collected indoors using a Hokuyo UTM laser scanner mounted in a fixed configuration in front of a collection of three rectangular boxes. The objective is to locate a maximum likelihood rectangular objects in the scene. Since this is a planar problem, the hypothesis space for a single rectangle is five dimensional (position in \mathbb{R}^2 , orientation in $SO(2)$, and the extents of the two sides in \mathbb{R}^2),

$$x = (p_x, p_y, d_x, d_y, \theta)^T$$

where p_x and p_y are the coordinates of the center of the rectangle, d_x and d_y are the extents of the rectangle sides, and θ is the orientation of the rectangle. This particular

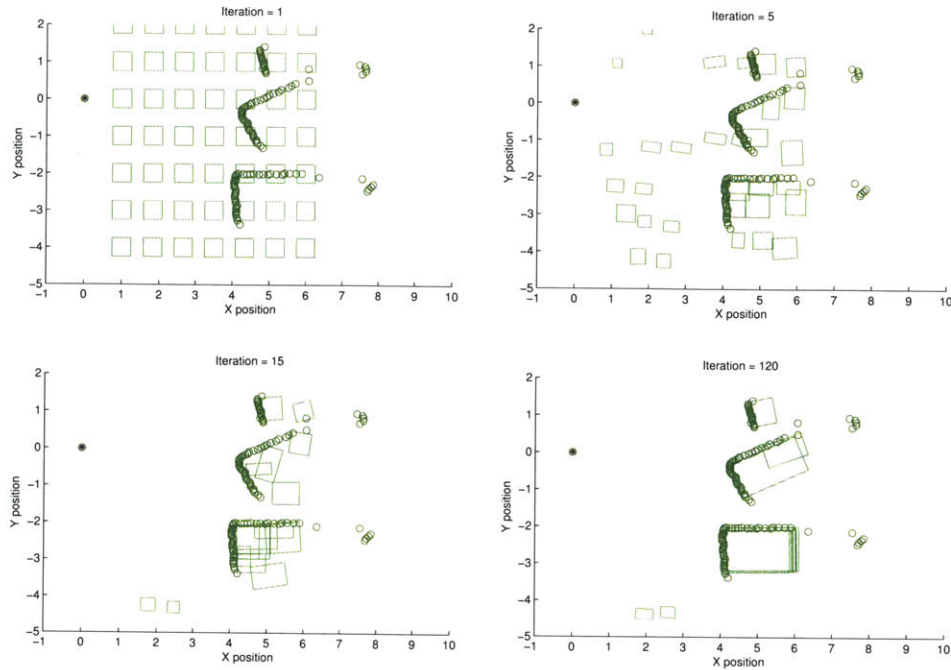


Figure 4-1: Illustration of localization in two dimensions for a collection of rectangles. Green boxes represent hypotheses, laser return points are circles, and the laser source is located at the star in the upper left corner.

implementation follows the gradient of the prioritized probability function with fixed steps.

Figure 4-1 shows four frames from the planar localization example. The starting sampling of hypothesis assumes nothing about the actual locations of the objects. It begins with uniformly spaced samples of identical small size. As the iterations progress, the high-priority relaxation of points in the shadow causes the hypotheses to migrate toward regions within the occluded regions in the point cloud. Hypotheses that do not migrate quickly enough shrink because the decreasing side extents decreases the number of points in the hypotheses occlusion regions. Hypotheses are removed if their likelihoods fall below a threshold. Once within the occluded regions in the point cloud, hypotheses slowly migrate toward configurations that maximize the number of points on their sides. After many steps, hypotheses stop moving in any meaningful way but do not converge to the same state. This is due to the fixed step gradient descent method of solving this function.

Notice that some of the hypotheses converge to maxima that do not correspond

to object locations. This reflects the fact that optimizations of the relaxed likelihood functions can find *any* local minimum. Not all optima of the combined relaxations correspond to objects in the scene. However, maxima that do not match the measurements well are given a low likelihood and can be removed.[35]

4.2 Static Three Dimensional Case

To demonstrate the advantages of the relaxation function algorithm over other point cloud tools developed by Rusu[29] based on RANSAC methods, a direct comparison was conducted. A nodding Hokuyo UTM laser was used to take data from three examples scenes containing objects of interest (boxes) and other occlusions (ground, walls, and cylinders). Each “sweep” of the laser took approximately five seconds and collected approximately 10^5 points. After restricting points to a bounding box around the work area and subsampling so that no two points were closer than about a centimeter, the point cloud was reduced to about 1800 points. While the algorithm maintained its properties by including more points, these steps greatly reduced the computational load.

