

ROBOTIC GRASPING OF UNKNOWN THREE DIMENSIONAL OBJECTS

A NEW APPROACH FOR ROBOTIC GRASPING OF UNKNOWN
THREE DIMENSIONAL OBJECTS

BY

MICHAEL JAMES KRISTENSEN IRVINE, B.Eng.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER

ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

© Copyright by Michael Irvine, September 2012

All Rights Reserved

Master of Applied Science (2012)

McMaster University

(Electrical & Computer Engineering)

Hamilton, Ontario, Canada

TITLE: A New Approach for Robotic Grasping of Unknown
Three Dimensional Objects

AUTHOR: Michael Irvine

B.Eng., (Electrical Engineering)

McMaster University, Hamilton, Canada

SUPERVISOR: Dr. David Capson, Dr. Gary Bone

NUMBER OF PAGES: iv, 231

Abstract

Automated grasping of objects of unknown geometry *a priori* has applications in many industries such as clearing a mine shaft after blasting, agricultural applications such as fruit and vegetable handling, and many roles in the service industry such as fetching items for a handicapped individual. In these roles the system environment is highly unstructured, and the robot must be able to react to different types of objects needing to be grasped. In this thesis a vision guided robotic grasp planner for unstructured environments is presented. An evaluation method for robotic grasping involving two distinct sets of objects is also presented. Both the grasp planner and evaluation metric are evaluated by experimentation using an articulated robotic arm with an eye-in-hand video camera, line laser, and pneumatic gripper. Multiple grasping experiments were performed with the objects in random poses on a modified tabletop deemed the playfield that did not allow objects to rest flat.

The grasp planner focused on using a created model of the object from camera observations using silhouetting and line laser data. The object model and its computed convex hull were used to evaluate and select a single facet and point creating a grasping pair for the pneumatic gripper jaws. The grasp was attempted and then evaluated using a secondary camera and the developed evaluation method.

Grasp success rates ranged from 80.30% (Rectangular Block on playfield 137 attempts) to 97.69% (Hexagonal Nut 173 attempts), with a mean grasp computation time for the hexagonal nut of 0.57s.

Acknowledgements

I would like to thank my co-supervisors Dr. Gary M Bone, and Dr. David W Capson for their guidance. I would like to thank my family and friends for their support. I would like to thank the second floor technicians for their knowledge and assistance with machining, the Formula SAE, Formula Hybrid, and McMaster Autonomous Robot Racing Team for assisting in the development of the technical skills that I would need. Finally, I would like to thank my labmates for bouncing ideas off of, helping with experiments, and generally keeping me sane.

Contents

Abstract	iii
Acknowledgements	v
Contents	vi
List of Figures	xii
List of Tables	xxii
List of Abbreviations	xxiii
Chapter 1 Introduction and background	1
1.1 Motivation	1
Chapter 2 Literature Review	6
2.1 Introduction	6
2.2 Previous work	6
2.2.1 Model-based Grasp Planning	6
2.2.2 Compliant grippers	19
2.3 Grasping Evaluation	19
2.4 Summary	20
Chapter 3 Software and Hardware Overview	22
3.1 System Hardware	22

3.2	Execution Overview.....	25
Chapter 4	Camera Calibration	27
4.1	Objective	27
4.2	Relationships between the Coordinate Frames.....	27
4.3	Details of the Camera Calibration	33
4.3.1	Key Locations and Notation	33
4.3.2	Calibration Method	34
4.3.3	Procedure for Coplanar Calibration	40
4.4	Calibration Strategies for Object Modeling, Laser Range Measurement, Initial Object Finding and Pose Measurement.....	46
Chapter 5	Model Creation	50
5.1	Introduction.....	50
5.2	Object Model Creation from Image Silhouettes	50
5.2.1	Overview	50
5.2.2	Image Capture.....	51
5.2.3	Object Boundary Detection and Silhouette Computation.....	55
5.2.4	Model Creation from the Set of Image Silhouettes.....	56
5.3	Object Model Refinement Using Structured Light System	60
5.3.1	Introduction	60
5.3.2	Procedure for Generating the Point Cloud Model.....	60
5.3.3	Convex Assumption.....	62
5.3.4	Combination of Silhouette and Structured Light Models.....	64

5.4	Parameters Used in Our Implementation	65
5.5	Conclusions.....	67
Chapter 6	Grasping Strategy.....	68
6.1	Overview of Grasping Strategy.....	68
6.2	Convex Hull.....	69
6.2.1	Definition of Convex Hull	70
6.2.2	Computation of Convex Hull	72
6.2.3	Triangular and arbitrary polygonal facets.....	75
6.2.4	The qconvex function.....	76
6.3	Grasp Planning Theory	76
6.3.1	Closure	77
6.3.2	Rotation.....	86
6.3.3	Facet Area and <i>COM</i>	88
6.4	Finding Potential Grasping Surfaces.....	89
6.4.1	Grasp Planner inputs and outputs	92
6.4.2	Pre grasp-planner steps	94
6.4.3	Grasp planner.....	99
6.4.4	Reduction of Grasp Set	111
6.4.5	Handling Exceptions.....	120
6.4.6	Final Grasp Selection.....	121
6.5	Grasp Execution.....	123
6.5.1	Calculation of Approach Vector and Gripper Orientation Angles	123
6.5.2	Calculation of approach point.....	131
Chapter 7	Evaluation Technique.....	133

7.1	Grasp Evaluation Purpose	133
7.2	Object Grasp Evaluation Procedure	134
7.3	Object Categorization.....	136
7.3.1	Grasp Repeatability Object	137
7.3.2	Offset Calculations	139
7.3.3	Definition of Graspability Objects.....	140
7.4	Object Pose Identification	141
7.5	POSIT Computation	141
7.6	Target detection and Identification	142
7.6.1	Target Segmentation	142
7.6.2	Barcode Detection and Code Extraction.....	146
7.7	Calculation of Pose from Data.....	154
Chapter 8	Experimental Procedure	157
8.1	Test Objects.....	157
8.2	Grasp Repeatability Object	157
8.3	Graspability Objects	158
8.3.1	Cube	158
8.3.2	L-Shaped Block	158
8.3.3	T-Shaped Block.....	159
8.3.4	Hexagonal Nut.....	160
8.4	Baseline System Errors	160

8.4.1	Imager Pixel And Stationary Robot Errors	160
8.4.2	Robot Repeatability Errors.....	161
8.5	Grasping.....	161
8.5.1	Flat Tabletop Workspace	161
8.5.2	Playfield.....	162
8.5.3	Testing procedure for grasps	162
Chapter 9	Experimental Results	164
9.1	Stationary Robot Test Results	164
9.1.1	Stationary Robot Test Results Data	165
9.1.2	Stationary Robot Test Results Discussion	167
9.2	Robot Repeatability Test Results	169
9.2.1	Robot Repeatability Test Results Data.....	169
9.2.2	Robot Repeatability Test Results Discussion	170
9.3	Grasping Results for Grasp Repeatability Testing	170
9.3.1	Robot Grasping Rectangular Box Initially on Flat Tabletop	171
9.3.2	Robot Grasping Rectangular Block Initially on Playfield	174
9.4	Grasping Results for Graspability Testing	177
9.4.1	Robot Graspability Testing Data	177
9.4.2	Robot Graspability Testing Discussion	178
9.5	Timing information.....	179
9.5.1	Timing data for the Hexagonal Nut.....	179
9.5.2	Timing data discussion for the Hexagonal Nut	183
9.6	Discussion	184
Chapter 10	Conclusions and Recommendations.....	185

10.1	Summary and Key Achievements.....	185
10.2	Future Work	186
	References	190
Appendix A	F3 Robot Arm Workspace [45]	204
Appendix B	Example visual targets for rectangular box object.....	205
Appendix C	Hue Saturation Value (HSV) colour-space.....	206

List of Figures

Figure 3-1 Eye-in-hand video camera, line laser, custom ring light and pneumatic gripper mounted on the 6DOF robotic arm.	24
Figure 3-2 Exposed playfield showing obstacles.	25
Figure 3-3 Overview of the system execution.	26
Figure 4-1 Coordinate Systems for the: w world, p pose, rb robot base, r robot, c camera, g gripper, and l laser frames.	27
Figure 4-2 Robotic Setup Side view	28
Figure 4-3 Gripper transformation parameters shown in the $Xr-Yr$ plane.....	30
Figure 4-4 Gripper side view shown in the $Xg-Zg$ plane.	31
Figure 4-5 Input image transformation between the robot, grasp, and camera frames..	32
Figure 4-6 The robot to laser transformation parameters for the matrix Hrl	33
Figure 4-7 Tsai Calibration system Image from [35]	34
Figure 4-8 Calibration pattern (not to scale).	40
Figure 4-9 Radial alignment constraint from Tsai [35]	41
Figure 4-10 First camera calibration overhead used to align the coordinate systems.	49
Figure 4-11 Laser scanner calibration setup. The calibration pattern is placed in the same plane as the laser light.	49
Figure 5-1 Demonstration of silhouette object modeling error.	51
Figure 5-2 Geometry of the image capture locations.....	53

Figure 5-3 Input images transformation	54
Figure 5-4 An example set of images taken of a T-shaped block.	54
Figure 5-5 An example of an object being filtered. The top left is the original image of the object, top right is the Sobel filtered version of the same image, bottom left is the thresholded version of the filtered image, and the bottom right is the boundary of the object.	56
Figure 5-6 A single silhouette image showing the focal plane of the camera. If this was the only image taken then the model of the object would be the shaded cone.	57
Figure 5-7 Beam based modeling shorting by projecting the model on to the image plane. Image modified from [16].	58
Figure 5-8 Improvement of the model resulting from successively including the image silhouettes from the 8 viewpoints. For clarity the convex hulls of the point clouds are shown. a) overhead, $\theta_{im} = \pi/2, \theta = 0, hc = 309mm$, b) Image 2 $\theta_{im} = \pi/6, \theta = -1.92, hc = 267.6$ c) Image 2 $\theta_{im} = \pi/6, \theta = -1.40, hc = 267.6$, d) Image 3 $\theta_{im} = \pi/6, \theta = -0.87, hc = 267.6$, e) Image 4 $\theta_{im} = \pi/6, \theta = -0.35, hc = 267.6$,f) Image 5 $\theta_{im} = \pi/6, \theta = 0.17, hc = 267.6$,g) Image 6 $\theta_{im} = \pi/6, \theta = 0.70, hc = 267.6$, h) Image 7 $\theta_{im} = \pi/6, \theta = 1.22, hc = 267.6$,i) Image 8 $\theta_{im} = \pi/6, \theta = 1.75, hc = 267.6$	59
Figure 5-9 Laser projection onto object showing wrist rotation.	61

Figure 5-10 This figure demonstrates the laser convex assumption. a) Object and laser line. b) Laser line over center of object. c) Discontinuous laser line. d) Laser line with convex assumption.	64
Figure 6-1 Narrow and wide gripper jaws acting on an object. The narrow jaws produce unstable contacts.	69
Figure 6-2 Left: a two dimensional shape. Right: its convex hull.	70
Figure 6-3 Left: a random set of points. Right: The red line represents the convex hull of the same set of points.....	71
Figure 6-4 Left: A point cloud of an object containing 2704 points. Right: The convex hull of the same set of points with 68 vertices and 63 facets.	72
Figure 6-5 Gift wrapping of a convex hull in two dimensions.	74
Figure 6-6 Gift wrapping of a convex hull in three dimensions first edge.	74
Figure 6-7 Gift wrapping of a convex hull in three dimensions second edge.	74
Figure 6-8 Left: a triangular facet. Right: an arbitrary polygonal facet.	75
Figure 6-9 Friction cone for a hard finger contact with a normal force f_n , an angle α , and a coefficient of friction of μ	78
Figure 6-10 Friction cone for a soft finger contact with a force f_n , an angle α , a coefficient of friction of μ . A torque τ_{sf} around the contact point can be exerted.....	78
Figure 6-11 Frictionless contact for a force f_n , this is the same as a hard finger contact with a coefficient of friction $\mu = 0$ and therefore the friction cone has an angle α of 0.	79

Figure 6-12 The forces acting on an object during a parallel jaw grasp, including gravity, shown from the front.....	80
Figure 6-13 The forces and torques acting on an object during a parallel jaw grasp, including gravity, shown from the side.....	80
Figure 6-14 Front view of the grasping of an object with a minimum number of contact points.	82
Figure 6-15 Left: a front view of the object being grasped with the contact points and friction cones. Right: a side view of the object being grasped with the contact points, selected facet and friction cones.	82
Figure 6-16 Demonstration of the torque around a point and the forces that act because of it.	84
Figure 6-17 The view from the parallel jaw plane, with the finger contact points resisting a rotation along the grasping axis.....	86
Figure 6-18 Left: the opposite point lies interior to the projected facet and the grasp is stable. Right: the opposite point lies exterior to the projected facet and the object rotates in the gripper.	87
Figure 6-19 The force from the grasping axis and from the force of gravity.	89
These are illustrated in Figures 6-20, 6-21 and 6-22.	90

Figure 6-23 The n point facet, facet point and opposite point. The facet point is the intersection of a line projected from the opposite point along the facet normal and the plane the encompasses the facet	90
Figure 6-24 Left: an image of the n point facet from the side such that the facet extends into the page. The facet normal is at a right angle to the facet. It can be seen here where $pfacet$ lies on the intersection of that facet normal projected from the opposite point. Right: The same facet rotated along the by $\pi/2$ about yw and tilted such that the view is along the normal of the selected facet. Here $pfacet$ and $popposite$ are in line with each other.....	91
Figure 6-25 This is the same as Figure 6-24 with the exception that here $pfacet$ appears outside the facet while projected back along the normal.....	92
Figure 6-26 an example of a 6 sided polygon ($n = 6$) divided into 4 separate triangles $6 - 2 = 4$	95
Figure 6-27 calculation of area A_i and centroid $ctri$ of a triangle tri	96
Figure 6-28 $pfacet1$ lies inside the facet, while $pfacet2$ lies outside the facet.	103
Figure 6-29 Calculating whether the point lies interior or exterior.	104
Figure 6-30 Interior/exterior point example.	106
Figure 6-31 Distance between the line joining $pfacet$ and $popposite$ and the $pCOM$	107
Figure 6-32 By projecting the facet onto the z plane the largest and smallest values for the x and y values are found.....	110

Figure 6-33 Procedure to find the tip center	111
Figure 6-34 Grasping size constraints. <i>popposite</i> is <i>dista</i> away from the selected facet, the point <i>pdistmax</i> is <i>distb</i> away from the selected facet. If <i>distb - dista</i> is greater than the threshold then the grasp is removed.....	113
Figure 6-35 Left <i>distb - dista</i> is small resulting in almost no rotation, Right <i>distb - dista</i> is large resulting in a large rotation.....	114
Figure 6-36 Visualization of the 20° no grasping zone	115
Figure 6-37 Image facet error due to height. The red area is the error the model will have due to the height of the camera over the object, the height of the object above the table, and the angle of the camera.....	117
Figure 6-38 Exaggerated visualization of the grasps that are removed due to the vision modeling error.	117
Figure 6-39 Exaggerated visualization of the all of the areas that have to be removed from modeling error and gripper maximum angle constraint. (not to scale).	118
Figure 6-40 Geometry of the gripper relative to the table.....	119
Figure 6-41 Lower boundary removal, removing the offending point from the convex hull, and hence the problem from the facet.	121
Figure 6-42 Procedure for tumbling the object when no grasp is found	122
The gripper will approach the object by moving along the “approach vector”. The approach vector, <i>a</i> , is calculated using the normal of the selected grasp, <i>n</i> , and the	

normal to the table, v . The method presented here is similar to the one used in the unpublished code developed by Bone <i>et al.</i> [16]. The three vectors are shown in Figure 6-44 and 6-43.	123
Figure 6-44 The grasp normal vector n , table normal vector v , and approach vector a ..	124
Figure 6-45 Vector a is the negative of the component of v orthogonal to n	124
Figure 6-46 The problem with a non-zero θ_x angle. As the gripper moves downwards the bottom right corner of its jaw may hit the table before the center of the gripper is positioned to grasp the object.....	126
Figure 6-47 Definition of the θ_y angle.....	127
Figure 6-48 Projection of a onto the $ywzw$ plane.	127
Figure 6-49 Calculation of the z rotation θ_z	129
Figure 6-50 Robotic arm physical constraints that must be considered to avoid collision with the laser bracket	130
Figure 6-51 Robotic arm physical constraints that are enforced whenever the arms yr coordinate is above $50mm$	131
Figure 7-1 The robot in the pose position with and object from set one in the gripper.	135
Figure 7-2 An image of a rectangular box taken by the pose evaluation camera.....	136
Figure 7-3 The grasp repeatability object, shown with circular targets attached.....	138
Figure 7-4 The grasp repeatability object with the orientations of the perfect pose p frame and object o frame shown.	138

Figure 7-5 Graspability objects.	141
Figure 7-6 This image shows the object and the variations in the apparent colour of the red targets. Note that all of the targets are the same true colour.....	144
Figure 7-7 Image of the object after thresholding in the HSV colour-space.	145
Figure 7-8 The image of the object after the morphological open close filter.	146
Figure 7-9 The boundary following algorithm showing four iterations.	148
Figure 7-10 An example where the direction ending criteria is needed to ensure that the boundary is not prematurely ended.....	149
Figure 7-11 Minimum size constraint for X and Y directions.....	150
Figure 7-12 The image of the object target identification, red circles indicate target locations.....	151
Figure 7-13 A test set of circular red targets.	152
Figure 7-14 Barcode identification numbers.	152
Figure 7-15 A circular target tilted to approximately 70°.	154
Figure 7-16 The stored model of a box, and its x, y, z components and barcode ID's. ..	154
Figure 7-17 The generated homogeneous transformation matrix file for the box object.	155
Figure 7-18 The image of the object with the targets, and its computed pose from POSEIT.	156
Figure 8-1 The cube graspability object.....	158

Figure 8-2 The L-shaped block graspability object.	159
Figure 8-3 The T-shaped block graspability object.	159
Figure 8-4 The hexagonal nut graspability object.....	160
Figure 9-1 Histograms of the object centroids from the stationary robot test.....	165
Figure 9-2 Histograms of the object pose frames from the stationary robot test.	166
Figure 9-3 Histograms of the robot repeatability from the robot repeatability test.	169
Figure 9-4 Histograms of the pose errors from the grasp repeatability test with the rectangular box object initially on the flat tabletop.	172
Figure 9-5 Histogram of the pose errors from the grasp repeatability test with the rectangular box on the playfield.	174
Figure 9-6 Top Left: Standard Grasp, Top Right: Small amount of offset along Y_o , Bottom Left, and Bottom Right: Large amount of offset along Y_o	177
Figure 9-7 Histograms of the time the algorithm presented in [16] (top) and the time the algorithm presented in this thesis (bottom) required to complete the grasp planning for the hexagonal nut object.	181
Figure 9-8 Histograms of the times the algorithm required to complete each step for the hexagonal nut object.	182
Figure 10-1 Top Left: Gripper element with center(shown in green); Top Right: Gripper elements around a corner; Bottom Left: 2D element to grasp; Bottom Right: Same 2D	

element with convex hull modifications (Red) and new modified convex hull modifications (green).....	187
Figure C-1 The representation of the HSV colour-space [82].....	207
Figure C-2 The representation of the RGB colour-space, shown as a cube [83].	207

List of Tables

Table 5-1 Parameters used in the grasping program	65
Table 7-2 HSV threshold values for red targets, front set.	143
Table 7-3 HSV threshold values for red targets, side set.....	143
Table 7-4 The element used for the morphological open, and morphological close.....	145
Table 7-5 Boundary following pixel location map.	147
Table 9-1 Means and standard deviations of target centroids	165
Table 9-2 Means and standard deviations of the pose frame.....	167
Table 9-3 Means and standard deviations of the object pose from the robot repeatability test.	170
Table 9-4 Means and standard deviations of the pose errors from the grasp repeatability test with the rectangular box object initially on the flat tabletop.	172
Table 9-5 Means and standard deviations of the pose errors from the grasp repeatability test with the rectangular box object initially on the playfield.	175
Table 9-6 Object graspability results	177
Table 9-7 Means and Standard Deviations for the two histograms presented above....	181
Table 9-8 Means and standard deviations of the times it took to complete each of the actions during the grasping routine.....	183

List of Abbreviations

POSIT	Pose from Orthography and Scaling with Iterations
OpenCV	Open Computer Vision
COM	Center of Mass
CGF	Candidate Grasp Facet
DOF	Degrees of Freedom
HSV	Hue Saturation Value
RGB	Red Green Blue

Chapter 1

Introduction and background

1.1 Motivation

Automated grasping of objects of unknown geometry *a priori* is still a difficult task for robots. Robotic grasping tasks are usually highly structured, using previously known objects and locations with a stored grasping procedure. In general most robots move known objects from one known location to another using predefined grasping points and are unable to cope with misaligned and/or shifted objects. This is especially true while attempting to grasp objects when the object is not resting flat on a surface. Because of these limitations, robotic grasping is usually limited to industrial settings where specific robotic motions are taught by operators and fixtures are able to hold objects in known locations with fixed orientations. The main disadvantage in this setting is the fact that exact models of the object are required. If any part of the object to be grasped changes, including but not limited to mechanical design revisions or manufacturing errors, the grasping system needs to be recalibrated. Another challenge is that mechanical fixtures are object specific and are usually expensive to produce. Robotic grasping is also very difficult in unstructured environments such as a home, where the objects shape data and location are not explicitly known, or with any processes that deal with random shaped objects such as raw material handling. The combined position and orientation of an object is termed its *pose*. When the object

rests on a flat surface, the deviations from its expected pose are limited to three degrees-of-freedom (DOF). For the more general case of an uneven surface, or an object that can rest on one of several of its sides, the deviations of the pose can include all six DOF. A solution to the above problems is a grasping system that is able to adapt to each object's pose and shape. A vision-guided robotic grasping system could generate a new distinct model and calculate a grasp for each object. The above solution would be flexible enough to adapt to changing industrial demands, as well as satisfy the grasping needs in the unstructured service industry. It would also be useful in areas where humans could not accompany the robotic device, e.g. radioactive or bio-contaminated locations; underwater or internal pipe repair; or space exploration where one way communication can take many minutes [1].

Current, commercially available, vision-guided robots for object grasping employ 2D and 3D computer vision. For example, the ABB IRB 360 flexpicker robot [2], introduced in 2001, is available with 2D computer vision for rapidly finding and picking up objects from a conveyor. The Baxter Robot introduced in 2012, uses 2D shape recognition to identify and grasp previously taught objects [3, 4]. With 2D vision the system can only measure and adapt to planar position and orientation changes (i.e. ΔX , ΔY and $\Delta \theta$). Typically it is not possible to adapt these systems to grasp previously unknown objects. While 2D vision-guidance is an important advancement only 3D vision-guidance will allow robots to succeed in unstructured and quickly changing environments in manufacturing and

service applications. Cognex, the company who provided ABB with the flexpicker vision system, has developed a 3D vision system called "VisionPro 3D" [5]. VisionPro 3D uses one or more cameras and finds multiple sets of predefined two-dimensional features, these features are then matched to find the precise pose of the object. Predefined grasps can then be used on the object. This system cannot work with unknown objects.

This research focuses on developing a vision-guided grasping system that can grasp previously unknown 3D objects. 3D model generation will be done online using computer vision and a standard computer. The system will then automatically generate and execute a grasp based on the acquired model. The generation of the model and grasp will be based on data obtained from an eye in hand camera and a line laser. This model will be used as an input to the grasp planner and a grasp generated for a simple parallel jaw gripper. After each grasp is executed it will be evaluated using a separate offline vision system.

Since the focus of this thesis is on key areas of 3D modeling using computer vision, laser scanning and grasp planning, some assumptions about the vision-guided grasping system are made. For maximum flexibility of the grasping system, the number of assumptions was limited as much as possible. The assumptions made are:

1. There is good contrast between the grasped object and the background. This assumption allows our vision research to focus on the area of 3D vision-based modeling rather than image segmentation.
2. The grasped object is located within a predefined robot-vision workspace, and the robot's dexterous workspace. The robot and camera workspaces are limited; the object must be located where the video camera can see the whole object for every viewpoint. It must also be located where the robot can reach and grasp it. Within the common workspace, the object may have any pose.
3. The gripper jaws used in the experiment are assumed to be much larger than the grasping surfaces. In other words, the gripper jaws do not place a restriction on the grasp planning task.
4. There exists enough friction between the gripper jaws and the object for grasping to be successful.
5. The object is rigid.

This thesis is broken down into 10 Chapters. Chapter 2 presents a review of the current literature on automated robotic grasping. The review focuses on grasping strategies where the model of the object is unknown a priori. Chapter 3 is an overview of the hardware and software used for the grasping system. Chapter 4 describes the camera calibration procedure, and includes the formulation of the homogeneous transformations between the different coordinate frames of the system. Chapter 5 deals

with image capture, segmentation, model creation, and laser-based model refinement.

Chapter 6 explains the model-based grasping strategy. Chapter 7 describes the method and procedure for evaluating the grasps. A novel methodology for evaluating the repeatability of the grasps is presented in 7.2. Chapter 8 documents the procedure used for the grasping experiments. Chapter 9 presents the experimental results and their analysis. Finally, Chapter 10 presents the conclusions and suggestions for future work.

Chapter 2

Literature Review

2.1 Introduction

In this chapter the relevant research work done by other researchers will be critically reviewed. Specifically, it will cover their methods for creating models of objects as well as their grasping strategies, and results obtained. Their grasp evaluation techniques will also be reviewed.

2.2 Previous work

2.2.1 *Model-based Grasp Planning*

Stansfield [6] presented a system for grasping 3D objects using a six DOF PUMA robotic arm and a Salisbury robotic hand. Each object was first placed on a motorized table. The object was rotated and translated under a laser scanner to generate a set of 3D points. These were then combined to form a 3D model of the object in the robot frame of reference. The model was then decomposed into a maximum of five (top, left, front, back) aspects by the operator. The 3D model and aspects formed the input to a rule based expert system that planned the grasp. To form the grasp, three hand pre-shapes were used: pinch, wrap and grip. An example of the operation of the expert system was given for a cup. For this object, the expert system stated that a pinch grasp should be used since there was a rim contour in the "top" aspect. Experimental results in the form

of images were presented for several custom objects as well as a gear. No numerical results were presented.

Taylor, Blake and Cox described a system intended for picking an object from the top of a pile [7]. The hardware consisted of an ADEPT-1 robot with a parallel-jaw gripper and wrist mounted camera. The camera was moved about the object, measuring the curvature of the object boundary and looking for increasingly better grasps based on its image silhouette. The directions in which the camera and gripper were moved, were based on the measured curvature of the object edges. An object that has small radii of curvature moves the image viewpoint and gripper upward, an object with large radii moves it downward. After the “best” view was achieved the grasp was attempted. The only experimental grasping results included were for a potato located on a table.

Bendiksen and Hager [8] used a parallel jaw gripper and a grey scale camera to find the edge data of objects lying on a planar surface. A search for object-gripper contact was done in a polar coordinate system centred at the approximate centre of mass (calculated using the captured image). Each candidate grasp was analyzed for static equilibrium against the object’s weight. The equilibrium grasp with the minimum grasping force was selected as the final grasp. In their experiments, evaluations of the grasps were only given as pass/fail. Ten of 14 objects were successfully grasped.

For the grasp planning method in [9], “Oct-Trees” were used to model the object. Oct-Trees represent the object as a binary 3D matrix of cubes (of dimension $2^n \times 2^n \times 2^n$, where n is the number of cubes). Each cube can be broken down into eight sub-cubes. Each cube is in one of three states, full (F) where its sub-cubes are also full, void (V) where its sub-cubes are also void, and mixed (M) where its sub-cubes are a mix of F and V. The representation breaks down the M cubes until all cubes are either F or V. An example given of a M cube is the “staircase” object.

In [10] a model of the object to be grasped was created using a stereo vision setup. Several stereo images were taken from viewpoints around the object and combined into a depth range map. Heuristic rules converted the range map into a 3D occupancy probability map which was then segmented into separate objects. The object to be grasped was converted to an Oct-Tree representation, while the other objects were simplified using convex hulls or simple bounding boxes. Their vision system did not produce sufficiently reliable models, so they used a CAD model of the object for grasp planning. To generate grasps, the model was cut into thin slices, and the occupied area was encapsulated by ellipses. The size of the ellipses was reduced further segmenting the object until the parallel-jaw gripper was able to fit around them. Each ellipse was then evaluated along its major and minor axes to see if gripping it would cause the jaws to collide with any other ellipses. If no collision was predicted the grip became a grasp

candidate. After all the slices were evaluated, they were used to construct a 3D model. The algorithm then searched for similar 3D grasps (similar in terms of ellipse size, orientation, and gripper access direction) from different slice levels of the model. The similar grasps formed the set of 3D grasp candidates. The grasping algorithm then chose a grasp pre-shape for the object based on the need for power or dexterity as well as the size and shape of the grasp. No results were presented; instead a series of images of the gripper mounted to a table demonstrating the grasp pre-shapes were given.

The system presented by Trobina and Leonardis [11] used a Mitsubishi MV R1 robot with a parallel jaw gripper. Two table-mounted stripe projectors and range sensors were used to model the objects. The models, consisting of planar patches, were used to plan grasps for a parallel-jaw gripper. The tallest object was picked up and removed first, a new 3D model was then constructed, and the process was repeated until all objects had been removed. One experimental result employing a milk carton, coffee cup, correction fluid bottle and tape dispenser was included.

Borst, Fischer and Hirzinger [12] used a four finger DLR hand with 12 DOF. A pre-determined CAD model of the test object was used to calculate the grasp. The first contact point **1** is arbitrarily chosen. Next, rays are projected through the friction cone from **1**. These rays intersect with the object surface at location **3**. From the midpoint of the line connecting **1** and **3**, labeled **4**, two more cones **5a** and **5b** are created such that: all contact points lie in a plane; and cones **2**, **5a**, and **5b** are 120 degrees apart. The

location where the rays beginning from location **4** reach the surface through cones **5a** and **5b** are the contact points labeled as **6a** and **6b**. Their algorithm works with convex and non-convex objects. To evaluate the grasp quality, they presented a static grasp stability measure defined as the ability of the grasp to resist external forces and moments for a banana. They graphed quality vs. the number of grasp candidates. As the grasp candidates increased the quality of the worst grasps increased, until it plateaued at around 300 candidates.

Using a Motoman SV3X robotic arm equipped with a parallel-jaw gripper and a JAI M70 camera, Sanz *et al.* [13] used the assumption that the object's size in the Z direction (parallel to the camera's optical axis) was much smaller than that in the X-Y plane. This reduced the 3D grasping problem to 2D. From an overhead image, the visual centroid and minimum inertial axis of the object to be grasped were found. Grasp selection was based on: the radius of the contact patch that existed where the gripper jaws and the object would meet; the angle between the grasping line, from the centre of the first jaw to the centre of the second, the minimum inertial axis of the object; and the closest distance from the grasping line to the centroid. Objects grasped included pliers, a small toy alligator, random plastic shapes, and lettuce. No results were shown, but the system was said to be in use at a lettuce factory.

An approach that used simplified shape models based on primitives was proposed in [14]. Models of the objects were decomposed into simple shapes by the user. These

shapes included spheres, cylinders, cones and boxes. The simplified shapes were then passed to the grasping algorithm where predefined heuristics were used to generate grasps. The generated grasps were evaluated with the then unreleased Graspl! simulator shown in [15]. The simulation results included general statistics such as the total number of grasps evaluated, the number of force closure¹ grasps found, and the time spent testing the grasps for four isolated objects as well as two objects together in a complex scene. They did not perform any physical experiments.

Using a six DOF robotic arm equipped with a parallel-jaw gripper, camera and line laser, Bone *et al.* [16] modeled the object using a pillar model based on image silhouettes. The model was then further refined using a laser scanner. Grasping points were based on finding nearly flat, nearly parallel surfaces with a common projected area above a certain threshold. They stated that a larger common projected area makes the grasp less sensitive to object pose errors and external forces or moments. They included experimental results for objects that did not rest flat on a table. This tilting of the objects increased the difficulty of modeling and grasping them. Only two objects were shown being grasped: a metal bracket and a hex nut. No grasp quality results were presented.

Richtsfeld *et al.* [17] used a laser range scanner with an AMTEC seven DOF arm and Otto Bock prosthetic hand to grasp objects from a table top. The scanner produced a point

¹ Please see section 6.3.1 for the definition of force closure.

cloud. After the points belonging to table top were removed, the remaining points were segmented into different objects by clustering based on Euclidian distance, followed by a cylindrical fitting algorithm. If the object was found to be roughly cylindrical and smaller than the max opening distance of the hand then the object was grasped using its outside surface. If the cylindrical object was too large for the hand and there was another object outside of it that was larger than a set threshold, the secondary object was assumed to be a handle and became the grasping target. If the object was found not to be roughly cylindrical the convex hull of the top of the object was examined for vertices whose interior angle was smaller than the average. These were termed “corner points”. Next, the edge with the longest distance between corner points was used as the first side of the grasp while the other side was taken as any edge within a 5° tolerance of the first side, or failing that any corner point opposite to the first side. They stated that the grasp was then stored offline and the trajectory to the object was planned using commercial software. No grasping data was shown and no results of grasping were presented.

Several approaches have tried using a combination of trained models and sensor data. Saxena, Driemeyer and Ng used a STAIR 1 robot equipped with an articulated arm, parallel jaw gripper, PGR Bumblebee2 stereo vision system and a wrist mounted Logitech Quickcam Pro camera [18]. They proposed a strategy of looking for “good areas” in images for grasping, such as the stem of a wine glass or handle of a mug. A

probabilistic model was trained via supervised learning using synthetic object and image data. In their experiments, the first image was used to find a suitable grasping point on the object based on the probabilistic model. A second image from a different known camera location was used to find the grasping point in 3D. An evaluation of grasp success rate and mean absolute error in the predicted position of the grasp point was presented for 15 real objects. The grasping test set consisted of six objects similar to those in the training set, and nine other objects. Each object was evaluated during a four grasp trial with the exception of a stapler which was evaluated with a 10 grasp trial. The average grasp success was about 90%, and the average error in the position of the grasped object was about 20 mm.

The idea of using minimum volume bounding boxes as shape primitives for grasp planning was proposed in [19]. Each object was decomposed into a union of bounding boxes. Boxes were chosen for their simplicity; properties such as volume and centre of mass, could be easily computed. It was also noted that specific tasks such as "show the object" could be accomplished by grasping the smallest box. The grasp candidates were also easy to compute. For each box the approach vectors were taken as the normal of a face, and the four edges of the face were used to orient the gripper. The grasp candidate set could be reduced using geometric constraints such as the impossibility for the gripper jaws to reach through the toy duck's head-box. A distance map, calculated as the distance from the face of the box to the object, was formed and then down sampled to a

15x15 grid. The width, height, and depth information of the boxes along with the grid were fed as inputs to a neural network whose output was a planned grasp for the object. For the neural-network learning, the grasps were simulated using the GrasplIt! simulator [15]. During testing by further simulations, the set of candidate grasps was reduced using geometric constraints, the remaining grasps were fed to the neural network used to generate the grasp. The duck example had 86,310 points. Obtaining the box-based representation took 22 seconds, while applying the geometric constraints and computing the neural network output took less than 0.5 seconds on a Double Intel Core2 Quad 2.66GHz computer. No experimental results were included.

Goldfeder *et al.* [20] presented an approach that used partial data from sensors mounted on a robot, and similar stored object models for grasp planning. Several images were taken from locations selected to reside on a spherical cap. These images are processed to provide a depth map that is used with the bag of features shape invariant feature transform (BF-SIFT) algorithm presented in [21]. The BF-SIFT algorithm finds a set of models from a stored database that closely match the object being scanned. After the model was selected, the sensor data was aligned with the model. Their grasp selection algorithm used between 5 and 15 previously stored form closure² grasps from a database. Due to the incompleteness of the sensor data, the model from the database and the physical object are not necessarily the same, so some grasps may fail. To help

² Please see section 6.3.1 for the definition of form closure.

reduce these errors the grasp candidates were evaluated using the Graspl! [15] simulator with a Barrett hand. A static analysis was used to evaluate form closure grasps with a pass/fail criterion.

To choose grasps for a PowerCube parallel jaw gripper mounted on a six DOF Staubli RX60, Popovic *et al.* [22] used a fixed Bumblebee2 stereo vision system and computed the 2D contour features from each camera. An exhaustive search was used to match each computed contour with a corresponding contour in the other image with the end result being either a matched pair forming a 3D contour, or the contour being removed from the set. Next, planes were fit to the 3D contours. Their algorithm generated four distinct grasps for each 3D contour. The grasps were tested on 14 household objects with various shapes such as cups, bowls and a stapler using 100 trials. The number of successful grasps, unsuccessful grasps, collisions, unstable grasps, and failures to generate grasps were presented for each object. The best object was a blue plastic frying pan with a 67% successful grasp rate. There were four worst case objects where the grasp success rate was 0%. Of these objects, a cookie cutter in the shape of a small dog was the worst with a 66% failure rate to even compute a possible grasp. The algorithm was also tested in a complex scene where it was allowed 30 grasp attempts to remove items from the table. The algorithm was extended with a learning algorithm in Bodenhagen *et al.* [23]. The extended algorithm evaluated each grasp based on the success rate of initially touching the object with both gripper jaws, and the success rate

of holding it aloft after movement. This data was fed to a neural network that was trained using both offline and online learning. For the learning and subsequent testing, evaluation of the grasps was done for three sets of objects: cylindrical, non-cylindrical, and a combination of both. Each set of test objects was grasped using the following three methods. First, as a baseline, unlearned grasps were used. For each of the cylindrical and non-cylindrical sets 256 grasp attempts were made. The offline and online learning testing had the objects split into two separate groups, a learned set and test set. A combination of 500 iterations of learning and grasp attempts were made. With the test set experiments, the grasps were successful approximately 40% of the time.

Bohg *et al.* [24] used a 6 DOF Kuka arm with a Armar III robotic head equipped with two stereo camera's and a three fingered Schunk Hand; as well as a Tombatossals torso system with a 7 DOF Mitsubishi PA10 arm equipped with a 4 DOF Barrett Hand, two head mounted DFK 3 BF03-Z2 cameras and a chest mounted Videre DcSG-STOC stereo camera. Stereo images of the object were taken. To estimate the unseen areas of objects placed on a table they assumed that most objects in service applications have symmetrical properties. Grasps were based on the algorithm presented in [17] where the centroid and boundary of the object were used and the fingertip positions were based off of the closest point to the centroid on the surface of the object and a point on the opposite side of the object in line with the centroid and the first fingertip location.

They experimentally evaluated their modeling system with seven objects placed with various orientations, and quantified the deviations between the modeled and true visual centroids. The chosen objects included mostly symmetric objects such as cans, cups and boxes. Only one object was grasped in an attached video and no statement was made on the number of successful trials.

To develop an object model for grasping, Dragiev *et al.*[25] used a Schunk seven DOF arm, Schunk seven DOF hand, and a Bumblebee stereo camera. Each object was modeled using surfaces estimated based on sensor data using Gaussian process implicit surface potentials. These potentials were used so that the sensor data could include surface point measurements (e.g. from a touch probe or a laser scanner) as well as surface normal measurements (e.g. from a haptic sensor). Their grasp planning algorithm uses the normal vector, and the potential field created by the Gaussian surfaces model to determine the grasp stages (described below), as well as the hand orientation. The potential field, defined by a function $f(x)$, was normalized such that at far distances $f(x) = 1$ and during contact $f(x) = 0$. The grasping routine consisted of three stages. First, when the arm was far away $f(x) \approx 1$ the hand was opened and the arm was moved towards the object. Second, when the arm was close to the object $0 < f(x) < 1$, and the hand continued to approach the object while orienting the fingers of the gripper along the normal of the surface to be grasped. Third, when at the surface, $f(x) = 0$, the fingers were closed using a pressure sensor to determine the contact

forces. No real world model generation data was shown. The grasping algorithm was shown to work for a ball, cylinder and cube in an attached video, but no mention of the number of successes or failures was mentioned.

Jiang, Moseson and Saxena used a six DOF Adept Viper s850 arm equipped with a parallel jaw gripper and a Bumblebee2 stereo camera [26]. A grasp was generated that used the pose of the object; as well as the gripper jaw width and jaw separation. Each grasp was generated using a two-step approach. Potential grasps are first found from the stereo image data using 17 histogram filters, and then the grasp was chosen based on a training set of grasps that were generated by the user. Evaluations of the grasps include the percentages for reaching the grasping location (without a collision), and grasping/holding successfully. Twelve objects including wire strippers, a martini glass, shoes, and a window wiper were presented. The average grasping success rate was 88%. The number of trials for each object was not stated.

Aleotti and Caselli [27] decomposed each test object into parts using Reeb graphs [28, 29], the decomposed objects were then matched to a predefined type stored in a database. Next, previously learned grasps (from human demonstration) for the matched database objects were used to plan the grasp for the test object. They included simulation results for five objects.

2.2.2 ***Compliant grippers***

Grasping in unstructured environments can sometimes be accomplished using compliant grippers. For example, Dollar and Howe [30] simulated a series of two DOF under-actuated grippers and evaluated how joint stiffness affected their ability to grasp different sized objects. They later extended this by building the best gripper design, and testing it compared to a simple manually adjustable joint angle gripper [31]. Their gripper was able to grasp 50mm objects with up to 100% position error, and 100mm objects with 33% position error. The main reason for its insensitivity to position errors was that it used a very large finger opening distance. This ensured that the target object would always lie between the jaws. This approach requires a relatively large and heavy gripper, and also relies on being able to contact all surfaces of an object (i.e. no restrictions on the grasping locations can be imposed), and the local environment must be free of other objects.

2.3 **Grasping Evaluation**

Testing of grasping routines also varies greatly from paper to paper. Several papers demonstrate a single object being grasped only once, e.g. [32], while others use multiple objects and define an arbitrary number of trials (sometimes changing between objects) and present the results as a grasping percentage, e.g. [20]. Another paper presented mean absolute position error [18]. Papers also have different definitions of whether an object is successfully grasped, usually it is assumed that a successful grasp occurs when

the object is held by the gripping jaws, but some papers add a minimum time constraint, e.g. the gripper must hold the object in its jaws for x time.

2.4 Summary

In recent years, the area of vision-guided robotic grasping research has been very active. With the advent of affordable cameras, inexpensive fast computers, and open-source computer vision software, researchers have investigated the use of 3D vision-based modeling methods to obtain accurate and detailed models of the object to be grasped. Creation of models for unknown objects is still very difficult due to the complex algorithms that are needed, and the poor reliability of the sensor data. For these reasons, many 3D vision modeling systems still need prior knowledge of the object from a stored model, and/or human intervention during the modeling/grasp planning process. Many researchers never implement their modeling/grasp planning strategies on a physical robot due to the ease of simulating grasp planning strategies. This trend has increased due to availability of the free GrasplIt! simulator [15]. Grasping implementations are split when it comes to end effectors. Multi-fingered hands are able to apply a variety of grasp types, e.g. pinch grasps and wrap grasps. Compared to the complex multi-fingered hand, the parallel-jaw gripper is still the most widely used in industry because of its low cost, low weight, simplicity and high reliability. For a parallel-jaw or two-fingered grasp, most existing grasp planning methods are based on two point contacts on opposite sides of the object. A point contact grasp is sensitive to the

position and orientation errors that always exist when an object is modeled using sensor data. Furthermore, some grasp planning algorithms for parallel-jaw grippers are based on a simplified 2D object model and cannot be applied to most 3D objects. Other grasp planning algorithms apply only to specific object shapes, such as cylinders.

The research in this thesis will address some of the limitations of the prior automated grasping research. In particular it will extend the previous work reported in [16] by developing a new grasp planning method; revised model creation based on the new grasping method, and a method for evaluating the quality of the physical grasps. These methods will be implemented on robotic hardware, extensive experiments performed and the results analyzed.

Chapter 3

Software and Hardware Overview

3.1 System Hardware

The grasping setup consists of a 6 DOF CRS F3 articulated robotic arm and C500 controller. The arm was rigidly mounted to a metal table. Its workspace is shown in Appendix A. The robot end effector was fitted with a custom mount housing a Robohand RPL-4 pneumatic parallel-jaw gripper operating at an air pressure of 40psi (to provide a gripping force of $\sim 26\text{N}$), and two $102\text{mm} \times 60\text{mm}$ jaws; and a Point Grey Research Dragonfly2 DR2-COL video camera, containing a Sony ICX424AL $1/3''$ CCD with a resolution of $648 \times 488\text{pixels}^2$ at 60 frames/second. The eye-in-hand camera was connected to a computer via IEEE-1394a (Firewire). A $3.5\text{-}8\text{mm}$ manually adjusted variable focal length auto iris lens was used. For structured light a computer controlled 5mW line laser was mounted on the end effector at an angle of 240° from the positive z axis with the laser line parallel to the camera's y axis. A custom designed USB controlled 30 red LED ring light was used for lighting the object during image capture. This hardware is shown in Figure 3-1. Layers of black foam were placed in front of the robot where the object was to be gripped to reduce the impact on the robot gripper assembly if an incorrect grasp was attempted. A "Playfield", shown in Figure 3-2 was added after initial baseline tests to force any object on it into a non-standard orientation and position.

Images for grasp evaluations and pose estimation were taken with a separate Point Grey Research Dragonfly2 video camera with a 25mm lens. This camera was rigidly mounted to the side of the robot controller. Illumination was provided with two computer switched halogen lights. A webcam was also used to record grasp attempts for analysis. The image processing and high level control hardware used for the experiments consisted of a 3.8GHz P4 PC. The robot motion commands are transmitted from the PC to the robot controller via an RS-232 serial connection. The code was written in C++, using the open source software libraries OpenCV [33] and POSIT based on [34]. Communication to the robot was done through a serial connection to the robot controller running a RAPL-3 program.

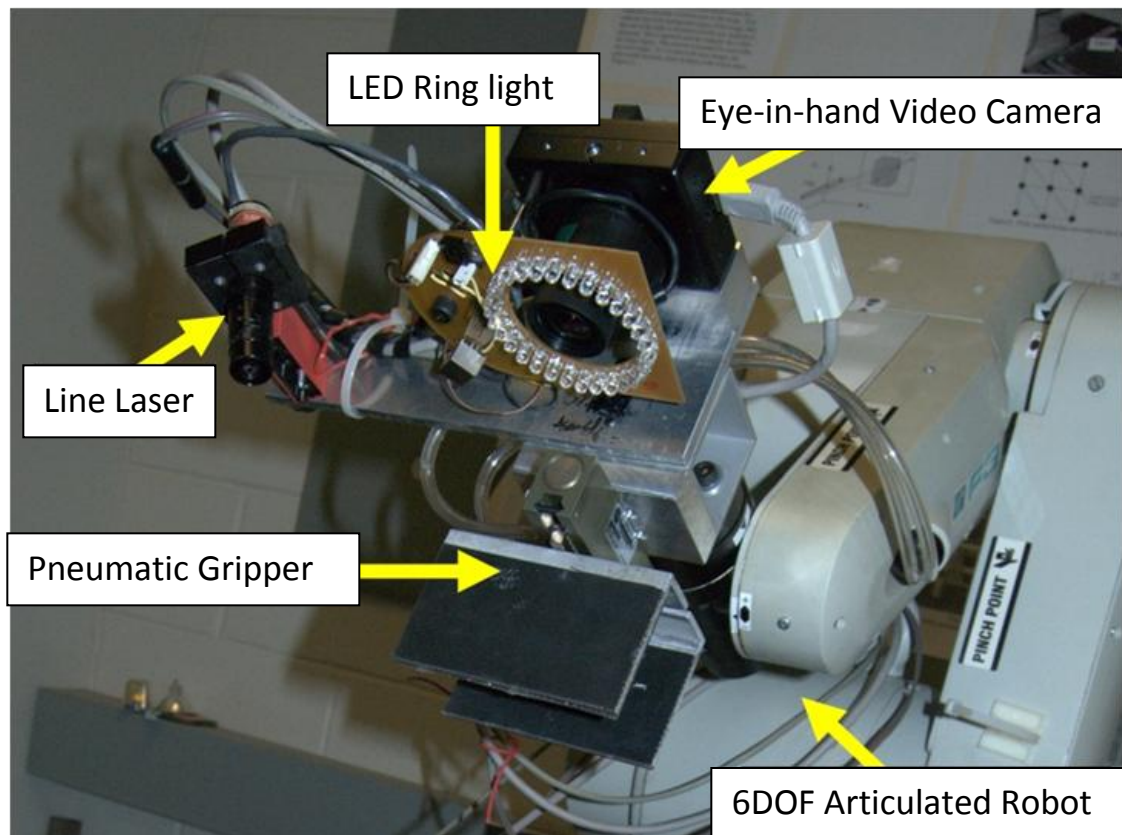


Figure 3-1 Eye-in-hand video camera, line laser, custom ring light and pneumatic gripper mounted on the 6DOF robotic arm.

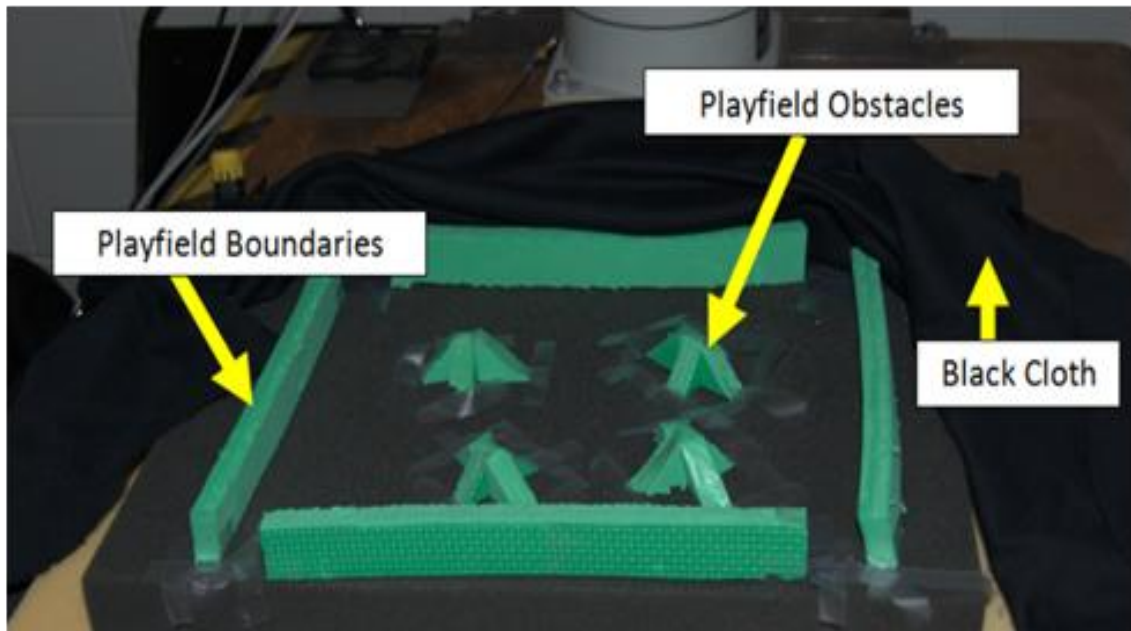


Figure 3-2 Exposed playfield showing obstacles.

3.2 Execution Overview

The execution begins with the object being placed on the table. This is done manually or using the automated procedure described in Section 8.5.3. Next, the robot captures an image using the eye-in-hand Dragonfly 2 camera centred above the table. It then finds the object on the table and moves the camera directly above it. Several more images are taken of the object from different viewpoints. These are used to create a preliminary object model using a silhouetting technique. A laser scan is then used to accurately identify and reshape the top of the model. The grasp is then calculated and executed. After the grasp is complete the pose estimation camera is used to evaluate the success of the grasp. This process is shown in Figure 3-3.

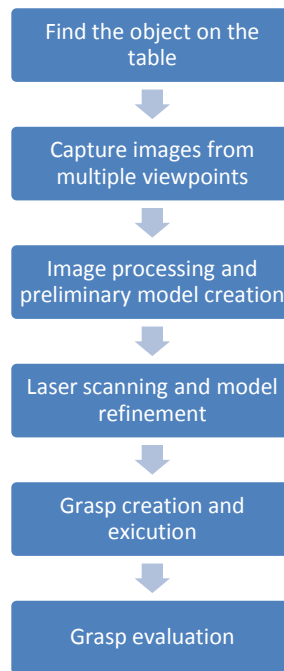


Figure 3-3 Overview of the system execution.

Chapter 4

Camera Calibration

4.1 Objective

Points defined in an image taken by a camera are defined in *image coordinates*. This 2-D coordinate frame has units of pixels. The actual position of the physical object is defined in *world coordinates*. This 3-D coordinate frame has units of mm. The objective of camera calibration is to transform points from image coordinates to world coordinates.

4.2 Relationships between the Coordinate Frames

Figure 4-1 shows all of the coordinate frames that are used for the grasping system. It also demonstrates the relationships between each frame.

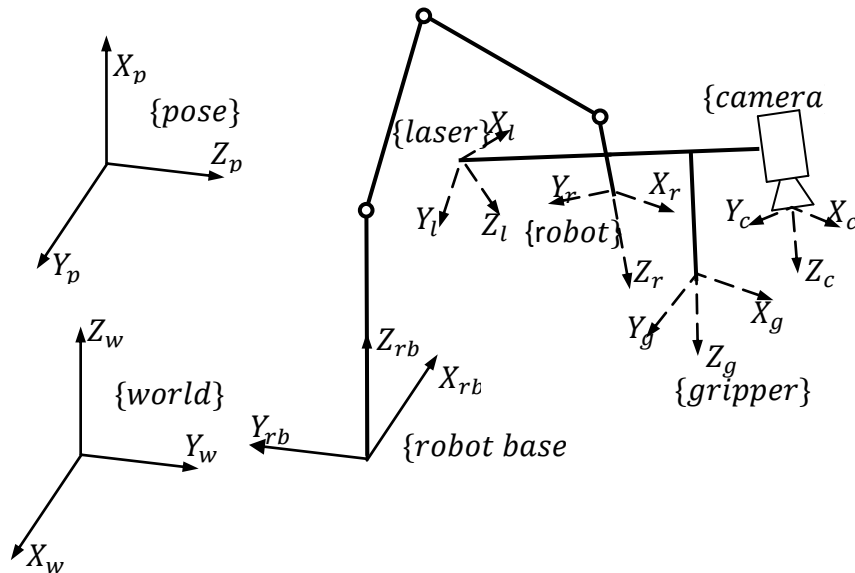


Figure 4-1 Coordinate Systems for the: w world, p pose, rb robot base, r robot, c camera, g gripper, and l laser frames.