In general, cuboids or boxes are described by a nine dimensional object configuration space: a translation in \mathbb{R}^3 , rotation in $SO(3)$, and size in \mathbb{R}^3 . in this experiment, objects on interest are boxes lying flat on the ground or on other boxes. Therefore, their states may be represented as:

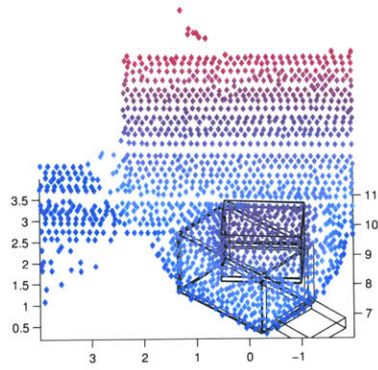
$$x = (p_x, p_y, p_z, d_x, d_y, d_z, \theta_z)^T$$

where $\{p_x, p_y, p_z\}$ are the coordinates of the box center, $\{d_x, d_y, d_z\}$ are the extends of the box along its principal axes, and θ_z is the rotation of the box about the z axis. Other rotations are not considered since they would results in boxes that did not lie flat on the ground or on other boxes.

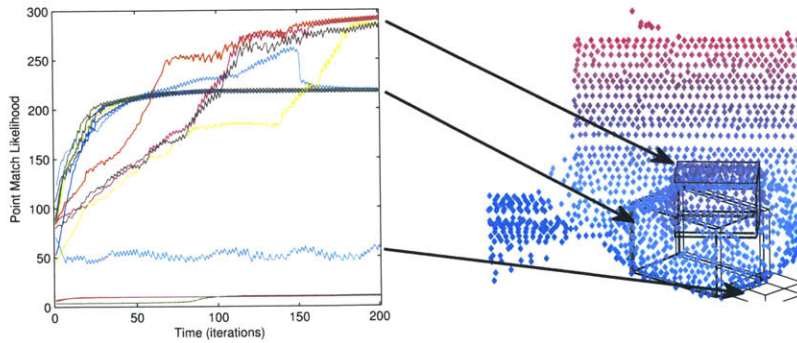
Figure 4-2 illustrates the process of localizing a pair of stacked boxes (shown in Figure 4-2(a)) in the seven dimensional object configuration space. The process begins



(a)



(b)



(c)

Figure 4-2: For a scene (a) a final state of the hypotheses (b) is illustrated. The time history of the point match likelihoods for for each hypothesis are plotted in (c) with arrows to their corresponding locations in space. Notice a spurious maxima below the floor but with low likelihood when compared to other hypotheses.

with 15 uniformly seeded object hypotheses. Figure 4-2(b) illustrates the hypotheses (shown in black) in their final configuration after they have converged to the maxima of the relaxed likelihood functions. Notice that the hypotheses cluster around the true locations of the two boxes. Figure 4-2(c) illustrates the optimization process in terms of hypothesis likelihood over time. Initially, all hypotheses have similarly low likelihoods. Over time, the likelihood of the boxes rises and ultimately clusters in two high-likelihood configurations. Once again, the hypotheses cluster but do not converge to the same state due to the fixed step gradient descent used. This can also be seen on the sawtooth patterns in the probability time history as the states step across a local minimum, then head backward.

As discussed in chapter 1, the primary localization methods used in robotic manipulation scenarios are compositional methods where modeled objects are located by first identifying shape primitives using methods such as RANSAC, Hough transforms, EM, *etc.* and then identifying the objects in terms of those primitives. The general approach is related to what is proposed proposed by Rusu [28]: a variant of RANSAC is used to find planes in the image, the planes are merged and the overall parameters of the plane are estimated using regression, and boxes are hypothesized to exist in places where the appropriate combination of planes is found.

The particular object configurations were chosen to cause difficulties in conventional methods by placing object planes near each other and covering key features such as edges with occluding objects. Figures 4-3, 4-4, and 4-5 illustrates the comparison. The three figures show an image of the scenario on the left, the box located by the compositional method in the center (the green outline superimposed on the point cloud), and the objects found by the relaxation method on the right. In Figure 4-3, the compositional method is confounded by the cylinder that prevents the algorithm from finding planes sufficiently nearby to be considered a box. In Figure 4-4, the compositional method finds only a single box that extends to the floor because it overestimates the size of the planes. Finally, in Figure 4-5, compositional methods fail to find enough planes to locate all boxes.

One way to improve performance of the compositional method on the above might

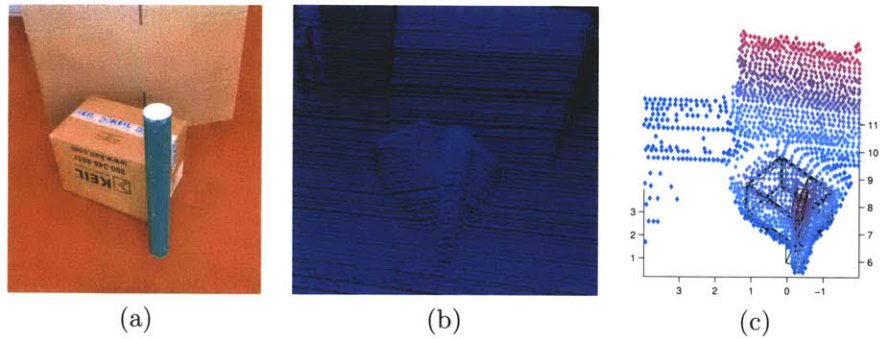


Figure 4-3: A single cylindrical obstacle in front of a box (a) causes a failure for the RANSAC method (b) but is located in the relaxation method (c).

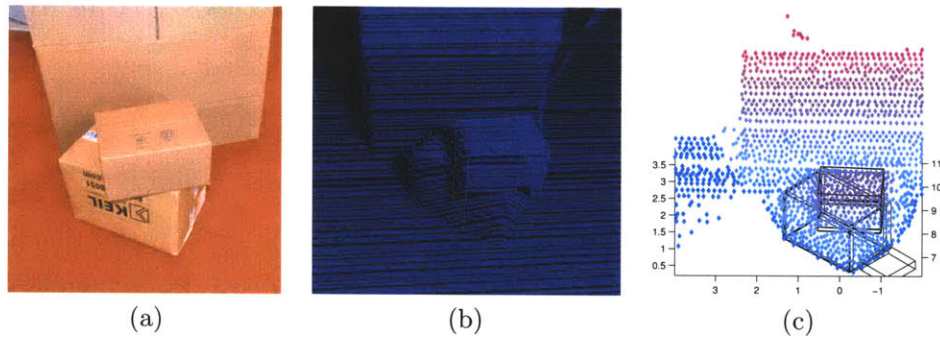


Figure 4-4: Two boxes where one edge aligns with the face of another (a) causes a bad hypothesis with the RANSAC method (b) but proper dimensions are found in the relaxation method (c).

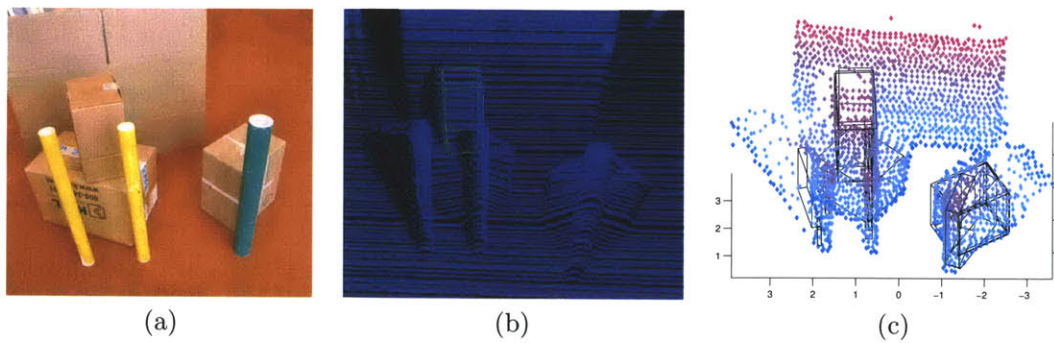


Figure 4-5: A crowded scene with many boxes and obstacles (a) interferes with all but one box in the RANSAC method (b) but poses minimal challenge to the relaxation method (c).

be to modify the plane-finding parameters so that a larger number of planes are found. However, this would risk finding boxes where none exist.