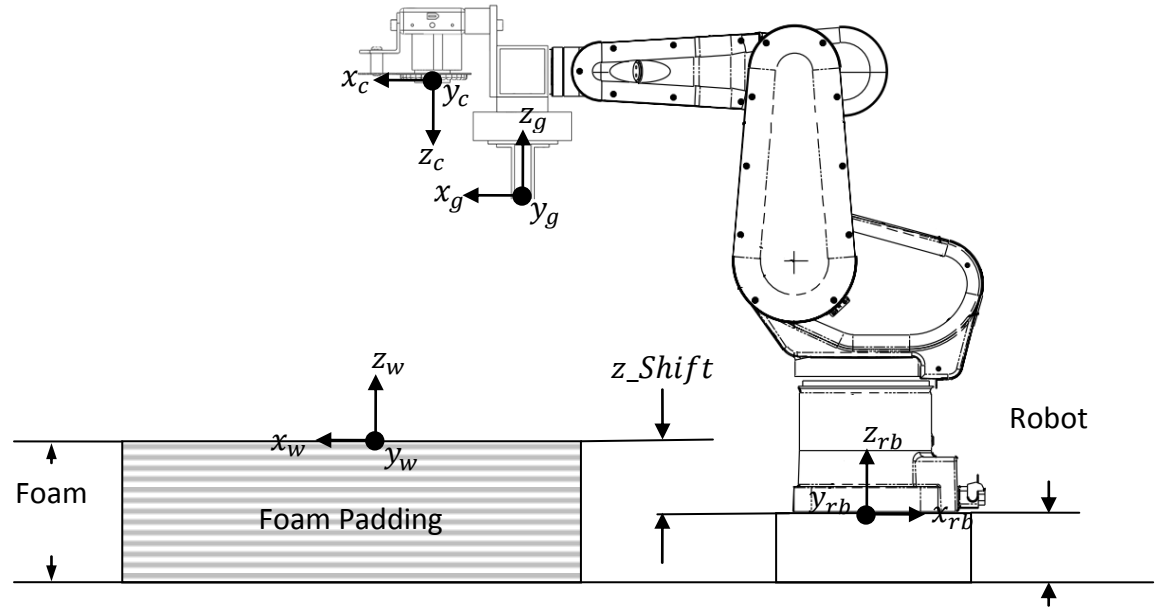


Figure 4-2 Robotic Setup Side view

The world frame $X_w-Y_w-Z_w$ and robot base frame $X_{rb}-Y_{rb}-Z_{rb}$ were aligned such that the corresponding axis was parallel but with opposite x and y axis directions, see Figure 4-1 and equation (4-1). The robot base to robot homogeneous transformation H_{rb}^r was calculated by the CRS C500 controller using a world position vector and roll-pitch-yaw angles. Note that the world position and orientation (x_w, y_w, z_w) and $(x_{rot_w}, y_{rot_w}, z_{rot_w})$ were sent to the robot controller and it calculated joint rotation angles.

$$H_w^r = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4-1)$$

The calculation of the homogeneous transformation from the robot frame (X_r, Y_r, Z_r) to the gripper tip centre frame (X_g, Y_g, Z_g) is pre-calculated based on the dimensions of the gripper jaws and mount. The transformation includes a rotation as well as translation because the orientation for mounting the gripping assembly to the robot is at a rotation of $-\frac{\pi}{4}$ along the robot Z_r axis; this is shown in Figure 4-3.

The transformation is calculated as:

$$H_r^g = \begin{bmatrix} \cos\left(\frac{\pi}{4}\right) & -\sin\left(\frac{\pi}{4}\right) & 0 & \frac{L_g}{\sqrt{2}} \\ \sin\left(\frac{\pi}{4}\right) & \cos\left(\frac{\pi}{4}\right) & 0 & \frac{L_g}{\sqrt{2}} \\ 0 & 0 & 1 & t_{zg} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4-2)$$

Where

$$t_{zg} = 26.1 \text{ mm}$$

$$L_g = 144.59 \text{ mm}$$

This can be seen in Figure 4-3 and Figure 4-4 where L_g is the distance from the robot end effector to the gripper tip centre on the Z_r plane, and t_{z_g} is the distance along the Z_r axis.

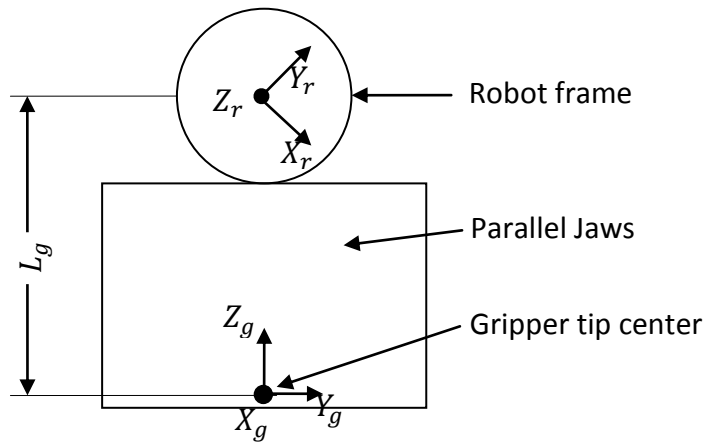


Figure 4-3 Gripper transformation parameters shown in the X_r - Y_r plane.

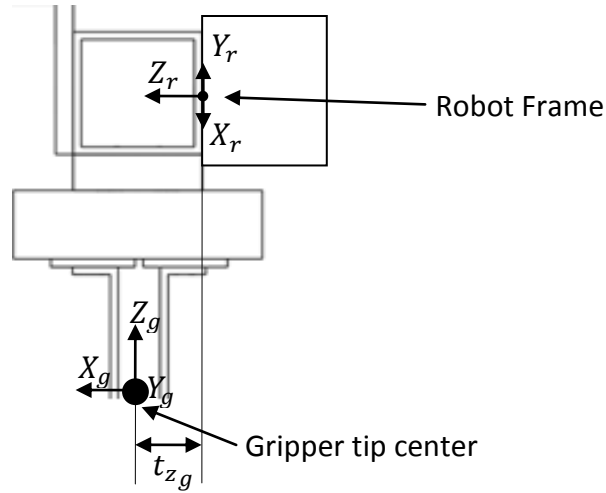


Figure 4-4 Gripper side view shown in the X_g - Z_g plane.

The robot frame to camera transformation H_r^c was calculated using (4-3) based on a calibration routine that measured the horizontal x and y world distances between the optical axis of the camera Z_c and the centre of the parallel gripper (X_g, Y_g) these distances were termed the x and y camera offsets: $cameraToToolCenter_x$, and $cameraToToolCenter_y$. t_{zg} is the Z_r translation from the robot frame to the gripper frame in the robot frame.

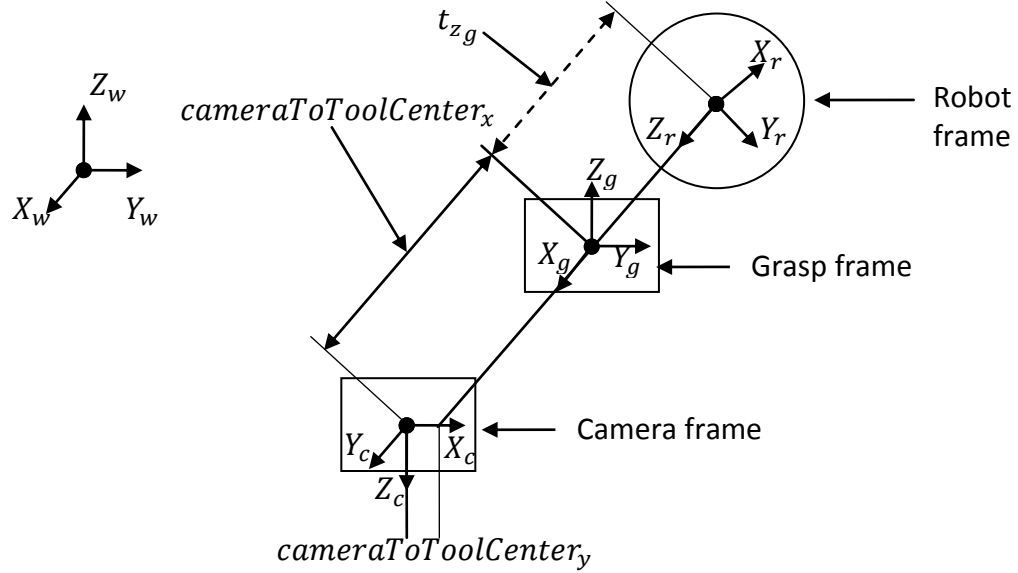


Figure 4-5 Input image transformation between the robot, grasp, and camera frames.

$$H_r^c = \begin{bmatrix} 0 & \cos\left(\frac{\pi}{4}\right) & -\sin\left(\frac{\pi}{4}\right) & \frac{cameraToToolCenter_y}{\sqrt{2}} \\ 0 & \sin\left(\frac{\pi}{4}\right) & \cos\left(\frac{\pi}{4}\right) & \frac{cameraToToolCenter_y}{\sqrt{2}} \\ 1 & 0 & 0 & t_{zg} + cameraToToolCenter_x \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4-3)$$

The robot frame to laser frame H_r^l was calculated using:

$$H_r^l = \begin{bmatrix} -\sin\left(\frac{\pi}{6}\right) & 0 & \cos\left(\frac{\pi}{6}\right) & -\frac{L_1}{\sqrt{2}} \\ \cos\left(\frac{\pi}{6}\right) & 0 & \sin\left(\frac{\pi}{6}\right) & -\frac{L_1}{\sqrt{2}} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4-4)$$

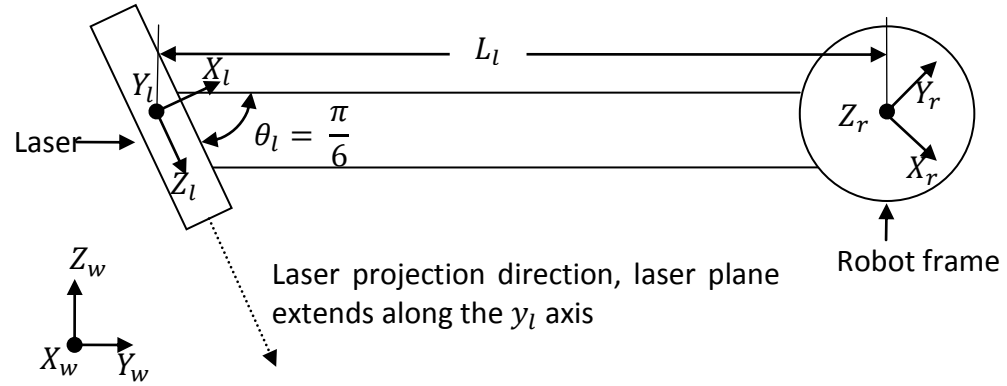


Figure 4-6 The robot to laser transformation parameters for the matrix H_r^l .

Where

$$L_l = 155mm$$

4.3 Details of the Camera Calibration

4.3.1 Key Locations and Notation

There are several image positions that must be defined for later use. The **overhead image** is when the camera is facing down ($Z_c = -Z_w$) at the centre of the tabletop. This position is used to initially find the object. The **slanted images** encompass the set of images that are taken around the object to create the computers 3D model, these image locations are further defined in Section 5.2. The starting location is where the robot

resets itself when the program starts, it is in the same orientation as the overhead image location but its z_w location is smaller.

4.3.2 **Calibration Method**

Calibration of the cameras was done using Tsai's method [35]. It was implemented using software originally written by Reg Willson from Carnegie-Mellon University [36] (The software source is now unavailable). It calculates Tsai's 11 parameter camera model.

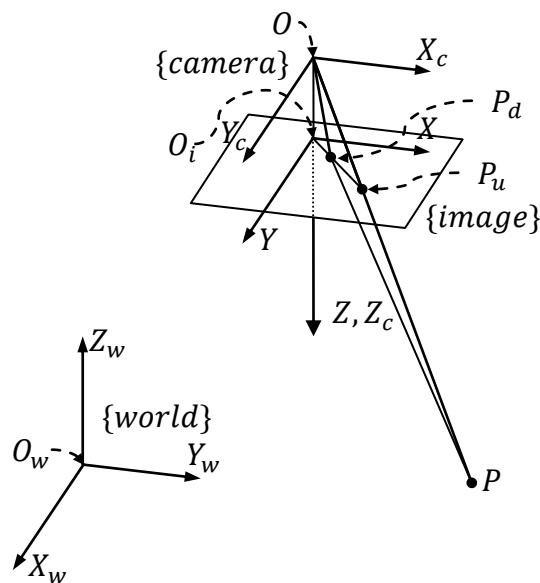


Figure 4-7 Tsai Calibration system Image from [35]

Tsai's method is a four step process using the following shown in Figure 4-7 and Figure 4-8:

(x_w, z_w, y_w) , the world coordinates of point P

(x_c, y_c, z_c) , the camera coordinates of point P , centred on the optical centre O .

(X, Y) , the image coordinate system centred around O_i , (X, Y) is parallel to (x, y) and O_i is a distance of the focal length along the z axis.

$P_u = (X_u, Y_u)$, the coordinate of the $P(x, y, z)$ on the front image plane using a perfect pinhole camera model.

$P_d = (X_d, Y_d)$, the actual coordinates of $P(x, y, z)$ on the front image plane differing due to lens distortion.

Step 1: Calculate matrices R and T

The transformation between (x, y, z) and (x_w, z_w, y_w) is given by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T \quad (4-5)$$

Where

R is a 3x3 rotation matrix

$$R = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \quad (4-6)$$

and T is a 3x1 translation matrix

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad (4-7)$$

Step 2: Calculate the effective focal length f

Using a prospective projection (x, y, z) can be converted to undistorted image coordinates (X_u, Y_u) as follows

$$X_u = f \frac{x}{z} \quad (4-8)$$

$$Y_u = f \frac{y}{z} \quad (4-9)$$

Step 3: Calculate the Radial lens distortion through the distortion coefficients k_i

The radial distortion (D_x, D_y) is calculated using

$$X_d + D_x = X_u \quad (4-10)$$

$$Y_d + D_y = Y_u \quad (4-11)$$

Where

$$D_x = X_d(k_1 r^2 + k_2 r^4 + \dots) \quad (4-12)$$

$$D_y = Y_d(k_1 r^2 + k_2 r^4 + \dots) \quad (4-13)$$

$$r = \sqrt{X_d^2 + Y_d^2} \quad (4-14)$$

Tsai found that for industrial machine vision only the first term of radial distortion needed to be considered, tangential distortion and the higher terms of radial distortion were found not to create an improvement and are ignored.

Step 4: Calculate the uncertainty image scale factor s_x

Real image coordinate (X_d, Y_d) to image coordinates (X_f, Y_f) are related by

$$X_f = s_x d'_x{}^{-1} X_d + C_x \quad (4-15)$$

$$Y_f = d_y^{-1} Y_d + C_y \quad (4-16)$$

Where

$$d'_x = d_x \frac{N_{cx}}{N_{fx}} \quad (4-17)$$

(X_f, Y_f) are the pixel row and column numbers of the image pixel

(C_x, C_y) are the row and column numbers of the centre of the image

d_x is the on centre(midpoint to midpoint) distance between adjacent sensor elements in the X direction

d_y is the on centre distance between adjacent sensor elements in the Y direction

N_{cx} is the number of sensor elements in the X direction

N_{fx} is the number of pixels in a line

To relate the 3D world coordinates to the 2D computer image coordinates the following equations were formed based on the combined earlier steps.

$$s_x^{-1}d'_xX + s_x^{-1}d'_xX_{k_1}r^2 = f\frac{x}{z} = f\frac{r_1x_w + r_2y_w + r_3z_w + T_x}{r_7x_w + r_8y_w + r_9z_w + T_z} \quad (4-18)$$

$$d'_yY + d_yY_{k_1}r^2 = f\frac{y}{z} = f\frac{r_4x_w + r_5y_w + r_6z_w + T_y}{r_7x_w + r_8y_w + r_9z_w + T_z} \quad (4-19)$$

Where

$$r = \sqrt{(s_x^{-1}d'_xX)^2 + (d_yY)^2} \quad (4-20)$$

The parameters are then classed into two separate categories,

1. Extrinsic Parameters

The transformation from world coordinates (X_w, Y_w, Z_w) to camera coordinates (x_c, y_c, z_c) consists of three rotation angles yaw (θ) , pitch (ϕ) , tilt (ψ) and three translations $T = (T_x, T_y, T_z)$. R can be redefined using the three rotation angles as:

$$R = \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\sin\theta & -\sin\theta \\ -\sin\psi\cos\phi + \cos\psi\sin\theta\cos\phi & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & \cos\theta\sin\phi \\ \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi & -\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi & \cos\theta\cos\phi \end{bmatrix} \quad (4-21)$$

2. Intrinsic Parameters

The transformation from the camera coordinates (x_c, y_c, z_c) to image coordinates (X, Y) uses five parameters which are:

f the effective focal length.

k_1 the first order lens distortion coefficient

s_x the uncertainty scale factor for x, due to camera scanning and acquisition timing errors.

C_x coordinate of the centre of the computer image plane in the x direction

C_y coordinate of the centre of the computer image plane in the y direction

4.3.3 *Procedure for Coplanar Calibration*

The images were calibrated with a dot grid array consisting of 25 2mm red dots spaced 20mm apart in a 5x5 matrix created using Microsoft PowerPoint; it is shown in Figure 4-8. The image was first filtered using a red threshold to separate the dots from the background. The centroid of each dot was found and passed to the calibration routine.

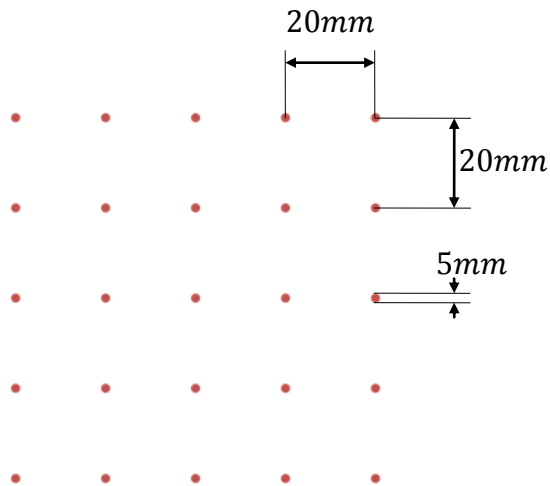


Figure 4-8 Calibration pattern (not to scale).

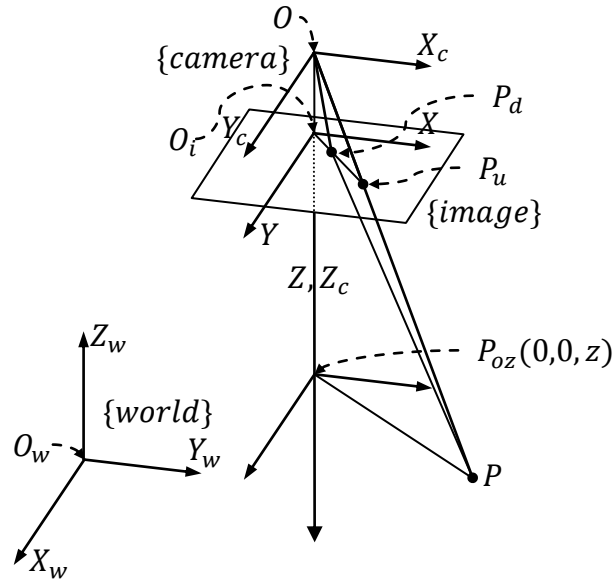


Figure 4-9 Radial alignment constraint from Tsai [35]

Because the calibration pattern is coplanar, the world coordinate system (x_w, y_w, z_w) is chosen such that for the points on the pattern $z_w = 0$. The origin of X - Y is chosen to be at the centre of the image sensor. The origin of (x_w, y_w, z_w) is also chosen such that it is not near the centre of the image sensor or near the y axis of the camera frame. The steps of the coplanar calibration procedure are:

1. Compute the distorted image coordinates (X_d, Y_d) for $i = 1 \cdots N$, The number of calibration points,
 - a. Find the row and column for each calibration point (X_{fi}, Y_{fi})

- b. Using manufacturer information for the camera, find N_{cx} , N_{fy} , d'_x , d'_y
- c. Take C_x, C_y to be the centre of the image frame in memory
- d. Compute

$$X_{di} = s_x^{-1} d'_x (X_{fi} - C_x) \quad (4-22)$$

And

$$Y_{di} = d'_y (Y_{fi} - C_y) \quad (4-23)$$

Note: N must be greater than five to solve the next system of equations

- e. Compute $T_y^{-1}r_1, T_y^{-1}r_2, T_y^{-1}T_x, T_y^{-1}r_4, T_y^{-1}r_5$ from the system of equations resulting from the calibration points (X_{di}, Y_{di}) , and world coordinates (x_{wi}, y_{wi}, z_{wi}) using:

$$\begin{bmatrix} Y_{di}x_{wi} & Y_{di}y_{wi} & Y_{di} & -X_{di}x_{wi} & -X_{di}y_{wi} \end{bmatrix} \begin{bmatrix} T_y^{-1}r_1 \\ T_y^{-1}r_2 \\ T_y^{-1}T_x \\ T_y^{-1}r_4 \\ T_y^{-1}r_5 \end{bmatrix} = X_{di} \quad (4-24)$$

If N is large an over determined set is solved for the five unknown parameters

f. Compute $(r_1, \dots, r_9, T_x, |T_y|)$

Let C be a 2x2 submatrix of R where

$$C \equiv \begin{bmatrix} r'_1 & r'_2 \\ r'_4 & r'_5 \end{bmatrix} \equiv \begin{bmatrix} \frac{r_1}{T_y} & \frac{r_2}{T_y} \\ \frac{r_4}{T_y} & \frac{r_5}{T_y} \end{bmatrix} \quad (4-25)$$

If a whole row or column of C does not vanish then

$$T_y^2 = \frac{S_r - [S_r^2 - 4(r'_1 r'_5 - r'_4 r'_2)^2]^{\frac{1}{2}}}{2(r'_1 r'_5 - r'_4 r'_2)^2} \quad (4-26)$$

Where

$$S_r = r_1'^2 + r_2'^2 + r_4'^2 + r_5'^2 \quad (4-27)$$

Otherwise

$$T_y^2 = (r_i'^2 + r_j'^2) \quad (4-28)$$

Where r'_i, r'_j are the elements that do not vanish.

2. Pick a point where the computer image coordinate is not close to the centre of the image

a. Assign $sgn(T_y)$ to be positive

b. Compute the following:

$$r_1 = (T_y^{-1}r_1) \cdot T_y \quad (4-29)$$

$$r_2 = (T_y^{-1}r_2) \cdot T_y \quad (4-30)$$

$$r_4 = (T_y^{-1}r_4) \cdot T_y \quad (4-31)$$

$$r_5 = (T_y^{-1}r_5) \cdot T_y \quad (4-32)$$

$$T_x = (T_y^{-1}T_x) \cdot T_y \quad (4-33)$$

$$x = r_1x_w + r_2y_w + T_x \quad (4-34)$$

$$y = r_4x_w + r_5y_w + T_y \quad (4-35)$$

- c. If x and X have the same sign and y and Y have the same sign then $sgn(T_y)$ is positive otherwise $sgn(T_y)$ is negative and r_1, r_2, r_4, r_5 , and T_x must be recomputed.

3. If the effective focal length f computed in the next step is greater than zero R is calculated using the following

$$R = \begin{bmatrix} r_1 & r_2 & (1 - r_1^2 - r_2^2)^{\frac{1}{2}} \\ r_4 & r_5 & s(1 - r_4^2 - r_5^2)^{\frac{1}{2}} \\ r_7 & r_8 & r_9 \end{bmatrix} \quad (4-36)$$

Otherwise

$$R = \begin{bmatrix} r_1 & r_2 & -(1 - r_1^2 - r_2^2)^{\frac{1}{2}} \\ r_4 & r_5 & -s(1 - r_4^2 - r_5^2)^{\frac{1}{2}} \\ -r_7 & -r_8 & r_9 \end{bmatrix} \quad (4-37)$$

Where $s = -\text{sgn}(r_1 r_4 + r_2 r_5)$ r_7, r_8, r_9 are computed from the outer product of the first two rows using the fact that the three axis are orthonormal and the right hand rule.

4. An approximation of f and T_z is calculated using an over-determined set of linear equations based on the calibration points Lens distortion is ignored at this time.

$$\begin{bmatrix} y_i & -d_y Y_i \end{bmatrix} \begin{bmatrix} f \\ T_z \end{bmatrix} = w_i d_y Y_i \quad (4-38)$$

Where

$$y_i = r_4 x_{wi} + r_5 y_{wi} + r_6 \cdot 0 + T_y \quad (4-39)$$

$$w_i = r_7 x_{wi} + r_8 y_{wi} + r_9 \cdot 0 \quad (4-40)$$

Note: If the image frame and calibration plane are parallel the system of equations from (4-38) will become linearly dependent.

5. Using equation (4-19) and an optimization routine (the paper [35] suggested the steepest descent method), compute the exact solution for f, T_z and k_1 . Initial guesses are taken as the values calculated in 4 and zero for k_1 .

4.4 Calibration Strategies for Object Modeling, Laser Range Measurement, Initial Object Finding and Pose Measurement

Each location from which the camera needed to take images was separately calibrated. The calibration uses the known world coordinates of the dots, and the camera's image coordinates. The initial overhead and slanted images were taken with the calibration pattern lying flat on the grasping table surface with the robot moving around it. The first camera calibration image to be taken is directly over the robot's starting location and the calibration pattern is placed in the centre of the field of view of the camera. When calibrating, an attempt is made to manually line up the dot pattern so that the rows and columns are approximately parallel to the world X_w and Y_w axes. Since this will never be exactly correct the initial calibration is made using a temporary coordinate system. After the calibration parameters have been found the transformation between the temporary coordinate system and the world coordinate system is found. This transformation is a simple rotation around the world Z_w axis. The system then transforms the modeled calibration dots by this calculated rotation about the Z_w axis. The modified model calibration dots (in the world frame) and the image coordinates (on the image plane) are then used to recalibrate the camera in the world frame with the adjusted world locations dots.