A potential failure of the relaxation approach in the scenarios above is that the occluding poles in the image are misinterpreted as boxes. This is a consequence of the optimization process. Since all points in the cloud are assumed to belong to a modeled object, the poles are interpreted as boxes. Of course, since the poles are not really boxes, they are given a low likelihood compared to other hypotheses and can be identified as poor box candidates.[35]

4.3 Dynamic Case

The performance of the relaxation based method has been demonstrated for the localization of objects in point clouds, but this section will demonstrate the performance of a tracking task. In order to handle this more difficult case, a completely new implementation was developed using the relaxation described in equation 3.14 and for cuboids in the full nine dimensional configuration space. Speed was increased by precomputing and storing common gradient terms and convergence was improved by using Levenberg-Marquardt numerical optimization to find the best solution to 3.14. This allowed for realtime processing of a single moving object.

Due to the higher required bandwidth for moving objects, the scanning laser used in previous experiments does not have a high enough bandwidth to capture an object moving at a reasonable speed. A Microsoft Kinect sensor was used instead. The Kinect is a camera based sensor that correlates an infrared image of the environment with a projected infrared “texture” imposed on the scene. This allows the sensor to provide a dense depth field measurement at 30 Hz, fast enough to conduct a dynamic experiment. A small box was taped to horizontally mounted wheel and spun by hand at slow speed of about 1 rotation every five seconds. This was recorded by the Kinect at 4 Hz to accommodate the write speed of the computer. The point cloud was then subsampled to around 3600 points. An initial guess of the box was supplied to mimic the guesses given by the static implementation of the algorithm. The results

of the algorithm can be seen in figure 4-6. This rotating box presents many problems similar to those in the previous section. As it turns, edges become fuzzy and disappear altogether, and new planes appear. A particularly challenging frame can be seen in frame (n) with unusually high noise on the sides of the box. Compositional methods would struggle to characterize these sides as planes as illustrate in the previous section. The relaxation algorithm is robust to this noise, since the model of the object as a whole is considered. A time history of the parameters can be seen in figure 4-7.

Figure 4-7 (a) shows the translational components of the box centroid. The slow sine wave seen can be attributed to the fact that the box was not centered on the wheel on which it was turning. This caused the centroid to move in a circle, producing the rough shape shown. The noisy spikes shown in figure 4-7 (b) correspond to specific rotational positions of the box when only two faces were visible. This introduced greater uncertainty and error in length estimation. The final figure 4-7 (c) shows rotational velocities of the box. Since these correspond to derivatives of the noisy rotation position, and have been filtered for viewing clarity. Note that the total rotational velocity remains nearly constant as well as it's components.

This chapter has demonstrated the effectiveness and performance of the algorithm proposed in this thesis. The following chapter will summarize the thesis and discuss applications and future work.