Calibrations are then performed for each slanted location, for each location the image dots must be correlated to their world coordinates. To aid in this the first slanted

position that is calibrated is directly forward of the robot such that the rows and columns of the initial calibration match the rows and columns of the now slanted calibration (i.e. the slant is a positive rotation about y_w). The world coordinates of the dots are known from the previous initial calibration and the slanted calibration is performed. In theory the other slanted locations could use extrinsic parameters transformed to match the camera location defined by the robot's program. Unfortunately it was found that the robot positioning was not accurate enough for this approach, and separate calibrations were needed for each of the other slanted locations. An explanation of the camera positions during model creation is presented in Section 5.2.2.

For the laser scanner the robot moved such that the laser line was parallel to the world z axis. Next, the calibration pattern was placed in plane with the laser light. This orientation is shown in Figure 4-11. Orienting the calibration pattern this way ensured that the camera's y axis was calibrated such that where the laser line made contact with the object the laser line could be directly used as the height without any further calculations as it was calibrated as the camera's y axis.

Additionally, an initial overhead calibration is taken with the camera at a location high enough above the table to see the entire playfield. These calibration parameters are used to initially find the object on the table and move over top of it for image capture (i.e. step 1 of the execution sequence shown in Figure 3-3).

Finally, an image is taken from the pose camera to a pattern held in the location for pose estimation and coplaner calibration is performed.

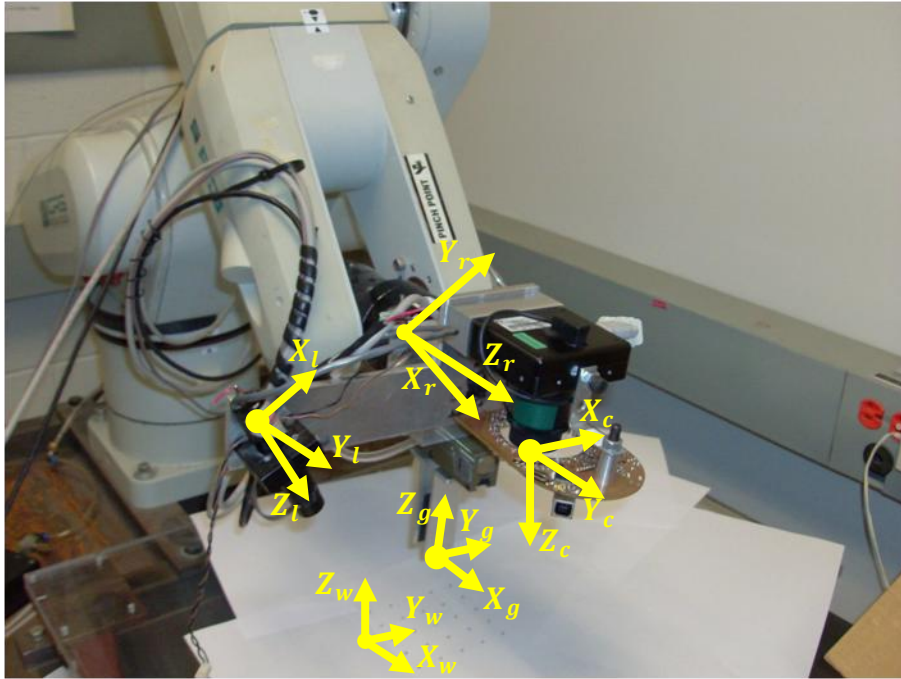


Figure 4-10 First camera calibration overhead used to align the coordinate systems.

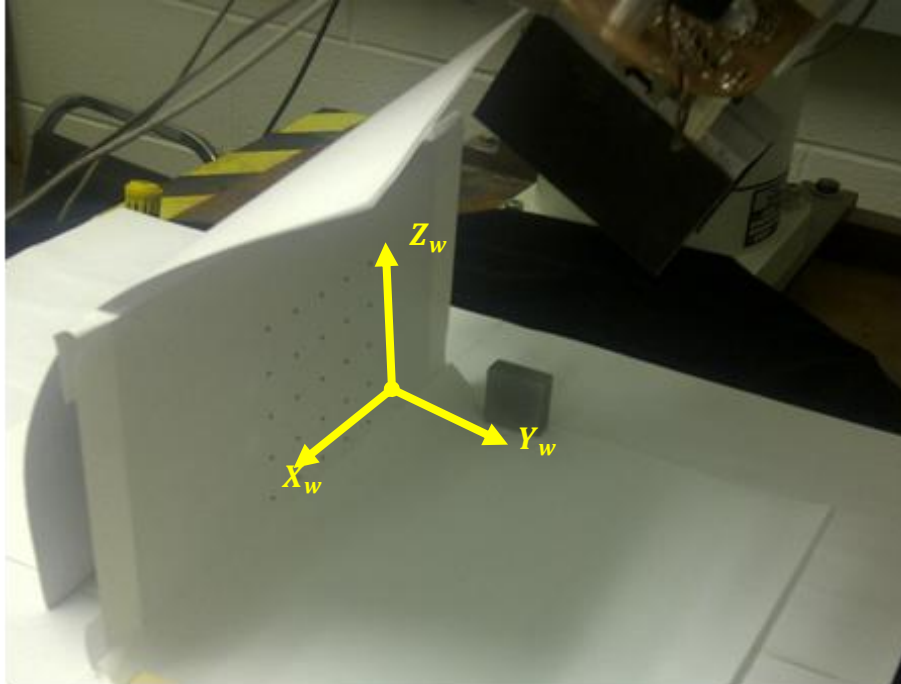


Figure 4-11 Laser scanner calibration setup. The calibration pattern is placed in the same plane as the laser light.

Chapter 5

Model Creation

5.1 Introduction

A 3D model of the previously unknown object is a necessary input to the grasp planning algorithm. This model is first created using the silhouettes of images taken from multiple viewpoints. It is subsequently refined using laser scan data. The final result is a 3D point cloud model of the object.

5.2 Object Model Creation from Image Silhouettes

5.2.1 *Overview*

With silhouette-based modeling, image silhouettes from multiple viewpoints are used to form a solid model of the object. Silhouettes were previously used by many researchers to construct a “visual hull” of an object in 2D or 3D, e.g. [37]. The visual hull contains the object and is a subset of the convex hull of the object when a large number of silhouettes are used. The visual hull typically begins as a block or cube that is larger than the expected size of the object. Each image silhouette is used to remove the sections of the block that do not contain the object. As the number of images increases the visual hull becomes a better approximation of the object, but can only represent it fully if the object is convex. A 2D example is shown with five viewpoints in Figure 5-1. The object is shown in yellow, and the modeling error is shown in red.

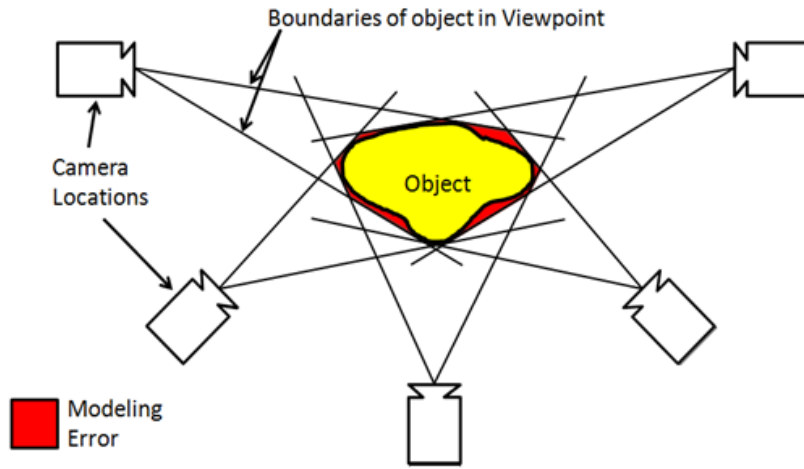


Figure 5-1 Demonstration of silhouette object modeling error.

5.2.2 *Image Capture*

A trade off exists between computation time and accuracy of the model. To increase the accuracy of the model, more images need to be taken and processed, taking more computation time. With less images, the processing time is reduced but the model of the object is less accurate. Another factor in selecting the number of images is the diminishing returns on accuracy. Initially each image makes large changes to the model, but as more images are processed the overall impact of each additional image is less. In our implementation, images of the object are captured from nine viewpoints. The first image is taken with the camera pointing its optical axis in the $-Z_W$ direction at a height of $h_c + z_{drop}$. The other eight images are taken such that the camera is located along a circular trajectory parallel to the Z_W axis with its origin lying passing through the object.

The circle is set at a height h_c above the tabletop. The radius of the circle is determined by the angle θ_{im} . The camera is then rotated around the centre axis by θ . The associated geometry is shown in Figure 5-10. Note that some values of θ cannot be used due to either robot self collisions or the corresponding end-effector location being outside of the robot's dexterous workspace. The equations used to define the camera locations were derived from Figure 5-2, Figure 5-3 and are as follows:

$$h = z_i - (z_{shift} + z_{drop}) \quad (5-1)$$

$$\theta_{im} = \theta_{input_desired} - \frac{\pi}{2} \quad (5-2)$$

$$d_c = h \sin(\theta_{im}) \quad (5-3)$$

$$h_c = h \cos(\theta_{im}) \quad (5-4)$$

$$t_{zim} = d_c - x_{offset} - t_{zg} \quad (5-5)$$

The rotation of the camera is performed using a tool transformation this can be explained with reference to Figure 5-3. The first image is captured with the camera's optical axis directly over the object while looking directly down along the $-Z_w$ axis from a height of $h + z_{drop}$. After this image is taken a tool transform is set such that the centre of the tool is over the object while the camera is located on the circle at the height h_c over the object. To do this t_{zim} must be calculated from (5-5). The tool transformation is then set as $-t_{zim}$. With this tool transformation set the remaining eight images are taken at different angles of θ calculated as:

$$\theta = -1.92 + \sum_{i=0}^7 \frac{\pi}{6} \text{ radians} \quad (5-6)$$

After all the images are taken the tool transform for image capture is removed and the nine images are sent to be processed. An example of a full set of captured images is shown in Figure 5-4 Figure 7-7.

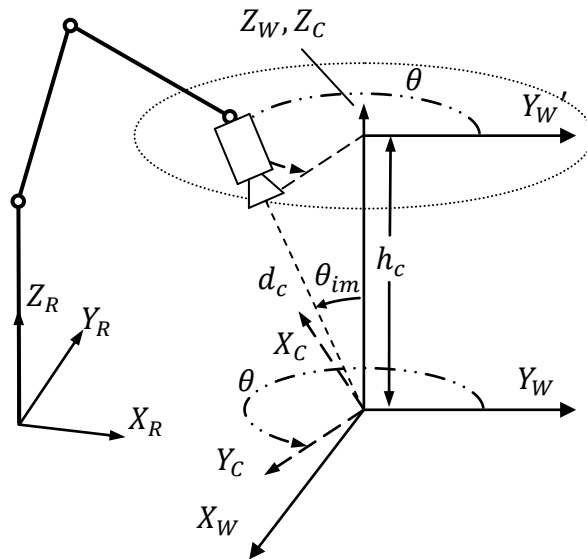


Figure 5-2 Geometry of the image capture locations.

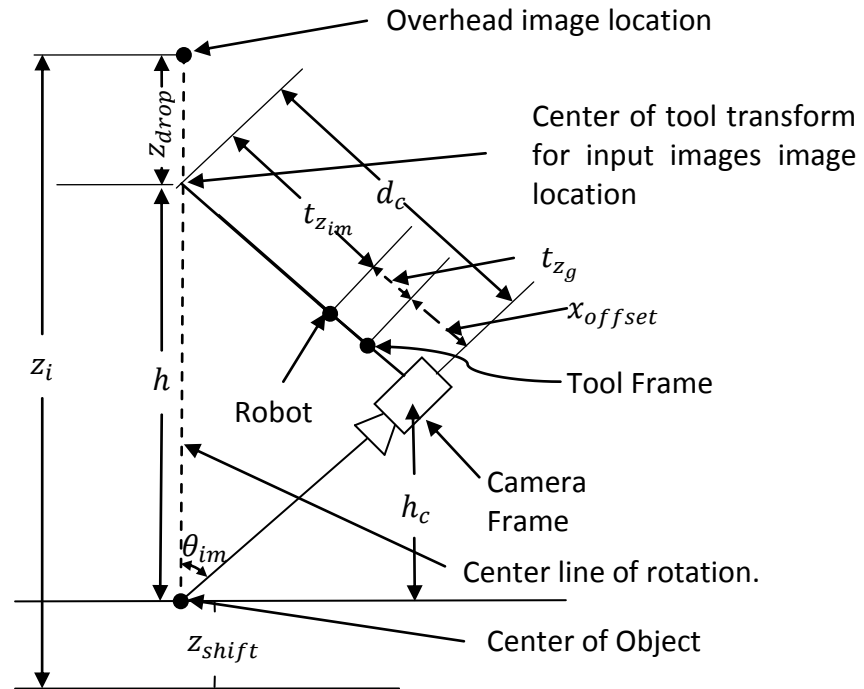


Figure 5-3 Input images transformation

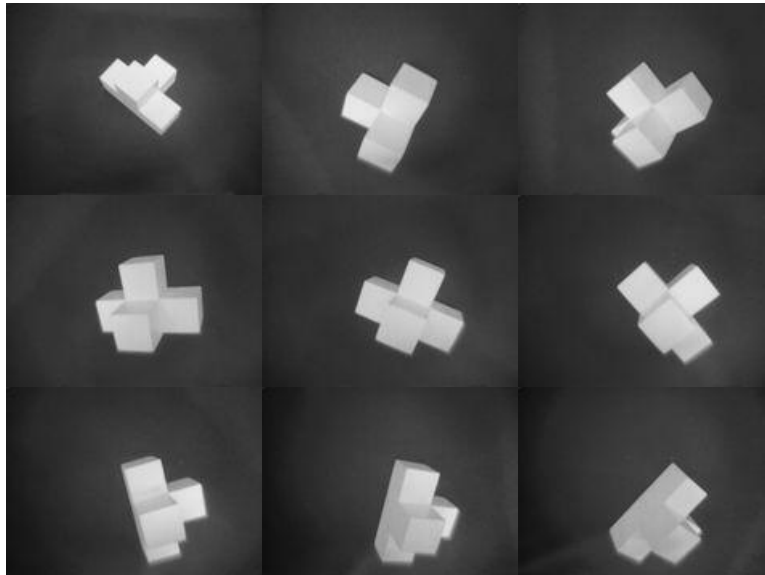


Figure 5-4 An example set of images taken of a T-shaped block.

5.2.3 ***Object Boundary Detection and Silhouette Computation***

The first step in obtaining a silhouette of the object is to enhance the edges in the image. To do this the Sobel filter is used [38]. After the Sobel filter is applied the resulting image is thresholded to detect the stronger edges that occur between the background pixels and the object pixels. When this is finished a boundary following algorithm is run to find the outline of the object in the image [38]. An example of an object going through this boundary detection process can be seen in Figure 5-5. The Sobel filtering result can be seen in Figure 5-5 on the top right. Each image silhouette of the object is obtained by filling the interior area of the detected boundary. Note that to help improve the results of the filtering process a black cloth was placed on the table for a constant non-reflective background and a custom made ring-light (previously shown in Figure 3-1) was attached to the camera to reduce the effects of shadows.

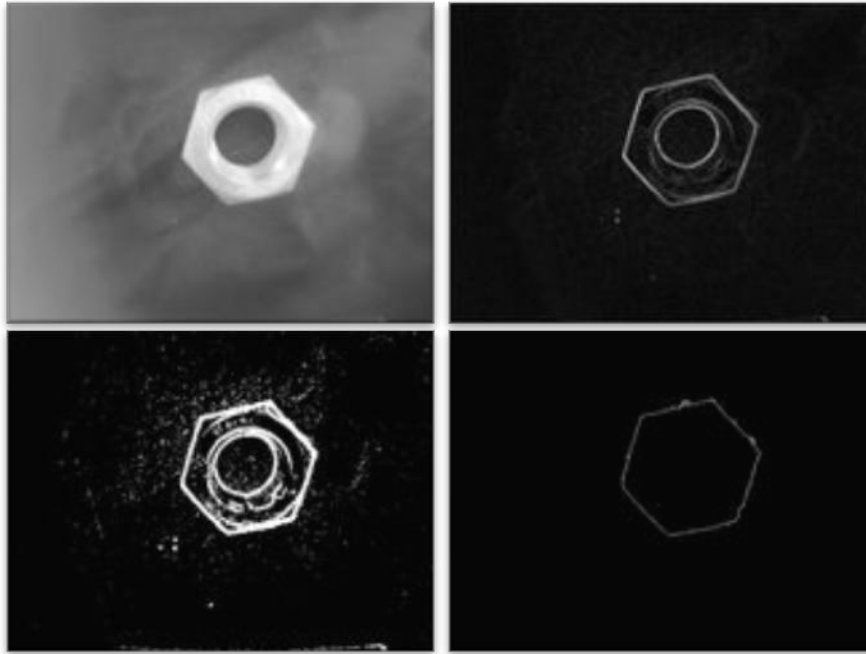


Figure 5-5 An example of an object being filtered. The top left is the original image of the object, top right is the Sobel filtered version of the same image, bottom left is the thresholded version of the filtered image, and the bottom right is the boundary of the object.

5.2.4 *Model Creation from the Set of Image Silhouettes*

For the case of a single image silhouette, the object must lie within the polyhedral cone formed by the image silhouette and the camera focal point, see Figure 5-6. The process of intersecting the series of cones forms a model that is progressively closer (with each additional intersection) to the real object.

A key issue is how to represent the volume elements of the model. In this research we use the beam-based approach presented in [39]. The beam-based approach has the

advantage of a simpler and easier to compute intersection test compared with the more common octree method [9]. The object is first represented by two stacks of square cross-section beams. The first stack is aligned with the X_w axis, while the second is aligned with the Y_w axis.

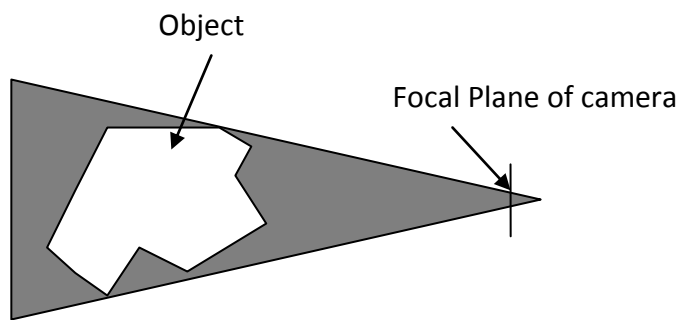


Figure 5-6 A single silhouette image showing the focal plane of the camera. If this was the only image taken then the model of the object would be the shaded cone.

For each object silhouette, and each beam in the image starting with the first beam:

1. Transform the world coordinates of the beam endpoints into the current camera coordinate system. This is done using the camera calibration of each slanted image discussed in Section 5.2.2.
2. Project the beam endpoints onto the image plane of the camera with the silhouette in on it.
3. If either of the projected endpoints of the beam lie outside the calculated silhouette (in the black region) then shorten the beam incrementally by b_{short}

and repeat steps 1 and 2 until both projected endpoints lie on the silhouette (in the white region) or the beam length equals zero (the beam does not exist because it is not in the model). An example of this beam shortening procedure is shown in Figure 5-7.

4. Repeat all the steps for the remaining beams.

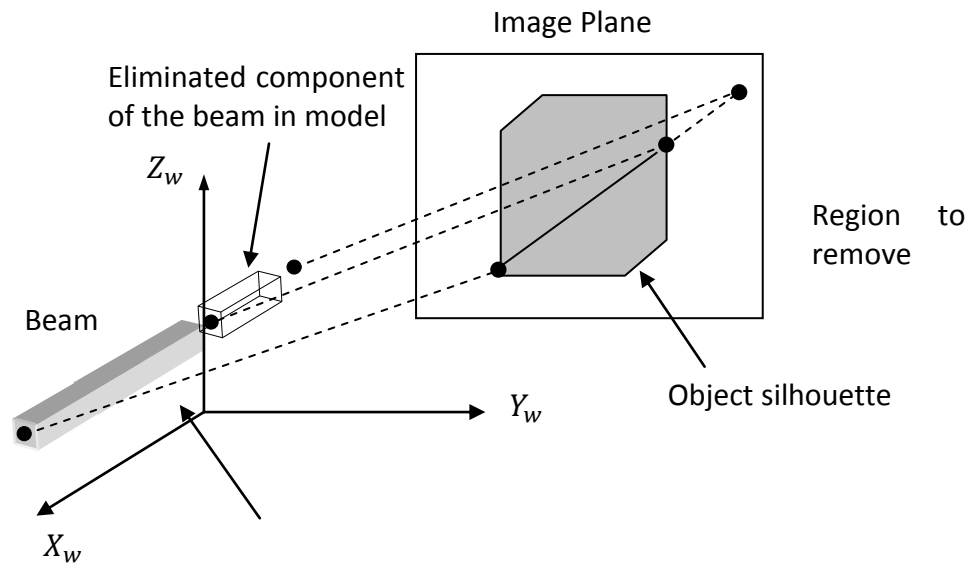


Figure 5-7 Beam based modeling shorting by projecting the model on to the image plane. Image modified from [16].

When the beam shortening procedure is finished the remaining beams form a model of the object. The point cloud of the object is the set of beam endpoints. An example of the model improvement resulting from the addition of each silhouette is shown in Figure 5-8. Since a point cloud is difficult to visually interpret, the convex hulls of the point cloud models are shown in this figure.

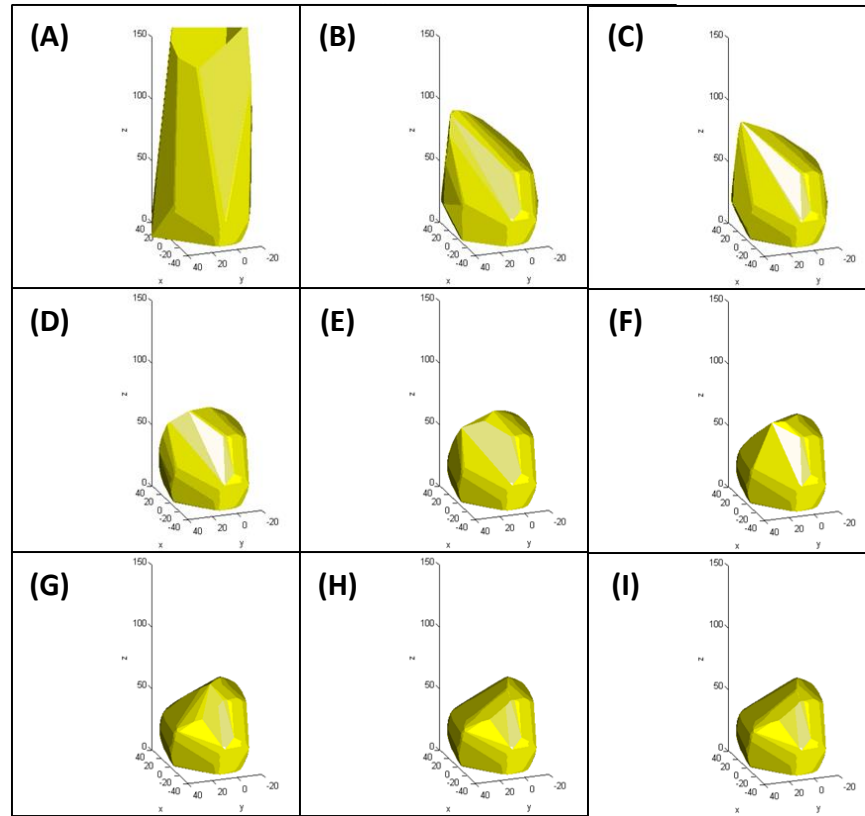


Figure 5-8 Improvement of the model resulting from successively including the image silhouettes from the 8 viewpoints. For clarity the convex hulls of the point clouds are shown. a) overhead, $\theta_{im} = \frac{\pi}{2}$, $\theta = 0$, $h_c = 309mm$, b) Image 2 $\theta_{im} = \frac{\pi}{6}$, $\theta = -1.92$, $h_c = 267.6$ c) Image 2 $\theta_{im} = \frac{\pi}{6}$, $\theta = -1.40$, $h_c = 267.6$, d) Image 3 $\theta_{im} = \frac{\pi}{6}$, $\theta = -0.87$, $h_c = 267.6$, e) Image 4 $\theta_{im} = \frac{\pi}{6}$, $\theta = -0.35$, $h_c = 267.6$, f) Image 5 $\theta_{im} = \frac{\pi}{6}$, $\theta = 0.17$, $h_c = 267.6$, g) Image 6 $\theta_{im} = \frac{\pi}{6}$, $\theta = 0.70$, $h_c = 267.6$, h) Image 7 $\theta_{im} = \frac{\pi}{6}$, $\theta = 1.22$, $h_c = 267.6$, i) Image 8 $\theta_{im} = \frac{\pi}{6}$, $\theta = 1.75$, $h_c = 267.6$

The silhouette-based modeling procedure has the limitation of not accurately modeling the top surface of objects. Since all of the images are acquired from elevated positions, the objects top surface is contained within the silhouettes. This results in excess cone-shaped volumes present on the top surfaces of the models. These excess volumes can be removed by using the method described in the next section.

5.3 Object Model Refinement Using Structured Light System

5.3.1 *Introduction*

After the model is created using the silhouette-based method, there are excess hat-like conical features present on the top surface of the model. If the grasp planner uses any of these excess (and physically non-existent) features to perform a grasp it is likely that the grasp will fail. It is therefore important to remove them before passing the model to the grasp planner. They are removed using a structured light vision system that measures the top surface of the object. The structured light is a line laser attached to the last link of the robot, see Section 4.2.

5.3.2 *Procedure for Generating the Point Cloud Model*

With the red line laser mounted on the robot end-effector, the robotic arm is used to scan a laser stripe across the object while the eye-in-hand camera acquires images. The robot wrist is reoriented such that a vertical plane of laser light is cast on to the object subject see Figure 5-9.

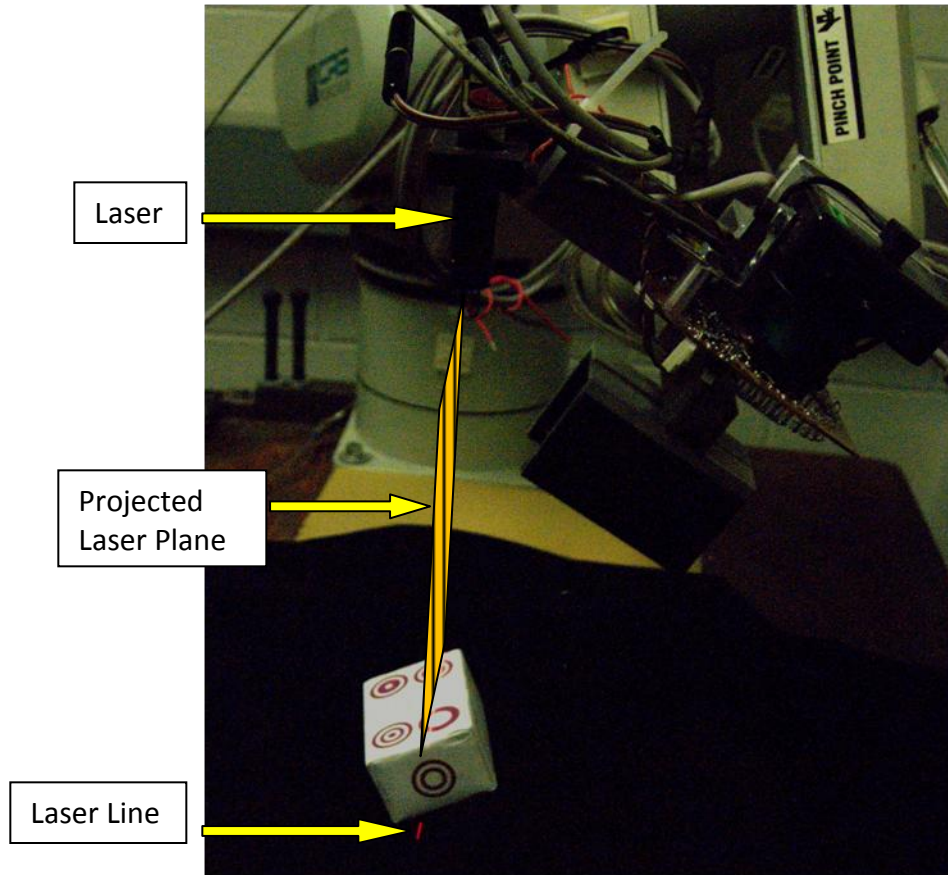


Figure 5-9 Laser projection onto object showing wrist rotation.

The line laser image points are transformed to world coordinates by utilizing the vertical plane calibration shown in Figure 4-11. The end effector speed and the camera frame rate are pre-calibrated and known, allowing determination of the pose of the camera and laser for each image. The camera frame rate f_{laser_cam} , robot linear speed v_{laser} relate to the distance between images d_{laser_cam} according to:

$$d_{laser_cam} = v_{laser} * \frac{1}{f_{laser_cam}} \quad (5-7)$$

The steps of the structured-light image acquisition and model generation are as follows:

1. The object's X_w coordinate boundary is determined from the previously obtained silhouette-based model, and the laser begins at this point oriented parallel to the Y_w axis.
2. The robot then scans a set distance along the $-X_w$ direction while the camera acquires colour images.
3. The red laser line is extracted from the Red-Green-Blue images by thresholding the red channel, followed by thinning. Using a modified convex assumption, explained in the next section, the portions of the line that contact the table are eliminated.
4. For each image, the vertical calibration data is used to convert the laser line image coordinates to Y_w and Z_w coordinates.
5. The X_w coordinate of each image laser plane is found from d_{laser_cam}
6. The structured-light model of the object is stored as a point cloud.

5.3.3 ***Convex Assumption***

Some objects have areas that are not properly detected using the structured light system, for example the hex nut has a large cylindrical void in the centre of it. If this is not adjusted then the structured-light point cloud will have no information about these areas and the base silhouette model, with the erroneous regions are used without modification. In the line laser images regions where this problem occur can be identified

when the line laser is discontinues across sections of the object, see Figure 5-10. To reduce the errors that are created the line laser points were joined together using basic interpolation between them. This modification assumed that the top of the object was concave along the Y_w axis and that the known points (on either side) were higher. When the line could not be seen from the camera it meant that the object (at the contact point of the laser line) was lower than a point previously seen, and that the camera's view was obstructed by the object. Using our assumption, the points that are created are taken as the upper bounds along the Z_w axis.

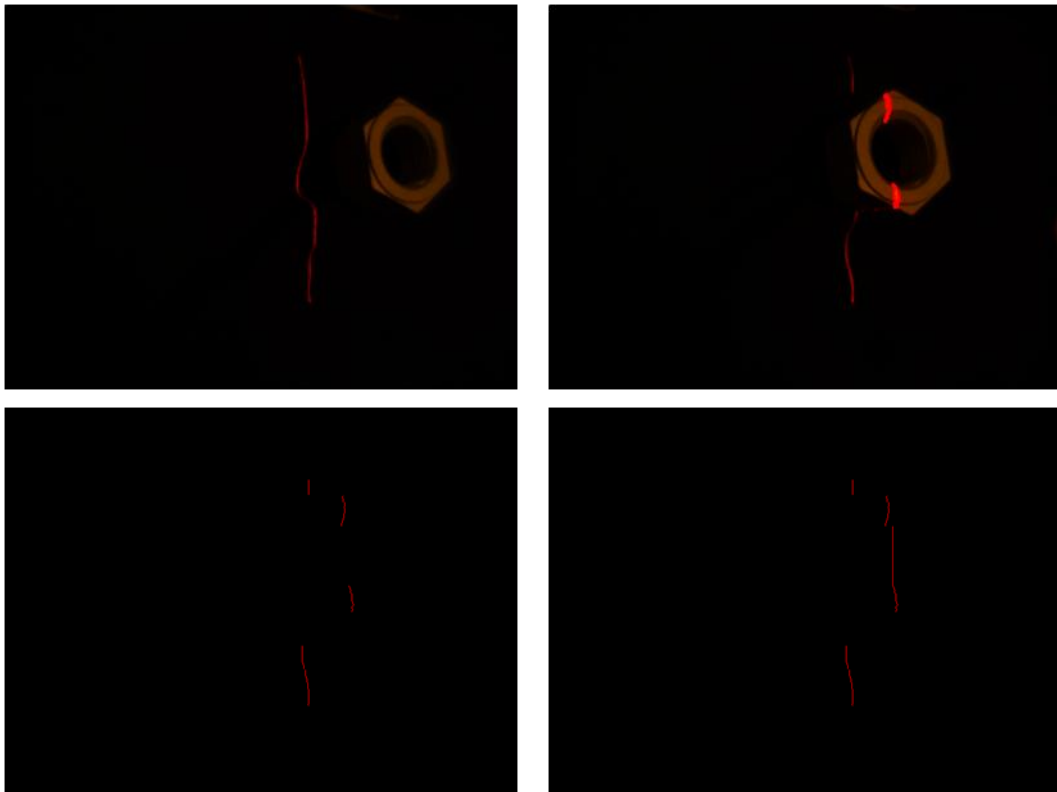


Figure 5-10 This figure demonstrates the laser convex assumption. a) Object and laser line. b) Laser line over center of object. c) Discontinuous laser line. d) Laser line with convex assumption.

5.3.4 *Combination of Silhouette and Structured Light Models*

As discussed earlier the base silhouette model contains excess volume that will be removed using the structured-light representation of the object's top surface. The approach is to use the structured-light surface representation like a cookie cutter, to cut the desired shape out of the silhouette model. The procedure to combine the two models is as follows:

1. Project both models into the Y_w, Z_w plane.
2. Starting with the first silhouette model beam, if either endpoint lies above (ie. greater z_w position) the corresponding structured-light model point, shorten the beam incrementally by b_{short} until both projected endpoints agree with the structured-light model or the beam length becomes zero.
3. Apply step 2 to the remaining beams.
4. Project both models into the world X_w, Z_w plane and repeat steps 2 and 3.

5.4 Parameters Used in Our Implementation

Table 5-1 Parameters used in the grasping program

Parameter	Value
b_{short} beam shortening distance	0.25mm
h	309 mm
z_i	316 mm
z_{shift}	66.4 mm
z_{drop}	15 mm
θ_{im}	$\frac{\pi}{6}$ rad
d_c	154.5 mm
h_c	267.6 mm
x_{offset}	81.22mm
y_{offset}	10.39mm

t_{zim}	$309mm * \sin\left(\frac{\pi}{6}\right) - 81.22mm - 26.1mm$ $= 47.18mm$
d_{laser_cam} distance between images in laser scan	$1mm$
v_{laser} Robot velocity during laser scan	$30mm/s$
f_{laser_cam} Frame rate of camera during laser scan.	$30fps$
θ initial, the initial rotation angle for slanted image capture	$-1.92 rad$
θ rotation the incremental rotation angle for the next 7 images	$\frac{\pi}{6} rad$
number of images taken	8
$\theta_{input_desired}$	$\frac{2}{3} \pi rad$
Tool Translation for grasping	(102.24,-102.24,26.1) (-45.0,0.0,0.0)
Tool Translation for image capture	(0,0,-10.08) (-45.0,0.0,0.0)
Tool Translation for pose position image capture	(88,-88.0,26.9) (0.0,0.0,0.0)
Initial Starting position and Orientation	(430.0,0.0,316.0) (0.0,90.0,0.0)
Pose position and Orientation	(666.0,20.0,410.0) (0.0,90.0,0.0)

5.5 Conclusions

This section described the creation of the object model from the camera and structured light sensor data in detail. The image capture and modeling method originally developed by Bone *et al.* [16] was modified to include the laser convex assumption. The full set of parameters used in our implementation are given in Table 5-1. The assumptions for the structured light system based on the type of object were presented. The computed object model will be passed to the grasping algorithm for the computation of the robotic grasp.

Chapter 6

Grasping Strategy

The following section describes the grasp planning phase of the algorithm. The grasp planner generates a set of grasps, evaluates their quality and selects the highest quality grasp from the set. The section begins with an overview of the strategy followed by some background information needed for the algorithms used. The algorithms forming the grasp planner are then presented in detail.

6.1 Overview of Grasping Strategy

A new grasp planning strategy was developed for a parallel-jaw gripper and a point-cloud model of the object to be grasped. The convex hull of the point-cloud plays a critical role in the planning strategy. The convex hull is employed to reduce the complexity of the model to allow the planning to be executed quickly online using a standard PC. This motivated the decision to make the gripper jaws wide (recall Section 1.1). Specifically, it is assumed that the jaw is as wide as the largest *facet width*, where facet width is defined as the largest distance between the vertices of the facets projected onto the world X_w - Y_w plane. In the low friction case this assumption prevents the jaws from making contact with a non-convex part of the object that could produce an unstable grasp. See Figure 6-1 for a comparison of the unstable/stable contacts that can occur when attempting to grasp with narrow/wide jaws.

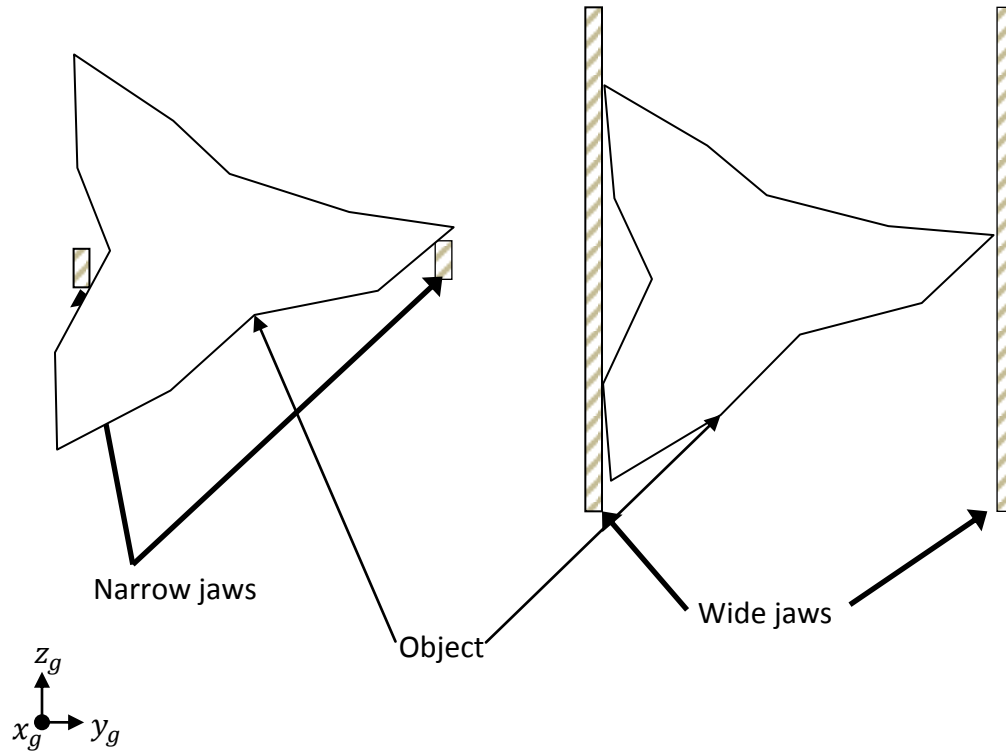


Figure 6-1 Narrow and wide gripper jaws acting on an object. The narrow jaws produce unstable contacts.

The planning strategy was developed for arbitrary objects that could have flat or curved sides and that rest with an arbitrary pose on a surface. The details of the first step, computing the convex hull model of the object, will now be described.

6.2 Convex Hull

The following section describes the convex hull of an object and its use in the grasping routine.

6.2.1 *Definition of Convex Hull*

For clarity the 2-D case will be explained first. The convex hull of a set of points S is defined by O'Rourke [40] as the intersection of all of the sets of all convex combinations of points of S . In 2-D it is the convex polygon P that encloses S such that there is no other polygon P' that encloses S such that $P \supset P' \supseteq S$. It is the unique polygon that contains all of the points in the set while having the smallest area and always remaining convex. In 3-D this is extrapolated to the smallest convex polyhedron enclosing S . Similarly it can be thought of as the unique shape that encompasses all of the points in the set with the smallest volume while always remaining convex. The convex hull of a 2-D object and its convex hull are shown in Figure 6-2. It can be seen that the non-convex area is not included in the hull, and the entire boundary of the hull is convex.

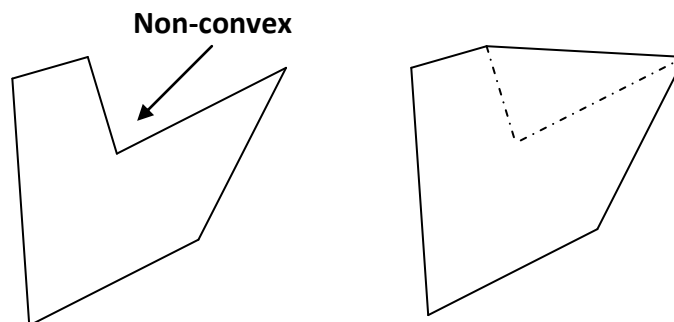


Figure 6-2 Left: a two dimensional shape.
Right: its convex hull.

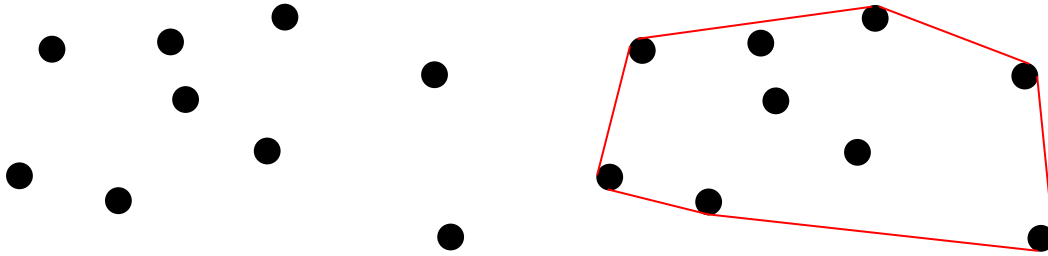


Figure 6-3 Left: a random set of points. Right: The red line represents the convex hull of the same set of points.

An analogy for a 2-D convex hull is a rubber band that is stretched over all points in a set and then allowed to contract. The rubber band encompasses all points in the set and minimizes the total area. In 3-D this can be thought of as a balloon that is stretched over a point cloud set and then allowed to contract.

This convex hull of a three dimensional point cloud set is a unique model of the object that eliminates the interior points. It simplifies the model of the object, diminishing the number of calculations needed for grasp planning.

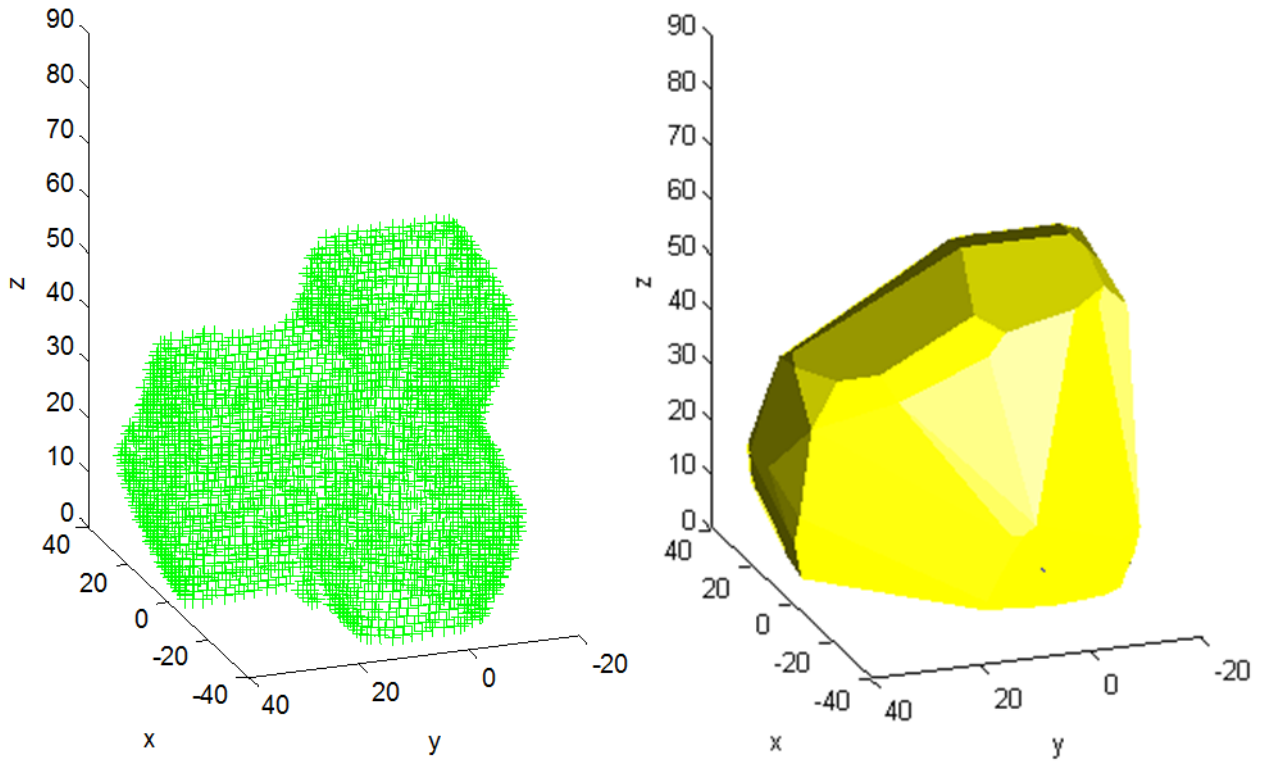


Figure 6-4 Left: A point cloud of an object containing 2704 points. Right: The convex hull of the same set of points with 68 vertices and 63 facets.

6.2.2 *Computation of Convex Hull*

As in the previous section, the 2-D case is explained first and then extrapolated to 3-D.

O'Rourke [40] presented two methods for obtaining a convex hull in 2-D and 3-D. The first method is "gift wrapping" developed by Chand and Kapur [41]. For a set of points $p \in S$, if the line from p_{i-1} to p_i is on the convex hull, then the next line that will lie on

the convex hull is the line beginning at point p_i to any other point p_j that has the smallest angle θ between the extension of p_{i-1} to p_i and p_i to p_j . This is then taken as the next line and the algorithm continues. The initial hull point can be taken as a boundary point, e.g. the lowest point. The analogy given by O'Rourke [40] is a shoelace that is wound around the set. An example is shown in Figure 6-5, where θ_1 is the smallest angle and p_{j_1} is the next point on the hull. In 3-D shown in Figure 6-6 if a facet is assumed to be on the convex hull. Then the plane formed by the three points (p_{i-1}, p_i, p_{i+1}) is rotated around one of the three edges (Figure 6-6 shows the first edge) until the plane intersects another point in the set in this case (p_j) the two points forming the line (p_i, p_{i+1}) and the new point (p_j) become a new facet on the convex hull if has not been previously added and adds two new edges to continue the process $(p_i$ to p_j and p_{i+1} to $p_j)$. The second edge of the original facet could then be analyzed again creating a new facet (see Figure 6-7). The main disadvantage of the "gift wrapping" algorithm is that is executed in $O(nh)$ time (worst case $O(n^2)$ time), where n is the number of points in the set, and h is the number of points on the hull.

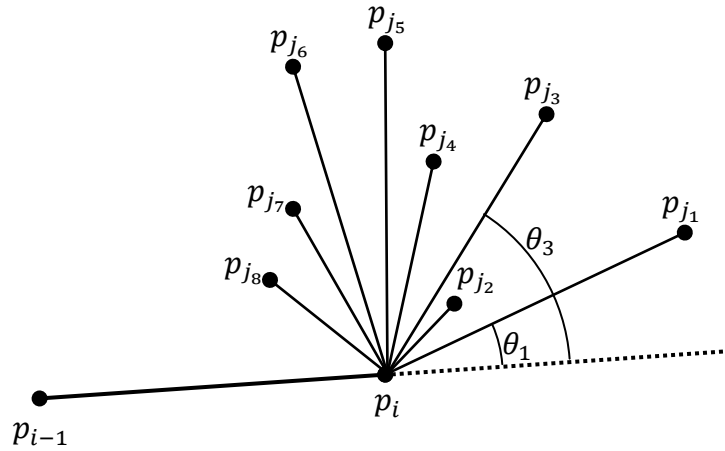


Figure 6-5 Gift wrapping of a convex hull in two dimensions.

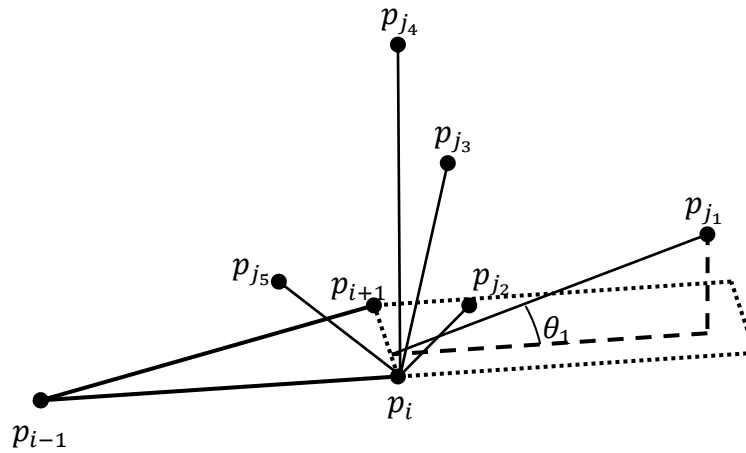


Figure 6-6 Gift wrapping of a convex hull in three dimensions first edge.

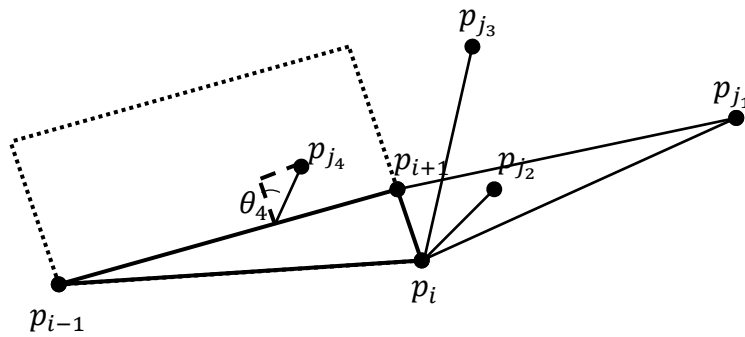


Figure 6-7 Gift wrapping of a convex hull in three dimensions second edge.

Quickhull, a technique developed by Barger *et al.* [42], was chosen for our implementation for its computational efficiency. It is able to compute the convex hull of a set in $O(n \log n)$ time.

6.2.3 *Triangular and arbitrary polygonal facets*

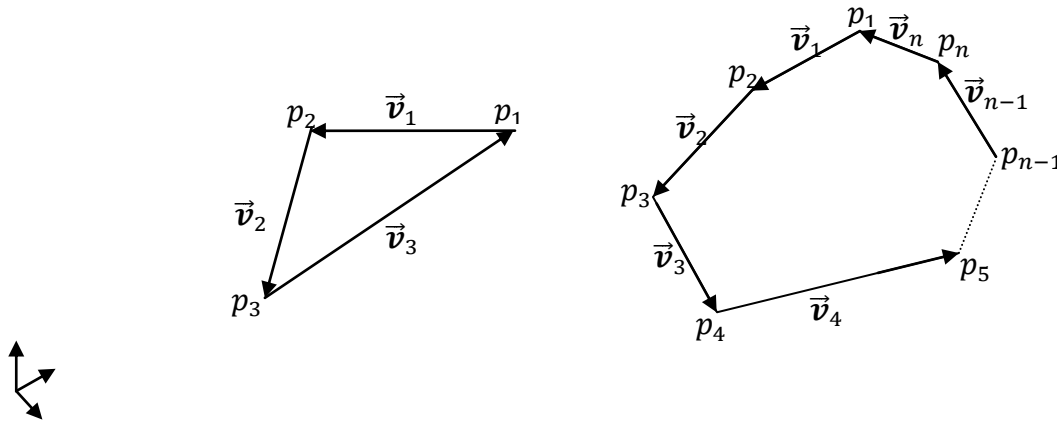


Figure 6-8 Left: a triangular facet. Right: an arbitrary polygonal facet.

Initially, in the grasp planner implementation the convex hull algorithm modeled the object using triangular facets. Later it was revised to use an arbitrary number of points to form a facet such that larger polygonal facets could be formed. Examples of the two types of facets are shown in Figure 6-8. To form a facet with more than three points while satisfying the convex criteria of a convex hull the additional point would have to be in plane with the other three. Due to sensor errors, the measured points are never perfectly accurate, a point on a flat face of the object would be offset and cause several

three point facets to be created. Rectangular object faces such as the side of a box or cube also had to be separated into at least two or more three point facets. To overcome this problem an allowance was made for a point to be up to 2mm off of a facet plane and still be included in that plane during the creation of the convex hull. This allowance helped to increase the size of the facets, by including the points only slightly off of them. This then reduced the number of facets in the final object model, decreasing the search space for the grasp planning algorithm.

6.2.4 *The **qconvex** function*

The **qconvex** function located in the Qhull[42] code library was used to implement the Quickhull algorithm. The qconvex function inputs were the point cloud generated by the sensors and the off plane allowance. The function generated the convex hull and its outputs consisted of the number of facets, the list of facets, the normal for each facet, and the facet area.

6.3 Grasp Planning Theory

The grasping strategy was designed to obtain a force closure grasp that is robust to external forces and moments; errors in the shape and pose of the object model; and errors in the gripper pose due to the robot. Force closure will be defined in the next subsection. The jaws of the gripper consisted of large parallel plates that are actuated along their common normal. The algorithm was designed such that one parallel plate would be in full contact with a convex hull facet and the other would contact a point

opposite to this facet which was termed $p_{opposite}$. The goals of the grasp planner are finding all pairs of facets and points that create a force closure grasp; and then selecting the best pair for grasp execution based on a quality metric.