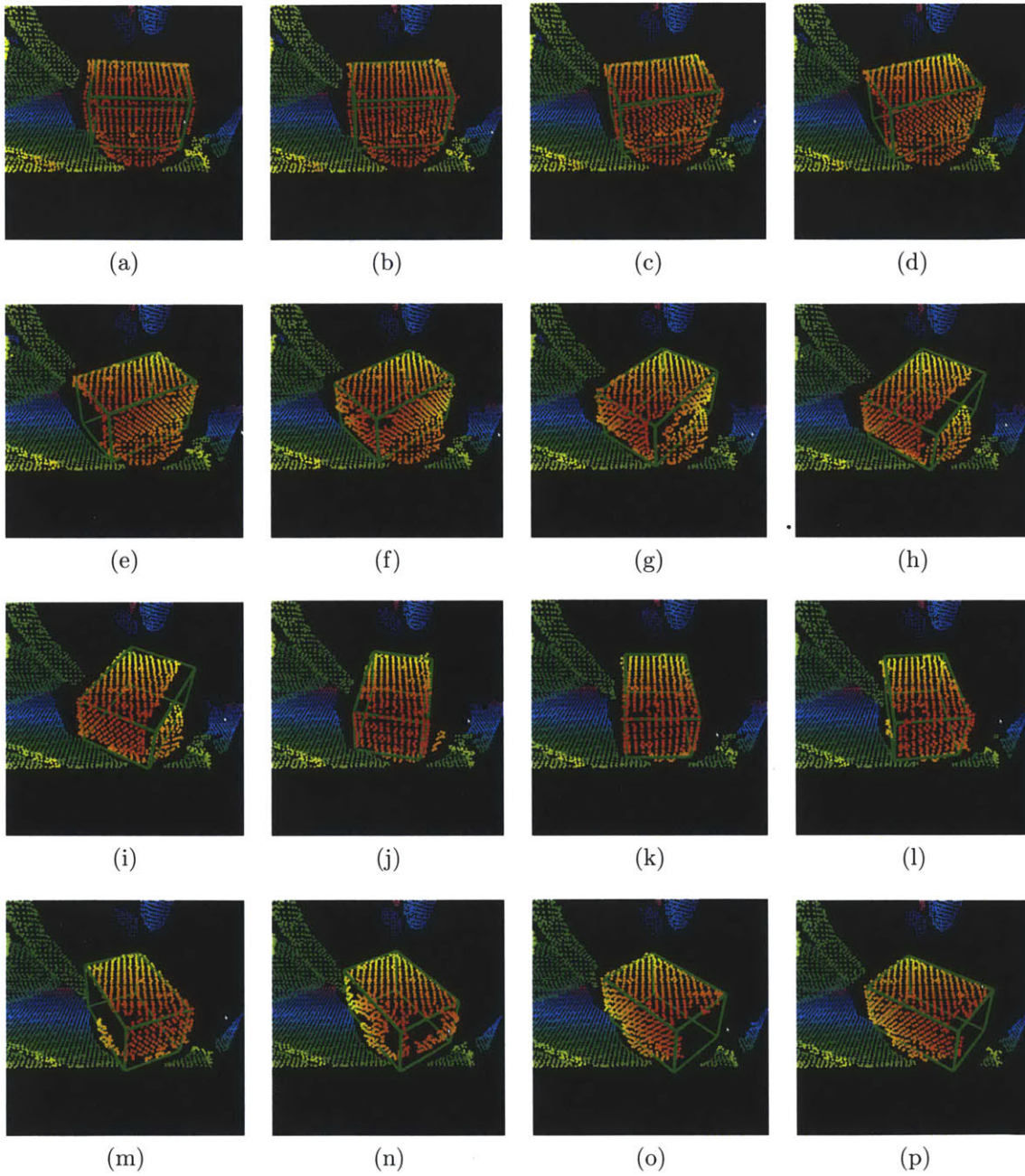
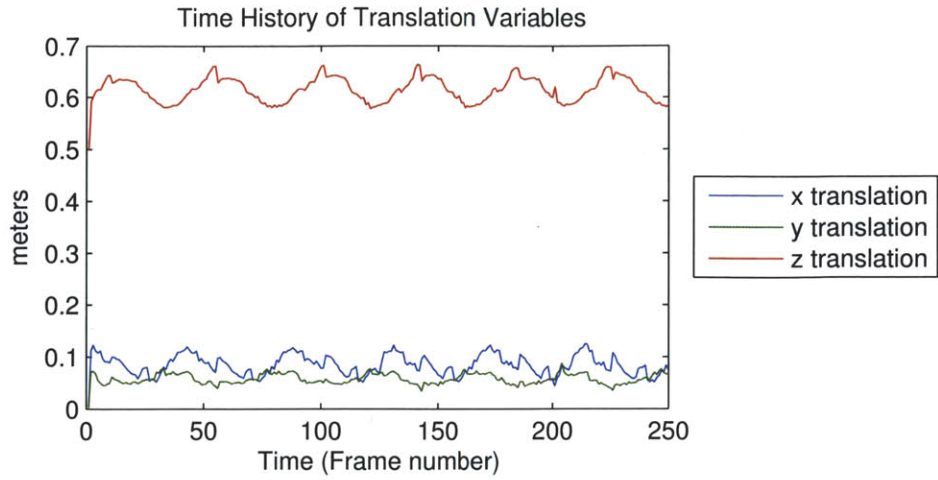
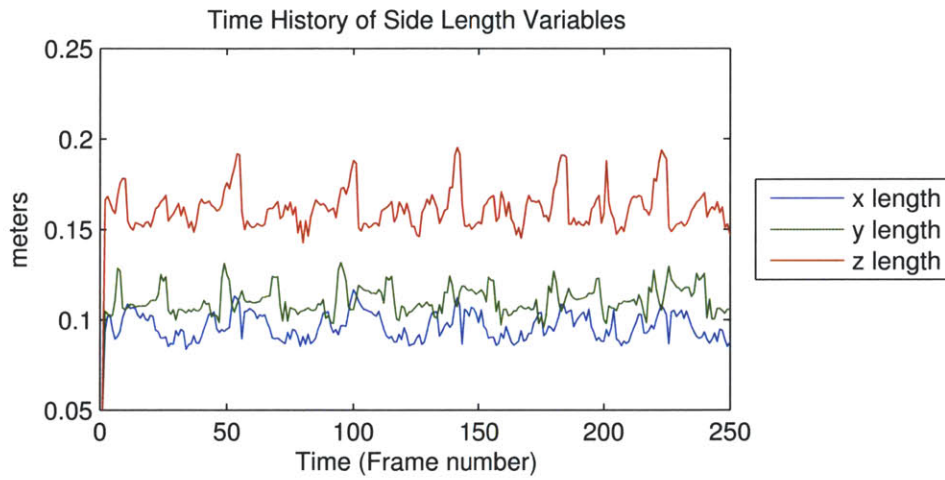