6.3.1 **Closure**

In grasping theory, the two main types of object restraint are termed force closure and form closure. Form closure occurs when an object is geometrically constrained by a gripper without requiring friction [43]. Form closure grasps usually require a multi-fingered hand, and numerous contacts between the hand and object [44]. Using a parallel jaw gripper with flat jaws it is not possible to completely enclose the object to be grasped so form closure is not possible. The other type of restraint is force closure. Force closure provides object restraint using contacts with friction. The inclusion of friction reduces the number of contacts needed [44]. Force closure allows the object to resist arbitrary forces and torques assuming the squeeze force can be infinite. The interaction of the contacts between the object and the gripper fingers are typically modeled in one of three ways:

1. **Hard-finger contact:** Friction exists between the contact point and the object with a coefficient of friction μ . A force can be applied through a cone at an angle α (see Figure 6-9), where $\alpha = \text{atan } \mu$, without slip occurring between the contact point and the object.

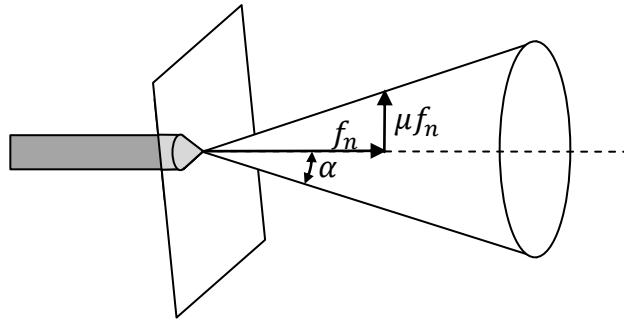


Figure 6-9 Friction cone for a hard finger contact with a normal force f_n , an angle α , and a coefficient of friction of μ .

2. **Soft-finger contact:** In addition to the hard-finger friction cone contact a soft finger contact can exert torques around the contact point. See Figure 6-10.

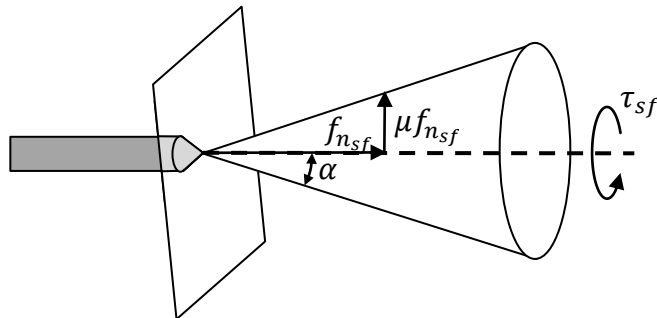


Figure 6-10 Friction cone for a soft finger contact with a force f_n , an angle α , a coefficient of friction of μ . A torque τ_{sf} around the contact point can be exerted.

- 3. Frictionless contact:** This is the same as a hard finger contact where the coefficient of friction $\mu = 0$. This means a frictionless contact is a hard contact that has a friction cone at an angle $\alpha = 0$. See Figure 6-11.

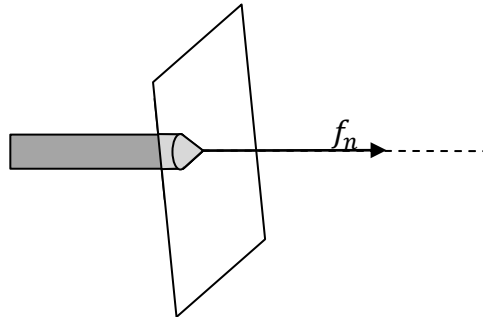


Figure 6-11 Frictionless contact for a force f_n , this is the same as a hard finger contact with a coefficient of friction $\mu = 0$ and therefore the friction cone has an angle α of 0 .

Most parallel jaw gripping papers rely on two antipodal points of contact, one on either side of the object. For this to be a force closure grasp two soft finger contacts are required. For an object held stationary in the air, the forces and torques acting on the object are shown in Figure 6-12 and Figure 6-13.

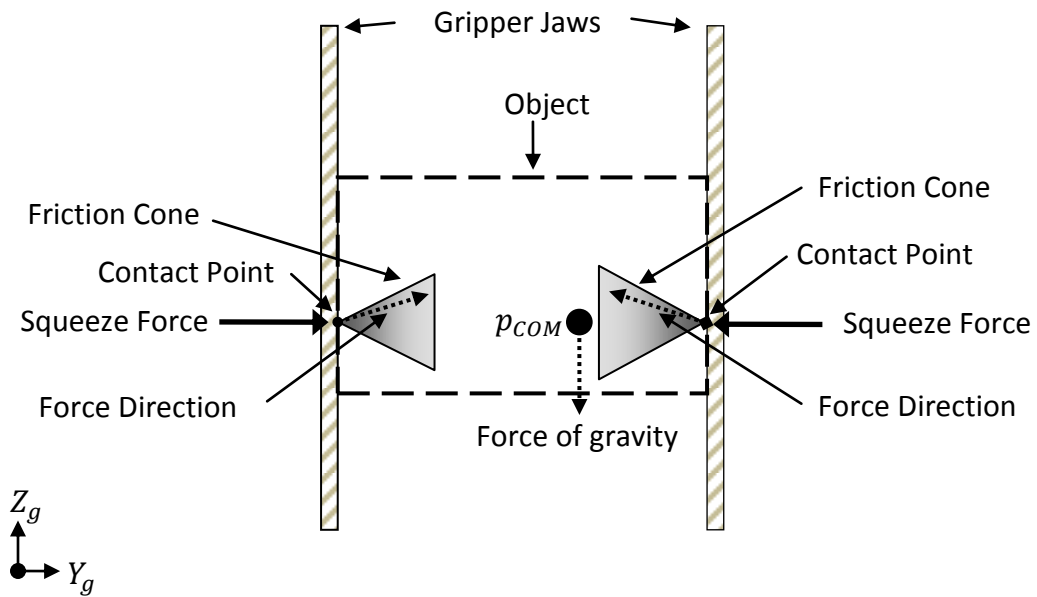


Figure 6-12 The forces acting on an object during a parallel jaw grasp, including gravity, shown from the front.

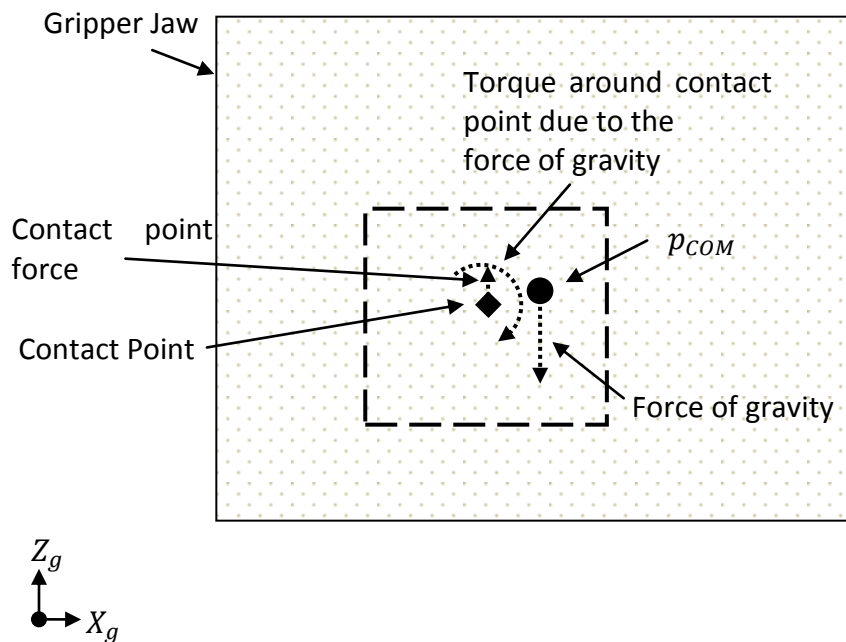


Figure 6-13 The forces and torques acting on an object during a parallel jaw grasp, including gravity, shown from the side.

With our grasping strategy, the selected facet on one side of the object provides a minimum of three points of contact with lines of action perpendicular to the facet plane, away from the gripper jaws. It cannot be assumed that the facet is a solid surface because any points internal to the three endpoints would be removed during the creation of the convex hull. Opposite from the selected facet is the point $p_{opposite}$. This point, as further discussed in 6.4.3, has a ray projecting from it in the opposite direction of the facet normal (N). The point where that ray intersects the plane that the facet lies on, is termed p_{facet} . For a successful grasp this point must lie internal to the selected facet. $p_{opposite}$ has its line of action pointing in the opposite direction of the facet normal, $(-N)$. Therefore, for each grasp there exists a minimum of three points of contact with the gripper jaw on the facet side with lines of action along N and at least one point of contact with the gripper jaw on the secondary or opposite side with line of action along $-N$.

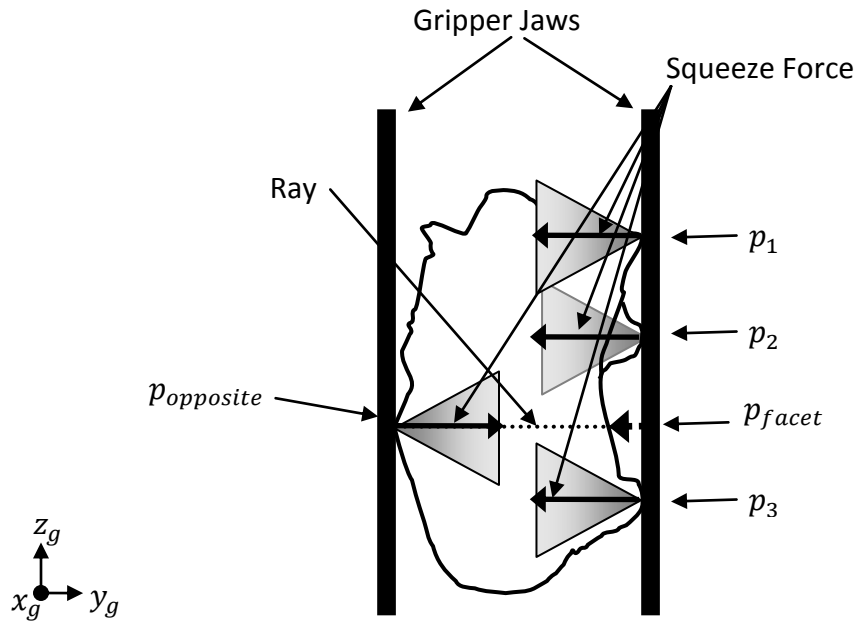


Figure 6-14 Front view of the grasping of an object with a minimum number of contact points.

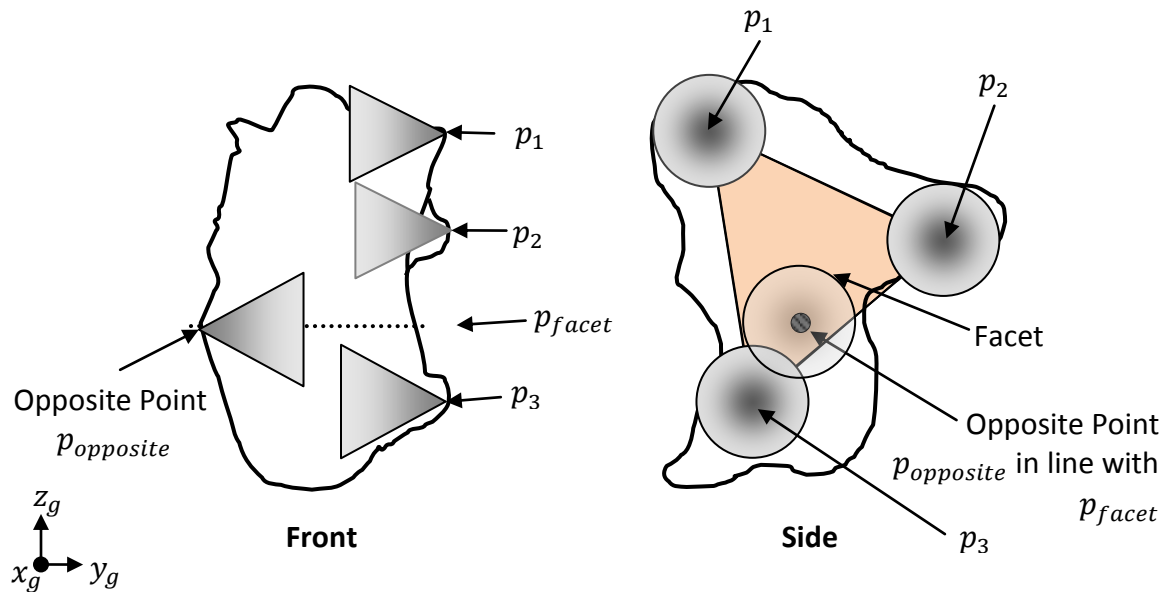


Figure 6-15 Left: a front view of the object being grasped with the contact points and friction cones. Right: a side view of the object being grasped with the contact points, selected facet and friction cones.

In the worst case of our grasping strategy there will only be four hard finger contacts. Figure 6-14 and Figure 6-15 show the object being grasped with four hard finger contacts. In Figure 6-15 the gripper fingers have been removed for clarity. The contacts due to the three point facet, p_1, p_2, p_3 can be reduced to a single contact interior to the facet p_{facet} , this point lies at the intersection of the dashed line in Figure 6-14 where it contacts the jaw. The problem is then reduced to the two soft-finger contact 3-D force closure grasp explained above. The main reason that soft fingers are required in a two point contact grasp is to resist the rotation of the object. Only a soft finger contact can exert torques around the contact point. Examining the three point facet in Figure 6-16 it can be seen that it is impossible to create a torque around a single contact point on the facet without creating forces that can be resisted by other two hard finger contacts. This is shown in Figure 6-16.

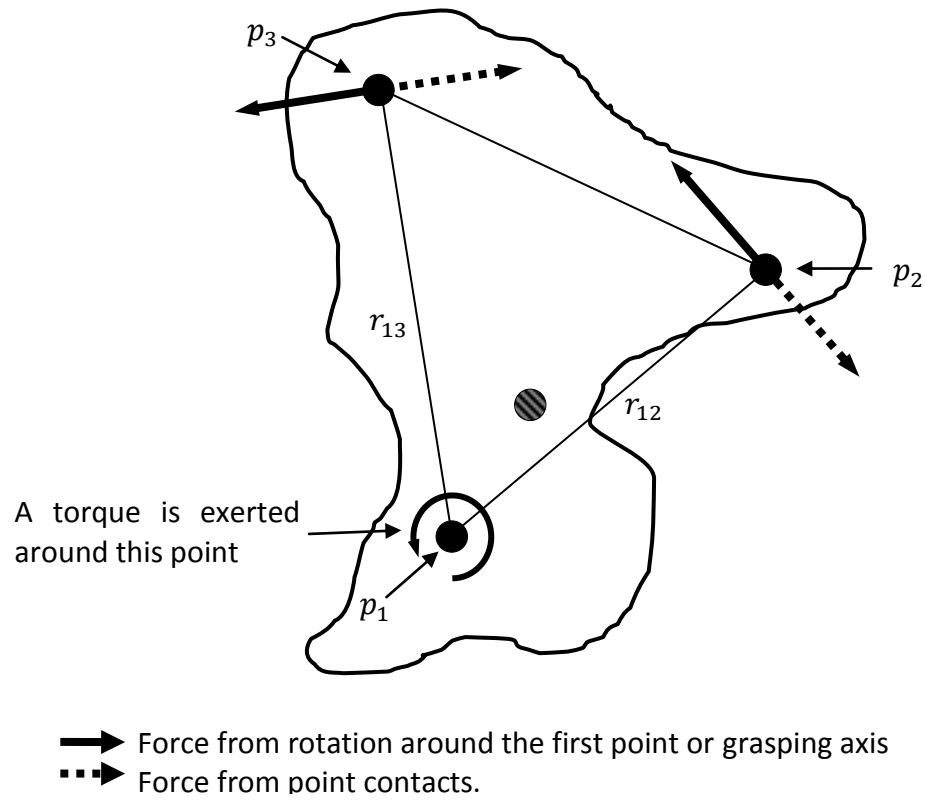


Figure 6-16 Demonstration of the torque around a point and the forces that act because of it.

The forces exerted at the second point, p_2 , and third point, p_3 , are given by:

$$F_j = \frac{\tau}{r_j \sin \theta_p} \quad (6-1)$$

Where

F_j is the force exerted at point p_j ($j = 2, 3$).

τ is the torque exerted at the first point (p_1)

r_j is the distance from the first point to point p_j ($j = 2, 3$).

θ_p is the angle between r_j and F_j .

If the forces are observed at right angles, $\theta_p = \left(\frac{\pi}{2}\right)$ then this equation simplifies to:

$$F_j = \frac{\tau}{r_j} \quad (6-2)$$

If a torque is created at any one point the other two points are able to resist it with a force. Because of this torque the constraint of a soft finger contact to stop a torque can be removed and a force closure grasp is accomplished with hard fingers only. If the point of contact from the simulated two finger grasp is inside the facet then three point contacts can be used to resist the force, this is shown in Figure 6-17 below.

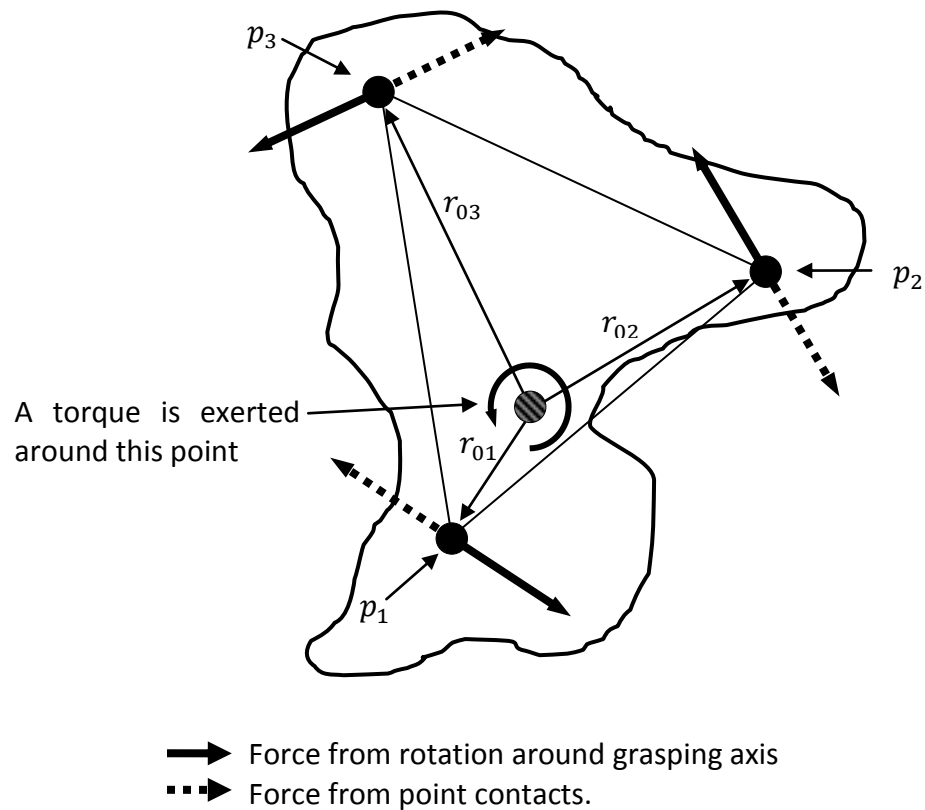


Figure 6-17 The view from the parallel jaw plane, with the finger contact points resisting a rotation along the grasping axis.

In this case again this axis of rotation is not in line with all the points on the grasping facet, so the three contacts on the facet (hard finger point contacts) are able resist the induced torque.

6.3.2 **Rotation**

To ensure that the object in the gripper did not rotate when the gripper jaws closed, the information from 6.3.1 is employed. In Figure 6-18 on the left, the opposite point ($p_{opposite}$) is chosen such, that is the point lies internal to the projection of the facet

across the object along the facet normal. When the gripper jaws close the force and torque from the opposite point ($p_{opposite}$) is balanced by the force and torque from the contact with the facet, and the object will not rotate. In Figure 6-18 on the right, the force and torque from $p_{opposite}$ will not be balanced by the force and torque from the contact with the facet (unless the coefficient of friction is very large), and as the jaws close the object may rotate counterclockwise, potentially creating an unstable grasp.

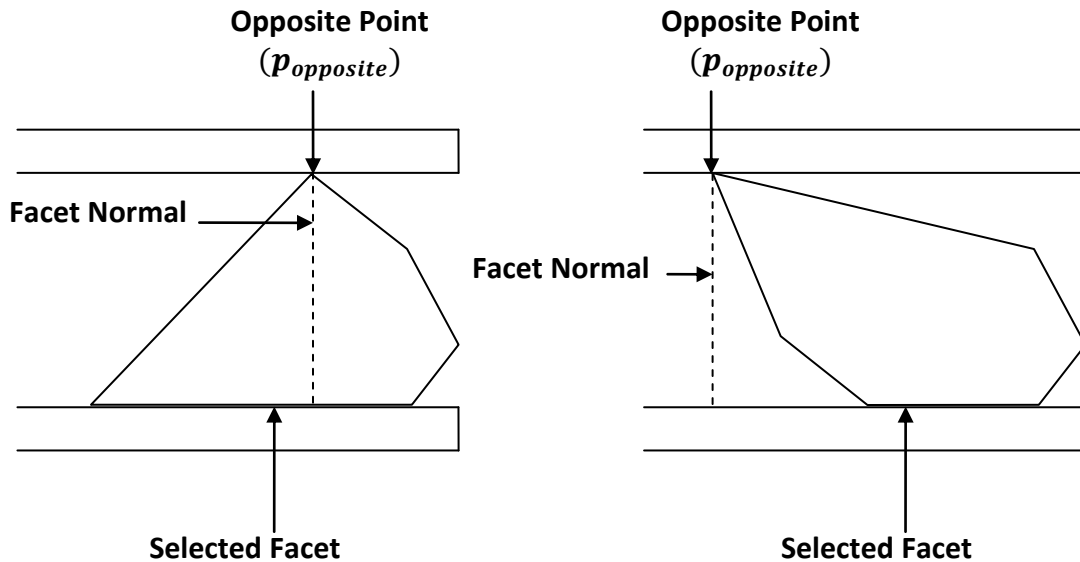


Figure 6-18 Left: the opposite point lies interior to the projected facet and the grasp is stable. Right: the opposite point lies exterior to the projected facet and the object rotates in the gripper.

6.3.3 ***Facet Area and COM***

For a more robust grasp the area between the gripper and the object should be maximized. This is the area between the gripper finger and the selected facet. A larger area allows for a grasp to be accomplished with more errors in physical modeling or with movement errors or offsets encountered during the grasp attempt. It also helps make the grasp more robust to external moments. For a grasp to be attempted a minimum area constraint must be met to rule out non-robust grasps, this also speeds up the grasp selection process by removing grasping candidates that are likely to fail earlier in the process by reducing the size of the grasping set. The grasp quality metric to be introduced in Section 6.4.3 will be heavily weighted on the facet area in order to favour larger facets. This can be done because the assumption that the gripper jaws have large flat faces ensures that the entirety of the selected facet lies in contact.

When an object is grasped and lifted above the table, a moment τ about the axis of rotation is caused by the force of gravity. The distance between the axis of rotation and the *COM* is defined as r . As the axis moves further from the p_{COM} the magnitude of the moment is increased. This is shown in Figure 6-19. In the best case when the grasping axis is through the p_{COM} ($r = 0 \rightarrow \tau = 0$) and at its worst case when r is half of the length of the object ($r = \frac{1}{2}l$), $\tau = \frac{1}{2}lf_g$.

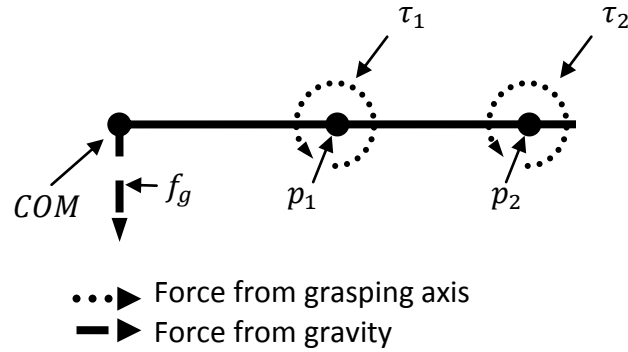


Figure 6-19 The force from the grasping axis and from the force of gravity.

If (6-1) is rearranged and the force of gravity is substituted the equation becomes

$$\tau = f_g r \quad (6-3)$$

Where r is the distance from the COM to the point (p_1, p_2, \dots, p_x) . From this equation it can be seen that since f_g is constant, as r increases τ increases.

6.4 Finding Potential Grasping Surfaces

To find the selected grasping facet and the opposite point $p_{opposite}$, the following naming convention was used to illustrate the steps of the grasping strategy:

$p_{opposite}$ the point opposite the chosen facet.

p_{facet} the point that lies on the facet that is projected along the facet normal from $p_{opposite}$.

$p_i, i = 1 \dots n$ the points (or vertices) that make up the facet, they are all in the same plane (or slightly off see 6.2.4).

c_{facet} the location of the centre of a facet.

These are illustrated in Figures 6-20, 6-21 and 6-22.

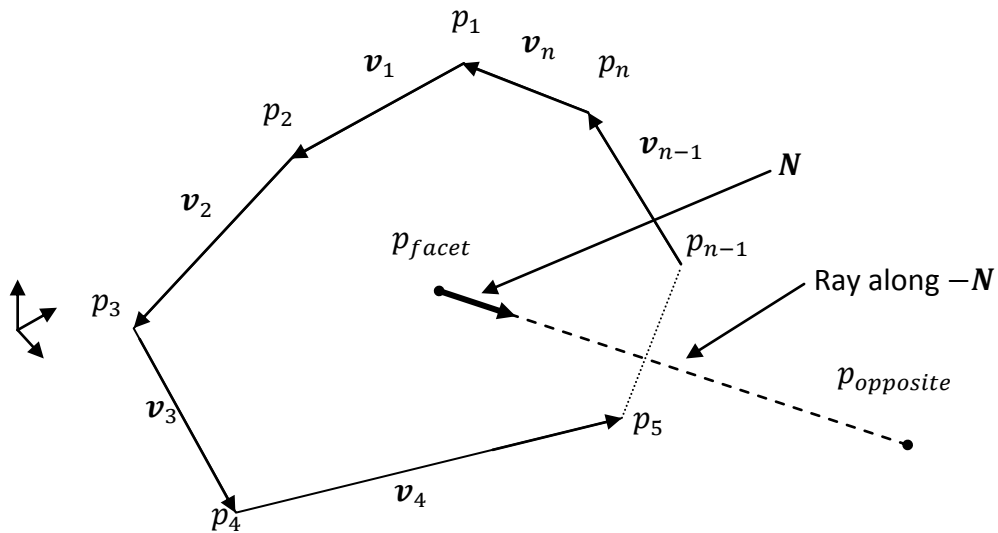


Figure 6-23 The n point facet, facet point and opposite point. The facet point is the intersection of a line projected from the opposite point along the facet normal and the plane the encompasses the facet

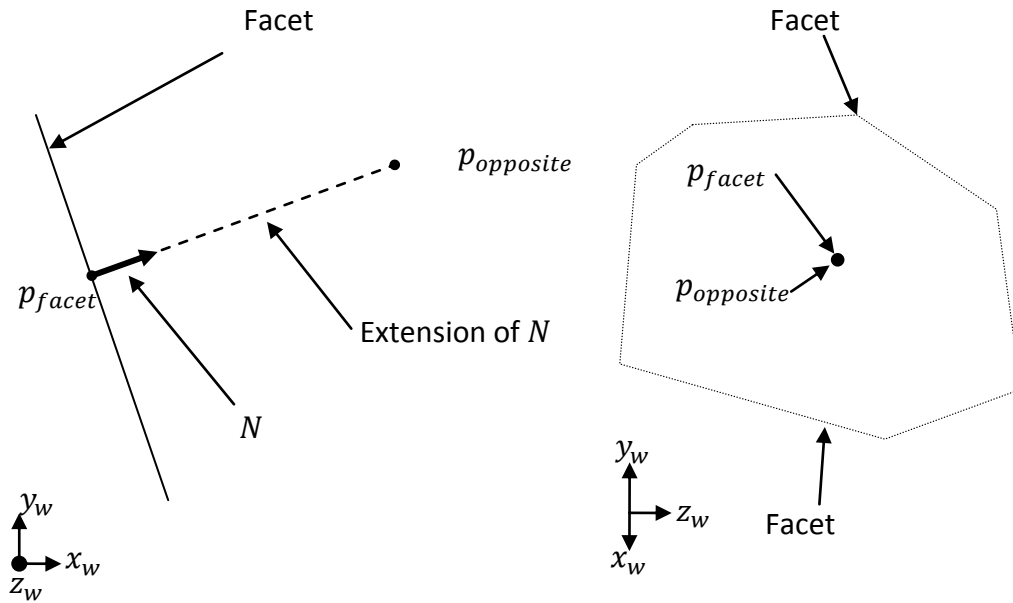


Figure 6-24 Left: an image of the n point facet from the side such that the facet extends into the page. The facet normal is at a right angle to the facet. It can be seen here where p_{facet} lies on the intersection of that facet normal projected from the opposite point. Right: The same facet rotated along the by $\frac{\pi}{2}$ about y_w and tilted such that the view is along the normal of the selected facet. Here p_{facet} and $p_{opposite}$ are in line with each other.

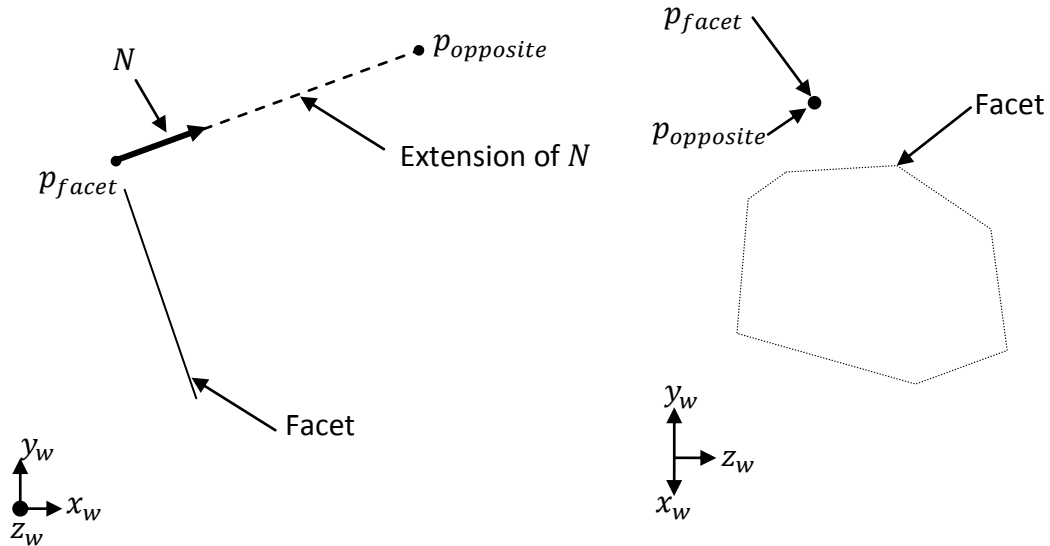


Figure 6-25 This is the same as Figure 6-24 with the exception that here p_{facet} appears outside the facet while projected back along the normal.

6.4.1 *Grasp Planner inputs and outputs*

The inputs to the grasping planner are:

- The point cloud model of the object
- The convex hull of the object including
 - A set $(1 \dots k)$ of candidate grasp facets (CGFs) to be evaluated, each having:
 - $n \geq 3$ facet vertices in the world frame (x_w, y_w, z_w)
 - Facet Normal $N = (n_x, n_y, n_z)$ from $(Ax + By + Cz = D)$
 - Facet Area A_{f_l}

- Maximum distance from the facet to any point in the cloud along the normal of the facet $dist_{p_{opposite}, p_{facet_{max}}}$
- The centre coordinate of the facet (x_{facet}, y_{facet})
 - The centre of mass (p_{COM}) of the convex hull of the object assuming uniform density.

The outputs of the grasp planner for each grasp before for evaluation are:

- The number of points in the CGF
- The facet endpoint coordinates in the world frame (x_w, y_w, z_w)
- The coordinates of the opposite point($p_{opposite}$) in the world frame (x_w, y_w, z_w)
- The coordinates of the point on the facet plane (p_{facet}) in the world frame (x_w, y_w, z_w)
- The Normal to the grasp ($A_g x + B_g y + C_g z = D_g$)
- The facet Area A_{fg}
- The maximum distance from the facet to any point on the cloud along the normal of the facet
- The minimum distance from the line connecting p_{facet} and $p_{opposite}$, and the p_{COM}
- The distance from the facet point(p_{facet}) to the centre of the facet (c_{facet})

- The location of the centre of the grasp in the object, this is the location that is centred between p_{facet} and $p_{opposite}$

6.4.2 ***Pre grasp-planner steps***

Before the grasp planner can compute actual grasps the following information must be determined and stored.

The set of facets and their endpoint coordinates in the world frame (x_w, y_w, z_w) are calculated using Qhull and stored in a structured array of facets. This array also includes the equation of the facet plane in the form $(Ax + By + Cz = D)$, and the facet area. p_{COM} of the object is calculated based on the convex hull of the point cloud and assumes a uniform density. It is computed as follows:

For a convex hull with k facets, each consisting of a convex polygon with n sides and j corners.

1. For each facet $l = 1 \dots k$ on the convex hull subdivide the n -sided polygon into $n-2$ triangles from tr_1 to tr_{n-2} .

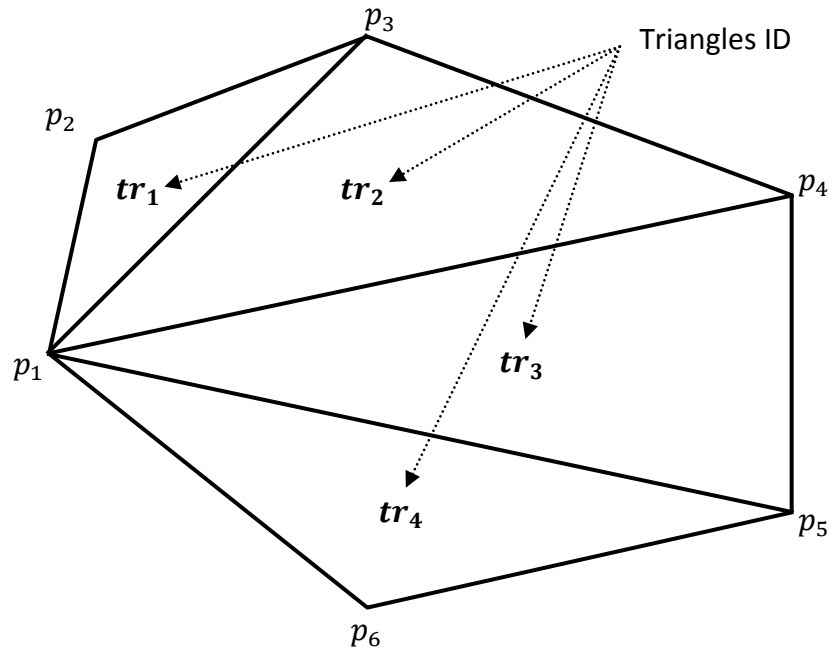


Figure 6-26 an example of a 6 sided polygon ($n = 6$) divided into 4 separate triangles $(6 - 2) = 4$.

2. For each triangle tr_i consisting of points $(p_1, p_{tr_{i+1}}, p_{tr_{i+2}})$

Compute the area and centroid.

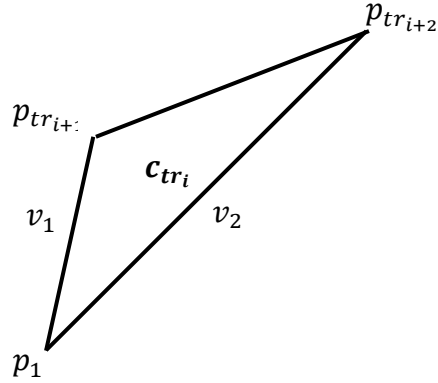


Figure 6-27 calculation of area(A_i) and centroid(c_{tri}) of a triangle t_{ri}

The area is calculated using the cross product of the vectors

$$v_1 = (p_{tri+1x} - p_{1x}, p_{tri+1y} - p_{1y}, p_{tri+1z} - p_{1z}) \quad (6-4)$$

$$v_2 = (p_{tri+2x} - p_{1x}, p_{tri+2y} - p_{1y}, p_{tri+2z} - p_{1z}) \quad (6-5)$$

$$A_{tri} = \frac{1}{2} |v_1 \times v_2| \quad (6-6)$$

$$c_{tri} = \frac{1}{3} \begin{bmatrix} p_{1x} + p_{tri+1x} + p_{tri+2x} \\ p_{1y} + p_{tri+1y} + p_{tri+2y} \\ p_{1z} + p_{tri+1z} + p_{tri+2z} \end{bmatrix} \quad (6-7)$$

where

$$c_{tri} = (c_{tri_x}, c_{tri_y}, c_{tri_z}) \quad (6-8)$$

3. Compute the area and centroid of the facet.

The area of the facet $(A_{f_l}, l = 1 \dots k)$ is the sum of the areas of the smaller triangles

$$A_{f_l} = \sum_{i=1}^{i=n-2} A_{tr_i} \quad (6-9)$$

The centroid of the facet c_{f_l} can be computed from the above as a function of the areas and centroids of all the triangles in the facet.

$$c_{f_l} = \frac{\sum_{i=1}^{i=n-2} A_{tr_i} (c_{tr_{ix}}, c_{tr_{iy}}, c_{tr_{iz}})}{A_{f_l}} \quad (6-10)$$

where c_{f_l} can be expressed as

$$c_{f_l} = (c_{f_{lx}}, c_{f_{ly}}, c_{f_{lz}}) \quad (6-11)$$

4. Calculate the total area and p_{COM} using all of the facets.

The total area of the object is

$$A_{ob} = \sum_{l=1}^{l=k} A_{f_l} \quad (6-12)$$

And p_{COM} is calculated using the centroid and areas of each facet.

$$p_{COM} = \frac{\sum_{l=1}^{l=k} A_{f_l} c_{f_l}}{A_{ob}} \quad (6-13)$$

or

$$p_{COM} = \frac{\sum_{l=1}^{l=k} A_{f_l} (c_{f_{lx}}, c_{f_{ly}}, c_{f_{lz}})}{A_{ob}} \quad (6-14)$$

where p_{COM} can be expressed as

$$p_{COM} = (p_{com_x}, p_{com_y}, p_{com_z}) \quad (6-15)$$

The maximum distance from the facet to any point in the point cloud was calculated using the following method:

1. For each CGF $i = 1 \dots k$
 - a. For each point in the point cloud $j = 1 \dots h$

Calculate the distance d from the facet to the point:

Using the point $p(x, y, z)$ and equation of the facet plane:

$$Ax + By + Cz = D \quad (6-16)$$

The distance is the projection of a vector from the plane to the point(w) onto the normal vector of the plane(v).

$$d = \frac{|v \cdot w|}{|v|} \quad (6-17)$$

Substituting in v and w :

$$d = \frac{|A_i x_j + B_i y_j + C_i z_j + D_i|}{\sqrt{A_i^2 + B_i^2 + C_i^2}} \quad (6-18)$$

The largest value is stored for each facet and is used by the grasp planner while determining if $p_{opposite}$ is viable.

6.4.3 ***Grasp planner***

The grasp planner takes the information from the pre-grasp planner and creates grasps using a set of grasping constraints that must be met. It then evaluates a set of criteria and computes a grasp quality metric. Lastly, the grasp with the highest quality is chosen.

To evaluate each grasp the grasp planner iterates through CGF's ($i = 1 \dots k$). Each facet has $n \geq 3$ vertices that make up the plane the facet lies in, as was shown in Figure 6-23. Each point in the model that is not in the CGF is then evaluated as $p_{opposite}$. Using the ray projecting method described earlier this $p_{opposite}$ is projected to the CGF. The intersection of the line and the plane is p_{facet} . The points and projections can be seen in Figure 6-24 and Figure 6-25. For the grasp to be stable p_{facet} must lie interior to the facet edges on the plane.

Create a set of grasps S that, along with the pre-grasp information, contain the following:

1. For each CGF, $i = 1 \dots k$
 - a. For each point in the object not making up the current facet, $j = 1 \dots h$
 - i. Find p_{facet} as shown in Figure 6-24 and Figure 6-25

Using the normal of the facet given as

$$Ax + By + Cz + D = 0 \quad (6-19)$$

and rearranging it into the form

$$(A, B, C) \cdot (x, y, z) = -D \quad (6-20)$$

substitute a point on the plane p_{facet} and the normal of the plane N

$$p_{facet} \cdot N = -D \quad (6-21)$$

with the equation of a line

$$p = p_0 + lt \quad (6-22)$$

with

p, p_0 points on the line

l a direction vector of the line

Then using (6-22) and substituting N for L , and $p_{opposite}$ for p_0

$$p = p_{opposite} + Nt \quad (6-23)$$

the equation of a plane can be represented as

$$(p_p - p_{p_0}) \cdot N = 0 \quad (6-24)$$

where

p_p, p_{p_0} are points on the plane

N is the normal of the plane

Combining (6-24) with (6-23), where $p = p_p$, the line and plane intersect.

$$(p_{opposite} + Nt - p_{p_0}) \cdot N = 0 \quad (6-25)$$

solving for t

$$t = -\frac{(p_{opposite} \cdot N - p_{p_0} \cdot N)}{N \cdot N} \quad (6-26)$$

substituting p_{facet} into p_{p_0} (a point on the plane)

$$t = -\frac{(p_{opposite} \cdot N - p_{facet} \cdot N)}{N \cdot N} \quad (6-27)$$

using equation (6-21) and subbing into (6-27)

$$t^* = -\frac{(p_{opposite} \cdot N + D)}{N \cdot N} \quad (6-28)$$

knowing that the intercept of the plane and the line is at t^* we can

find p_{facet} as

$$p_{facet} = p_{opposite} + Nt^* \quad (6-29)$$

substituting in equation (6-23)

$$p_{facet} = p_{opposite} - N \left(\frac{(p_{opposite} \cdot N + D)}{N \cdot N} \right) \quad (6-30)$$

ii. Find the distance from $p_{opposite}$ to p_{facet} using

(6-31)

$$dist_{p_{opposite}, p_{facet}} = \sqrt{(p_{opposite_x} - p_{facet_x}, p_{opposite_y} - p_{facet_y}, p_{opposite_z} - p_{facet_z})}$$

This is the distance between the grasping points and is the distance the gripper jaws must open; it cannot be larger than the max gripper separation (70mm). The largest value of $dist_{p_{opposite}, p_{facet}}$ termed $dist_{p_{opposite}, p_{facet_{max}}}$ must also be stored in the gripping structure to ensure that the grasp will contact $p_{opposite}$ before another point on the object see Section 6.4.4.

- iii. Find out whether p_{facet} lies interior or exterior to the facet edges on the plane encompassing the facet see Figure 6-28. For a successful grasp p_{facet} must lie interior.

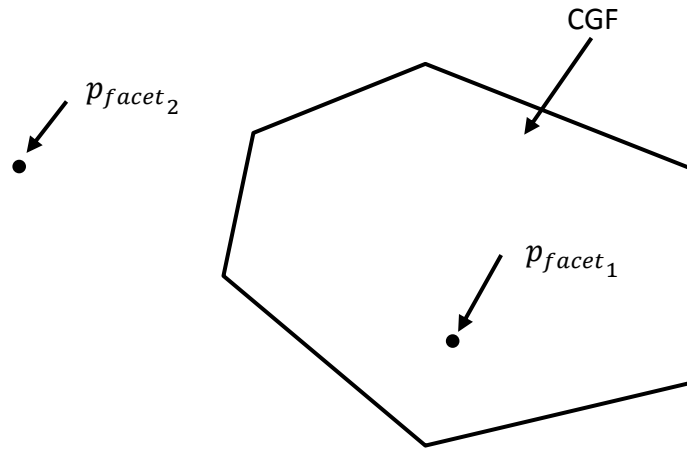


Figure 6-28 p_{facet_1} lies inside the facet, while p_{facet_2} lies outside the facet.

Using a simple case of a point and a three point facet the process for finding whether a point is interior or exterior is the following. In the case of the triangle in Figure 6-28, and Figure 6-29 with points(p_1, p_2, p_3) , edge vectors (v_1, v_2, v_3) and two test points with vectors v_{f_1} for p_{facet_1} and v_{f_2} for p_{facet_2} .

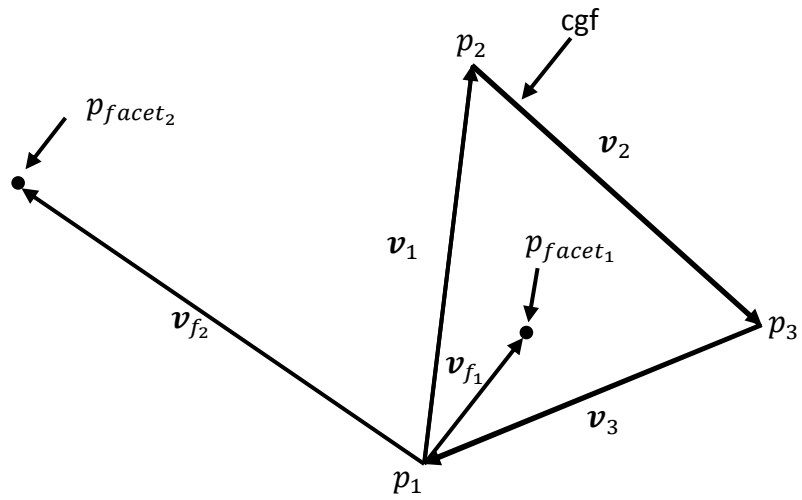


Figure 6-29 Calculating whether the point lies interior or exterior.

It can be seen that the point will lie in the interior of the facet if it lies on the side of each the edge vectors that are towards the centre of the facet. If the cross product of v_1 and v_{f1} , $(v_1 \times v_{f1})$, and v_1 and v_{f2} , $(v_1 \times v_{f2})$, are examined it can be seen that they will result in different signs as the vectors are on opposite sides of the edge v_1 . Using the right hand rule $v_1 \times v_{f1}$ will be negative, and $v_1 \times v_{f2}$ will be positive.

To figure out what sign is needed for a point to lie on the correct side of an edge any other point on the facet can be used, here it is p_3 . This is valid because the shape of the facet must be convex as it is based off of the convex hull hence no other point can be either in line with

the current edge vector or on the side of the vector that an exterior point would lie.

For an n sided convex planer polygon this is accomplished using the following algorithm:

- 1) For each of the edges of the polygon $x = 1 \dots n - 1$ shown in

Figure 6-23

Construct the vector v_x

$$v_{x_1} = p_{x+1} - p_x \quad (6-32)$$

Construct a vector from p_x to any other point on the CGF, p_{x+2} was used

$$v_{x_2} = p_{x+2} - p_x \quad (6-33)$$

Construct the vector v_f shown in Figure 6-29 from p_x to p_{facet}

$$v_f = p_{facet} - p_x \quad (6-34)$$

find the cross product of both

$$a = v_{x_1} \times v_{x_2} \quad (6-35)$$

$$b = v_{x_1} \times v_f \quad (6-36)$$

examine the dot product of the cross products.

$$result = a \cdot b \quad (6-37)$$

$$result = ((p_{x+1} - p_x) \times (p_{x+2} - p_x)) \cdot ((p_{x+1} - p_x) \times (p_{facet} - p_x)) \quad (6-38)$$

If *result* is positive then the point being examined (p_{facet}) lies on the interior side of the vector v_{x_1} the next $n - x$ edges are then examined if *result* is less than or equal to 0 then p_{facet} is outside the facet and the grasp is discarded.

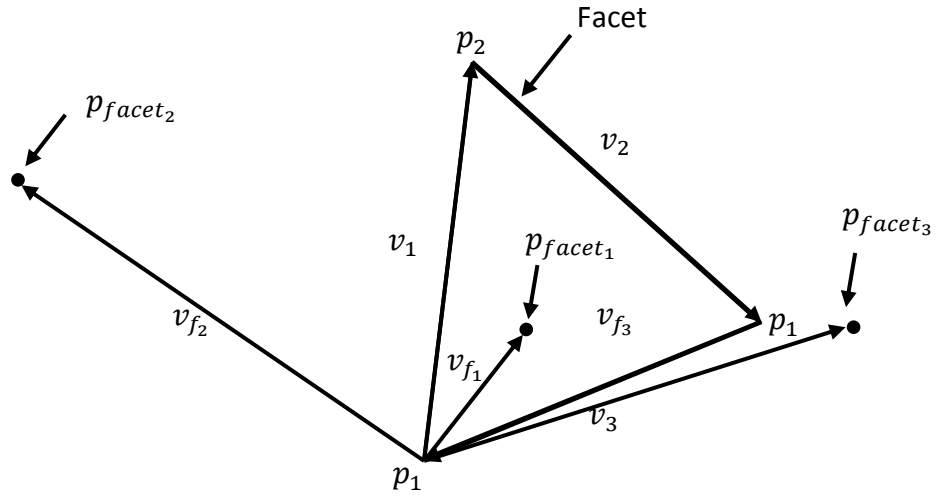


Figure 6-30 Interior/exterior point example.

In Figure 6-30 p_{face1} is interior to all edge vectors(v_1, v_2, v_3), p_{facet} is not interior to edge vector v_1 but is to v_2 , and v_3 , p_{facet3} is only interior to v_1 .

- iv. Find the distance between p_{COM} and the line joining p_{facet} and $p_{opposite}$ (d_{com}).

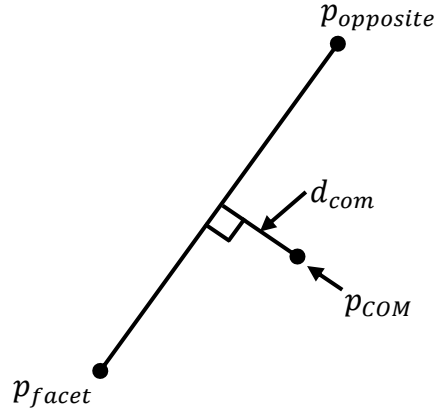


Figure 6-31 Distance between the line joining p_{facet} and $p_{opposite}$ and the p_{COM} .

The area of the triangle ($p_{facet}, p_{COM}, p_{opposite}$) can be represented as both

$$Area = \frac{1}{2} |(p_{COM} - p_{facet}) \times (p_{COM} - p_{opposite})| \text{ and} \quad (6-39)$$

$$Area = \frac{1}{2} |p_{opposite} - p_{facet}| d_{com} \quad (6-40)$$

where d_{com} is the height of the triangle, which is also the minimum distance between the line segment ($p_{opposite} - p_{facet}$) and p_{COM} .

Equating (6-39) and (6-40), rearrange and solve for d_{com} using

$$d_{com} = \frac{(p_{COM} - p_{facet}) \times (p_{COM} - p_{opposite})}{|p_{opposite} - p_{facet}|} \quad (6-41)$$

- v. Calculate the distance from the centre of the facet (c_{fi}) to p_{facet} using the distance formula

$$dist_{c_{fi}p_{facet}} = \sqrt{(c_{fi_x} - p_{facet_x})^2 + (c_{fi_y} - p_{facet_y})^2 + (c_{fi_z} - p_{facet_z})^2} \quad (6-42)$$

- vi. Compute the grasp quality metric according to

$$grip_{val} = Weight_{area} \left(\frac{A_{fi}}{A_{ob}} \right) + Weight_{DCM} \left(1 - \frac{d_{com_i}}{d_{com_{max}}} \right) + Weight_{dist_{c_{fi}p_{facet}}} \left(1 - \frac{dist_{c_{fi}p_{facet}}}{dist_{c_{fi}p_{facet_{max}}}} \right) \quad (6-43)$$

where

$grip_{val}$ is the grasp quality metric,

$Weight_{area}$ is a user defined weight on the area of the facet,

$\left(\frac{A_{fi}}{A_{ob}} \right)$ is the dimensionless ratio of the area of the chosen facet over

the surface area of the convex hull of the object (Note that

$0 < \frac{A_{fi}}{A_{ob}} < 1/2$).

$Weight_{DCM}$ is a user defined weight on the distance from the centre of mass,

$1 - \frac{d_{com_i}}{d_{com_{max}}}$ is a dimensionless number, with a range $0 <$

$\left(1 - \frac{d_{com_i}}{d_{com_{max}}}\right) \leq 1$, representing the distance from the line joining

p_{facet} and $p_{opposite}$ and the COM (Note that a closer line gives a higher number, *i.e.* a line through the COM would result in

$1 - \frac{0}{d_{com_{max}}} = 1 - 0 = 1$ while a line that is the maximum distance

away would produce $1 - \frac{d_{com_{max}}}{d_{com_{max}}} = 1 - 1 = 0$),

$Weight_{dist_{cf_i}p_{facet}}$ is a user defined weight on the distance to the centre of the facet, and

$\left(1 - \frac{dist_{cf_i}p_{facet}}{dist_{cf_i}p_{facet_{max}}}\right)$ is a dimensionless number, with a range

$0 < \left(1 - \frac{dist_{cf_i}p_{facet}}{dist_{cf_i}p_{facet_{max}}}\right) \leq 1$, representing the distance from p_{facet}

to $dist_{cf_i}$ where 0 means that that p_{facet} is at its maximum distance away from cf_i .