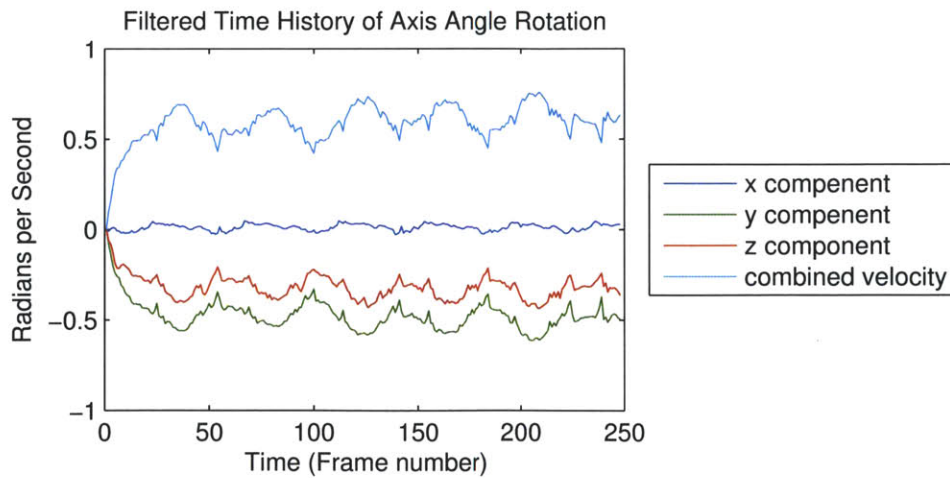
Figure 4-6: A sample set of frames from a rotation box. These were taken at approximately 4 Hz using a Microsoft Kinect. Notice that despite faces gaining and losing visibility, the relaxation algorithm successfully tracks the box pose, represented by the green cuboid.



(a)



(b)



(c)

Figure 4-7: A time history of box parameters variables for experiment shown in figure 4-6. (a) shows the translation components of the box centroid, (b) shows the side length components, and (c) shows a filtered rotation velocity.

Chapter 5

Conclusions

This thesis has demonstrated a method for localizing and tracking parameterized objects in a point cloud sensor input for both static and dynamic cases. This algorithm provides discrete state measurements for use by other methods to find more interesting dynamic characteristics, including center of mass, ratios of moments of inertia, and accelerations. While there are several adequate solutions available with the information in hand, first extracting this data from dense range and bearing sensors inputs has been a difficult problem.

The main idea behind using a parameterized object is to avoid using complicated and heuristic compositional methods for object localization. While these methods have distinct advantages of generality, they become cumbersome when used for particular objects on interest. By parameterizing types of objects with simple generation functions, many of the heuristic steps can be dropped. While this thesis mostly explored cuboids, any sufficiently parameterized object can be used. With a model in hand, the only other parameters to tune are the expected noise of the sensor measurements, and a threshold for how likely an object must be before it can be considered a true object instead of a spurious local minimum. Compositional methods require many more parameters. How perpendicular should planes be? How close? How square should a detected plane be? As objects increase in complexity, the number of parameters to tune increases greatly with compositional methods.

This method is significantly more robust to occlusion than other compositional

methods, as shown in a variety of static cases. Certain configurations of objects and occluding obstacles break existing algorithms since their heuristics have not accounted for the variety of cases that can arise and violate assumptions about the primitive shapes. By using this model based approach, the point cloud as a whole can be considered when analyzing a hypothesis. This captures more of the difficult cases than are solvable with compositional methods.

The relaxation algorithm presented has another major advantage over compositional methods for dynamic tracking. Compositional methods must recompute object primitives and rebuild hypotheses with each frame. This wastes a significant amount of processing power. The relaxation algorithm uses the previous state as a guess for the next state, and quickly accommodates new data to update objects of interest in real time. This allows for speed currently unachievable with compositional methods at the high bandwidth rates required of 30 Hz Kinect sensors.

This algorithm may be applied to a variety of autonomous robotic tasks. The robotic porter described in chapter 1 can use this algorithm to be more robust when locating boxes to manipulate. It will allow its environment to become more cluttered without forcing more computation, as required with a compositional method. The robot will also be able to handle a wide range of box sizes, since each has a common parameterization. The task could also expand to handle objects in motion since the proposed algorithm can handle dynamic objects while compositional methods will be too slow. This method would also apply to any case where a robot would wonder if an object of interest (a) was present and (b) where it was. For instance, a robot could find all chairs and their locations in a building, then move them to a predetermined pattern. Sensors placed on cars could locate and track all other car and human shaped objects for safety. Since autonomous robotics tend to focus on specific tasks and objects, this method could have wide applicability in the field.

Future work on this algorithm should first expand the complexity of objects parameterized. Boxes were chosen since they were too complicated to be found using RANSAC, but simple for easier implementation and testing. Objects like chairs, desks, or statues would more clearly demonstrate the superiority of parameterized

models over heuristic primitive assembly.

Bibliography

- [1] Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes: reconstruction and parameterization from range scans. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 587–594, New York, NY, USA, 2003. ACM.
- [2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. *ACM Trans. Graph.*, 24:408–416, July 2005.
- [3] P.J. Besl and R.C. Jain. Three-dimensional object recognition. *ACM Computing Surveys (CSUR)*, 17(1):145, 1985.
- [4] P.J. Besl and R.C. Jain. Invariant surface characteristics for 3D object recognition in range images* 1. *Computer vision, graphics, and image processing*, 33(1):33–80, 1986.
- [5] G. Biegelbauer and M. Vincze. Efficient 3D object detection by fitting superquadrics to range image data for robots object manipulation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2007.
- [6] Andrew Blake and Andrew Zisserman. Visual reconstruction. 1987.
- [7] Michael J. Brooks. *Shape from shading*. MIT Press, Cambridge, MA, USA, 1989.
- [8] Rodney A. Brooks. Symbolic reasoning among 3-d models and 2-d images. *Artificial Intelligence*, 17(1-3):285 – 348, 1981.
- [9] S. Chaudhuri and A. Rajagopalan. Depth from defocus: a real aperture imaging approach. 1999.
- [10] D.M Cole and P. M. Newman. Using laser range data for 3d SLAM in outdoor environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Orlando Florida USA, May 2006.
- [11] H. Delingette, M. Hebert, and K. Ikeuchi. Shape representation and image segmentation using deformable surfaces. *Image Vision Comput.*, 10:132–144, April 1992.
- [12] Nathaniel Fairfield, George Kantor, and David Wettergreen. Real-time slam with octree evidence grids for exploration in underwater tunnels. *Journal of Field Robotics*, 24(1-2):03–21, 2007.

- [13] Paolo Favaro and Stefano Soatto. *3-D Shape Estimation and Image Restoration: Exploiting Defocus and Motion-Blur*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [14] David A. Forsyth, Okan Arikan, Leslie Ikemoto, James F. O’Brien, and Deva Ramanan. Computational studies of human motion: Part 1, tracking and motion synthesis. *Foundations and Trends in Computer Graphics and Vision*, 1:77–254, 2006.
- [15] Jonas Garding. Shape from texture for smooth curved surfaces in perspective projection. *Journal of Mathematical Imaging and Vision*, 2:630–638, 1992.
- [16] F. Sebastin Grassia. Practical parameterization of rotations using the exponential map. *J. Graph. Tools*, 3:29–48, March 1998.
- [17] Berthold K. Horn. *Robot Vision*. McGraw-Hill Higher Education, 1st edition, 1986.
- [18] Berthold K. P. Horn. Shape from shading. chapter Obtaining shape from shading information, pages 123–171. MIT Press, Cambridge, MA, USA, 1989.
- [19] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathematics*, II(2):164–168, 1944.
- [20] Y. Liu, R. Emery, D. Chakrabarti, W. Burgard, and S. Thrun. Using EM to learn 3D models of indoor environments with mobile robots. In *Intl. Conf. on Machine Learning (ICML)*, 2001.
- [21] Anthony Lobay and David A. Forsyth. Shape from texture without boundaries. *International Journal of Computer Vision*, 67(1):71–91, 2006.
- [22] Jitendra Malik and Ruth Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *Int. J. Comput. Vision*, 23:149–168, June 1997.
- [23] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963.
- [24] D. Marr and H. K. Nishihara. Representation and Recognition of the Spatial Organization of Three-Dimensional Shapes. *Royal Society of London Proceedings Series B*, 200:269–294, February 1978.
- [25] Z. Marton, L. Goron, R. Rusu, and M. Beetz. Reconstruction and verification of 3d object models for grasping. In *Proc. of the 14th Intl. Symposium on Robotics Research*, 2009.
- [26] T. McInerney and D. Terzopoulos. A finite element model for 3d shape reconstruction and nonrigid motion tracking. In *Computer Vision, 1993. Proceedings., Fourth International Conference on*, pages 518–523, may 1993.

- [27] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Underst.*, 104:90–126, November 2006.
- [28] R. Rusu, Z. Marton, N. Blodow, M. Dolha, and M. Beetz. Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56:927–941, 2008.
- [29] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.
- [30] R. Schnabel, R. Wahl, and R. Klein. Efficient RANSAC for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226, 2007.
- [31] F. Solina and R. Bajcsy. Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:131–147, 1990.
- [32] S. Teller, M.R. Walter, M. Antone, A. Correa, R. Davis, L. Fletcher, E. Frazzoli, J. Glass, J.P. How, A.S. Huang, Jeong hwan Jeon, S. Karaman, B. Luders, N. Roy, and T. Sainath. A voice-commandable robotic forklift working alongside humans in minimally-prepared outdoor environments. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 526 –533, may 2010.
- [33] D. Terzopoulos and D. Metaxas. Dynamic 3d models with local and global deformations: deformable superquadrics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 13(7):703 –714, jul 1991.
- [34] P. Torr and A. Zisserman. MLESAC: a new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156, 2000.
- [35] R. Truax, R. Platt, and J. Leonard. Using prioritized relaxations to locate objects in points clouds for manipulation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2011.
- [36] G. Vosselman and S. Dijkman. 3D building model reconstruction from point clouds and ground plans. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(3/W4):37–43, 2001.
- [37] G. Vosselman, B. Gorte, and G. Sithole. Recognising structure in laser scanner point clouds. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 46(8/W2):33–38, 2004.
- [38] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic

systems. In *Proc. of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, Anchorage, AK, USA, May 2010. Software available at <http://octomap.sf.net/>.

- [39] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape from shading: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21:690–706, August 1999.