6. Calculate the centre point of the two gripper finger tips $tc = (x_{tc}, y_{tc}, z_{tc})$. To calculate the z component for the CGF tip centre (fac_{tc}) the lowest point on the CGF is chosen, this is ($z_{fa_{tc}}$). The x_w and y_w components of the vertices of the CGF are examined and the maximum (x_{max}, y_{max}) and minimum (x_{min}, y_{min}) coordinates in are found. The point centred between the two points is used for the facet tip

centre x , and y components of fac_{tc} . This is shown in Figure 6-32 and is given by the following equations:

$$z_{fac_{tc}} = z_{min}, \quad (6-44)$$

$$x_{fac_{tc}} = \frac{x_{min} + x_{max}}{2}, \quad (6-45)$$

$$y_{fac_{tc}} = \frac{y_{min} + y_{max}}{2} \quad (6-46)$$

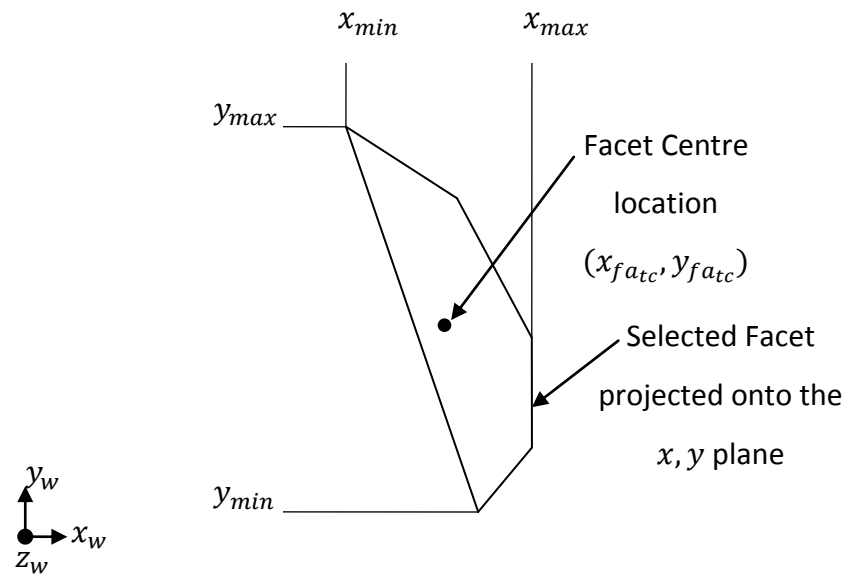


Figure 6-32 By projecting the facet onto the z plane the largest and smallest values for the x and y values are found

To get to the tip centre tc from fac_{tc} the point is projected along the normal half the distance from the facet to the maximum opposite point. As shown in Figure 6-33, this puts the tip centre in the middle of the

object and allows the gripper jaws to be equally spaced around the object when a grasp is attempted.

$$tc = fa_{tc} + \frac{1}{2} dist_{p_{opposite}, p_{facet_{max}}} \vec{N} \quad (6-47)$$

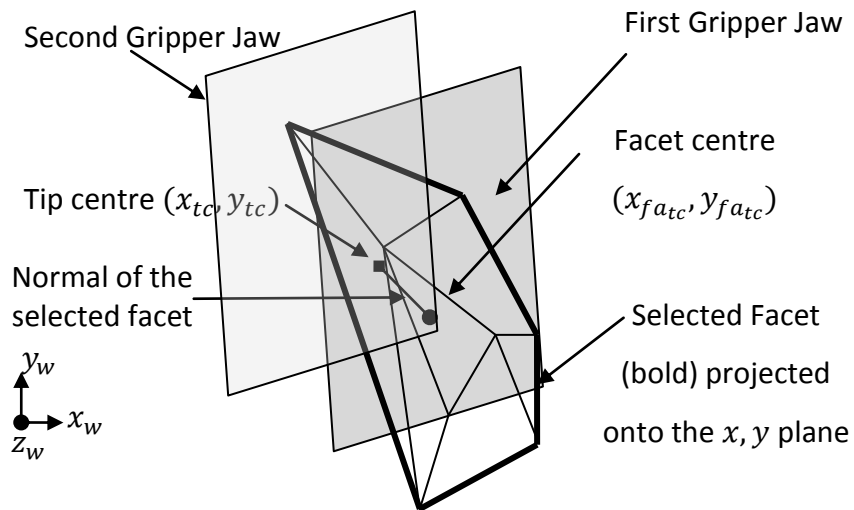


Figure 6-33 Procedure to find the tip center

6.4.4 ***Reduction of Grasp Set***

Using the information stored in the set S created as described in 6.4.2 and 6.4.3, grasps that cannot be physically realised, or are likely to fail, are eliminated using the following procedure:

1. The gripper jaws have a maximum distance that they can open (max gripper separation), and a minimum distance that they can close (min gripper separation).
 - a. If the min gripper separation is greater than $dist_{p_{opposite}, p_{facet}}$ the gripper cannot grasp the object as the object is too small and will not contact the gripper finger surfaces on both sides.
 - b. If the max gripper separation is smaller than $dist_{p_{opposite}, p_{facet}}$ then the gripper cannot grasp the object because the jaws are not able to open wide enough to allow the object to fit inside.
 - c. If the max gripper separation is smaller than $dist_{p_{opposite}, p_{facet_{max}}}$ minus a small threshold then the gripper cannot grasp the object because it will contact the object as it moves in to grasp it. This is shown in Figure 6-34.

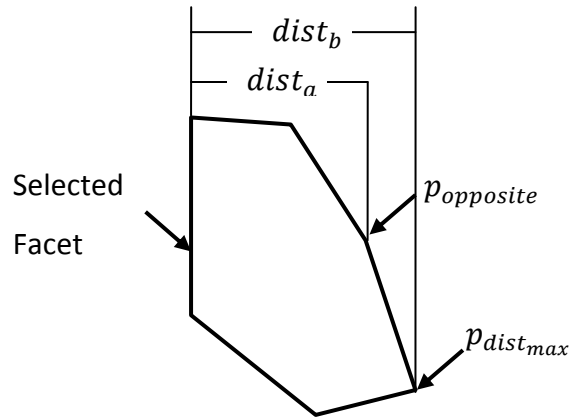


Figure 6-34 Grasping size constraints. $p_{opposite}$ is $dist_a$ away from the selected facet, the point $p_{dist_{max}}$ is $dist_b$ away from the selected facet. If $dist_b - dist_a$ is greater than the threshold then the grasp is removed.

In the above figure the selected point $p_{opposite}$ is $dist_a$ away from the selected facet, the point $p_{dist_{max}}$ is $dist_b$ away from the selected facet. If a grasp is selected between the facet and the $p_{opposite}$ the difference $dist_b - dist_a$ is used to determine whether a grasp attempt could be considered. If $dist_b - dist_a$ is large then as the parallel jaws close the point of contact between the jaws and the object will occur at $p_{dist_{max}}$. In Figure 6-34 this would result in a rotation in the clockwise direction and could lead to an unstable grasp. An unstable grasp will not always be created if $dist_b - dist_a$ is large. When $p_{opposite}$ is examined and found to be too far away from $p_{dist_{max}}$ $p_{opposite}$ is removed from set. If $dist_b - dist_a$ is small, then $p_{opposite}$ and $p_{dist_{max}}$ lie close to one another, as can occur if the points

are approximately on a plane and the plane is not exactly parallel to the selected facet. In this case it does not make sense to remove the point because it will not result in a significant rotation of the object. Both of these cases are shown in Figure 6-35.

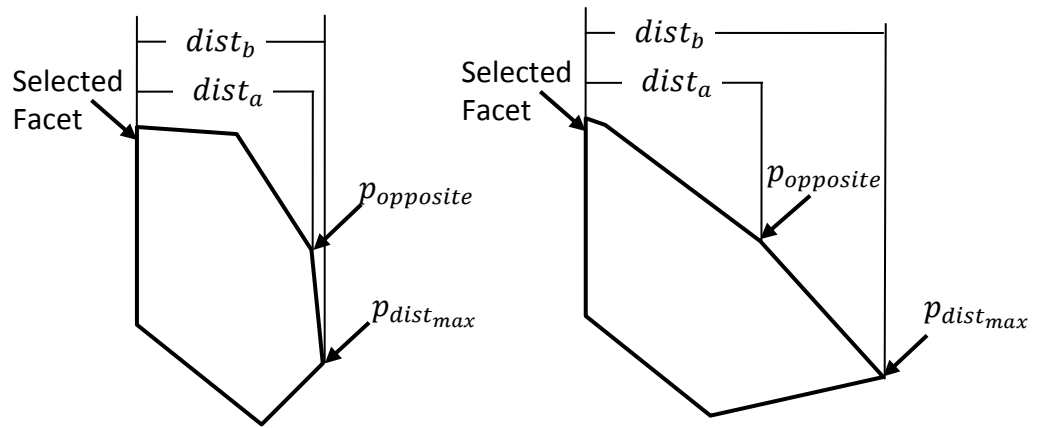


Figure 6-35 Left $dist_b - dist_a$ is small resulting in almost no rotation, Right $dist_b - dist_a$ is large resulting in a large rotation.

2. To grasp the object a minimum area is needed to be in contact with the selected facet. If the area is too small then the jaws may not be able to successfully grasp and lift the object. Also with a smaller area the locations of valid points on the opposite side (choices for $p_{opposite}$) are reduced due to the interior point constraint see Section 6.4.2. The area threshold was set at $250mm^2$ in our implementation.
3. The angle that the gripper can grasp an object has an upper and lower bound based on the physical limitations of the robot. In our implementation these

are set at ± 20 degrees from the z plane as shown in Figure 6-36. The angle threshold is checked using the dot product of the normal and the z axis (z_w) of the world coordinate given by

$$\theta = \arccos\left(\frac{z_w \cdot N}{\|z_w\| \|N\|}\right) \quad (6-48)$$

This simplifies to

$$\theta = \arccos(n_z) \quad (6-49)$$

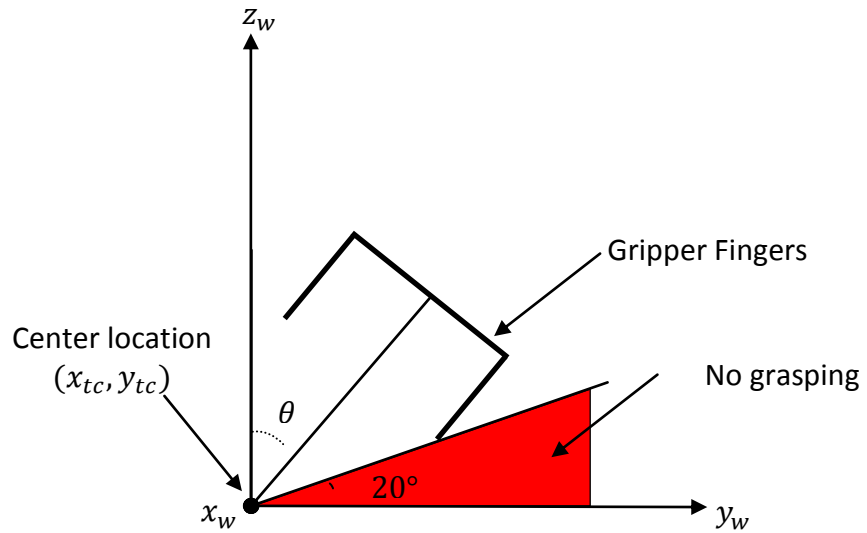


Figure 6-36 Visualization of the 20° no grasping zone

4. When the object is mounted on the playfield it is raised above the base height of the table. This height creates an error in the camera modeling process where the sides of the object, that may not exist, are not eliminated correctly. A large facet is created with an angle from the z_w axis similar to the

angle the camera is at during image capture (θ_{im} from 0). An example is shown in Figure 6-37. This error is more apparent as an object becomes shorter (smaller in the z_w axis) because the ratio of object size to distance from the table is greater. The facets that are created are based on the height of the object off of the table. Due to this model imaging error these facets can become larger than the facets created from the actual object and are thus erroneously chosen as the preferred grasp. To combat this problem, grasping angles that lie along or very close to this axis are removed from the grasping set because the probability that they are erroneous is high. This is shown in Figure 6-38, the angle α is the angle from the z_w axis to the beginning of the region when the grasp is removed, β is the degree threshold for removal of grasps. A grasp with angle γ is removed from the set if $\alpha < \gamma < \alpha + \beta$. These constraints occur along both the positive and negative angles from the z_w axis. A combination of the two constraints is shown in Figure 6-39. In 3-D the constraints can be regarded as two separate hollow or thin walled cones.

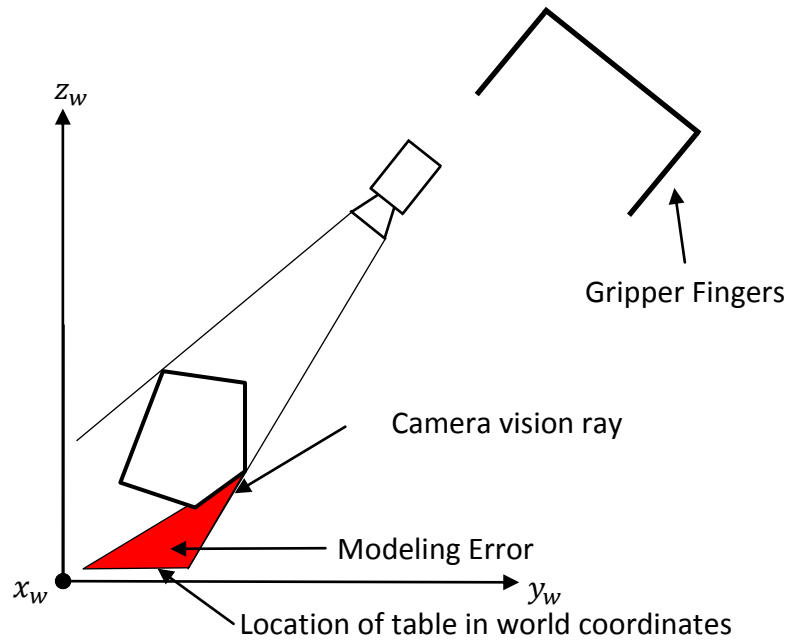


Figure 6-37 Image facet error due to height. The red area is the error the model will have due to the height of the camera over the object, the height of the object above the table, and the angle of the camera.

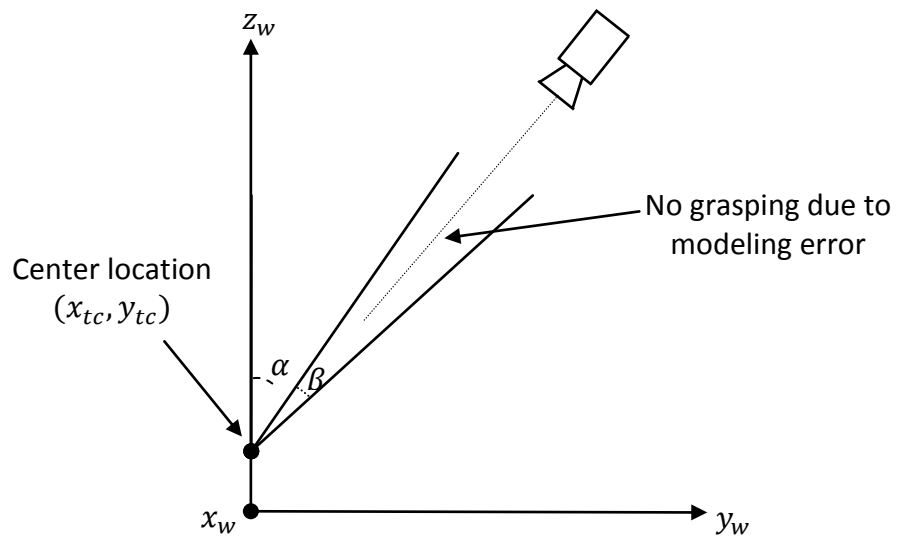


Figure 6-38 Exaggerated visualization of the grasps that are removed due to the vision modeling error.

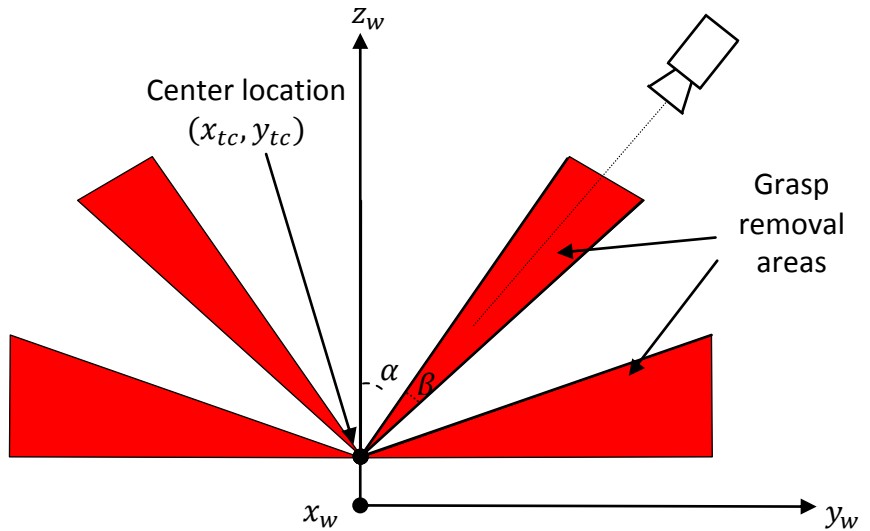


Figure 6-39 Exaggerated visualization of the all of the areas that have to be removed from modeling error and gripper maximum angle constraint. (not to scale).

5. To prevent a collision with the table the gripper jaws have a check based on the gripper geometry, grasping angle and location of the grasping point above the table z_{tc} to ensure that the gripping jaws do not come into contact with the table surface. The collision geometry is shown in Figure 6-40.

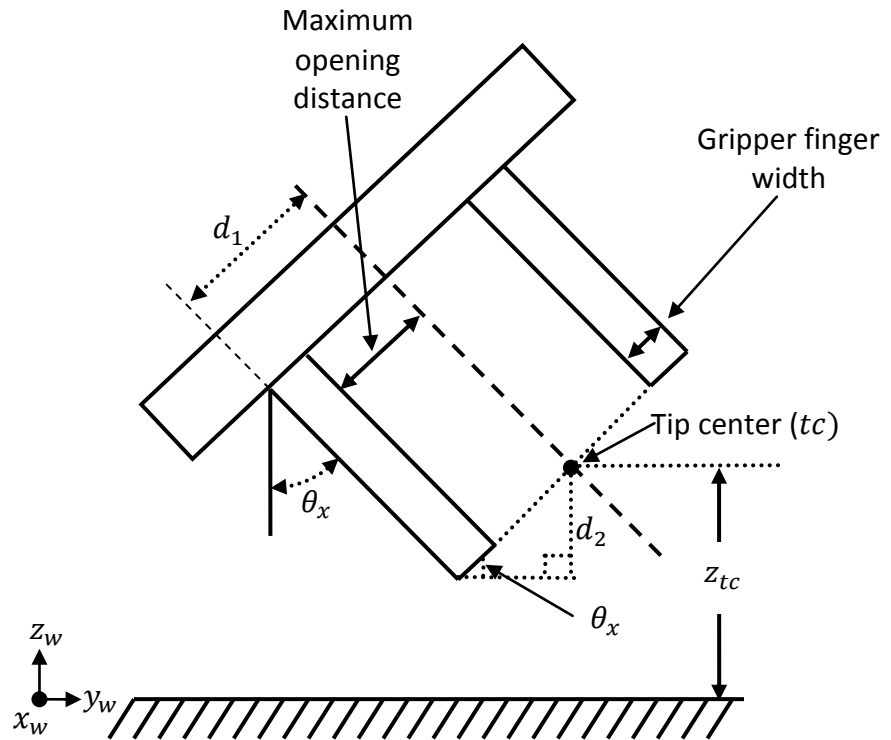


Figure 6-40 Geometry of the gripper relative to the table

As seen in Figure 6-40 a right triangle is formed with an angle of θ_x , hypotenuse d_1 and opposite side d_2 . d_1 is the half of the maximum opening distance of the gripping jaws plus the width of a single finger, d_2 is the distance of the gripping jaws plus the width of a single finger, d_2 is the vertical distance the lowest gripper finger will reach below the tip centre(tc). For a possible grasp the lowest gripper finger cannot be in contact with the table surface:

$$d_1 = \frac{1}{2}(\text{max gripper separation}) + (\text{Gripper finger width}) \quad (6-50)$$

$$d_2 = d_1 \sin \theta_x \quad (6-51)$$

To ensure no table contact

$$z_{tc} - d_2 > 0 \quad (6-52)$$

Combining equations 6-53, 6-54 and 6-55 gives:

$$z_{tc} - \left(\frac{1}{2} (\text{max gripper separation}) + (\text{Gripper finger width}) \right) \sin \theta_x > 0 \quad (6-56)$$

6. A minimum height above the table constraint is added (z_{tcmin}), this is just to ensure that the robot jaws never come into contact with the table. In our implementation, the constraint is set at $5mm$ so any grasp that has a z coordinate of the tip centre $z_{tc} < 5mm$ is discarded. A filter in the input to the convex hull was used so that grasps are not eliminated needlessly, this is explained in 6.4.5.

6.4.5 ***Handling Exceptions***

In Section 6.4.3 point 6 the location of the gripper jaws is calculated such that one of them entirely covers the CGF. If the lowest point on the CGF is below the 5 mm limit imposed in Section 6.4.4 point 6 then the CGF will be eliminated from the set. However, depending on the point cloud a valid grasp may exist for facet of reduced size. This problem was solved by removing the points below the 5 mm limit during the creation of the convex hull. This results in a valid CGF with all

vertices above the threshold, or insufficient points remain to create the CGF. An example of the former is shown in Figure 6-41.

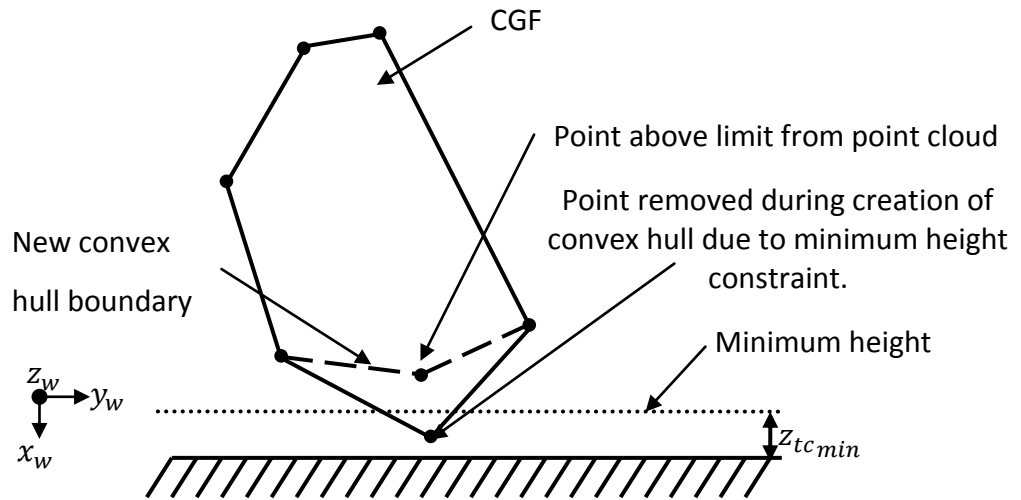


Figure 6-41 Lower boundary removal, removing the offending point from the convex hull, and hence the problem from the facet.

6.4.6 *Final Grasp Selection*

After the removal of the grasps from the grasping set S is performed as described in Section 6.4.4, the remaining grasps are compared and the grasp with maximum $grip_{val}$ is chosen as the grasp to execute, *i.e.*

$$SelectedGrasp = \max_{i=0 \dots \text{number of grips in } S} grip_{val_i} \quad (6-57)$$

If no grasp can be found then the grasp planner evaluates the grasps to find out and logs the reason, the most likely case is that the object is either too large for the gripper jaws, or too small for the gripping jaws. A grasp may also not be found on a complicated object because of the angle constraints. In these cases

the program assumes that it is possible to grasp the object in a different orientation and attempts to tumble the object into a new pose.

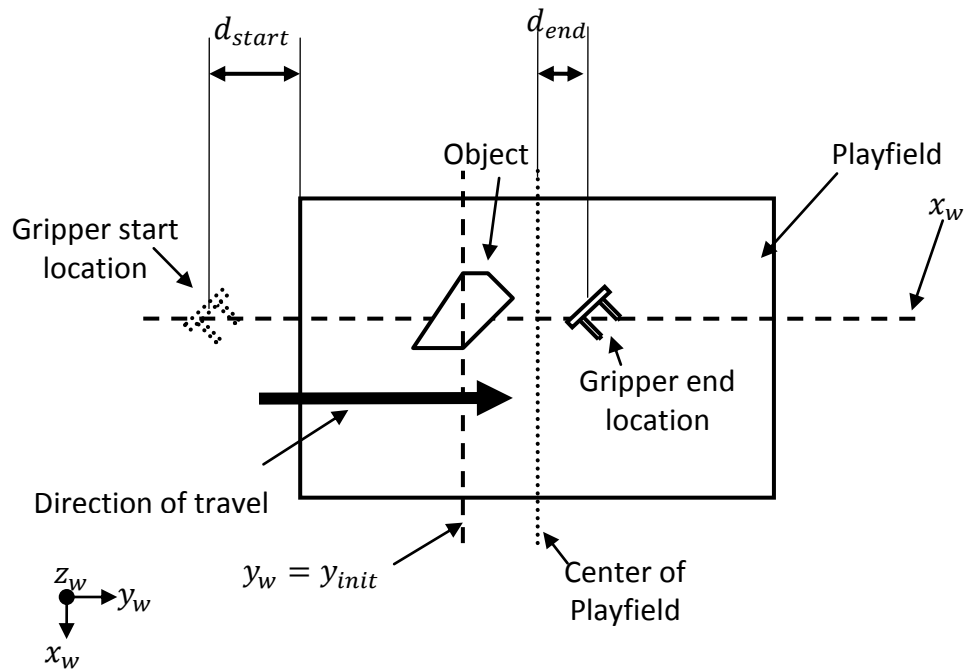


Figure 6-42 Procedure for tumbling the object when no grasp is found

To do this it takes the location of the centre of the object from the initial overhead view (x_{init}, y_{init}) and the z height as the height of the playfield, stored as a variable in the program. The object is identified as either on the left or right of the playfield using the y_{init} coordinate. The gripper is then moved to that side of the playfield remaining outside of its boundary by d_{start} and is positioned such that it's x position is the same as the x_{init} location. The height is set to the height of the playfield. The gripper is then moved horizontally along keeping the x position the same and raising the z position by $100mm$ until it reaches the

centre of the playfield + d_{end} . The raising action allows the gripper to contact the object while moving upward so that it does not just get pushed into the playfield table. When the gripper passes the centre of the table by $150mm$ it stops assuming that the object has been moved. The program then begins the grasping procedure shown in **Figure 3-3** again.

Several other methods to move the block were tried including moving the gripper beside the object and opening gripper to "pop" it into a new position. This method failed to have enough power to disturb the object. Another method involved moving towards the object in direction of the vector from the object to the centre of the playfield. This method was successful in most cases but when the object was near the extremities of the playfield the robot needed to start its move from a location outside its workspace and failed.

6.5 Grasp Execution

With the grasp selected, the execution of the grasp begins with the calculation of the approach vector, gripper orientation angles and approach point.

6.5.1 *Calculation of Approach Vector and Gripper Orientation*

Angles

The gripper will approach the object by moving along the "approach vector". The approach vector, \mathbf{a} , is calculated using the normal of the selected grasp, \mathbf{n} , and the

normal to the table, v . The method presented here is similar to the one used in the unpublished code developed by Bone *et al.* [16]. The three vectors are shown in Figure 6-44 and 6-43.

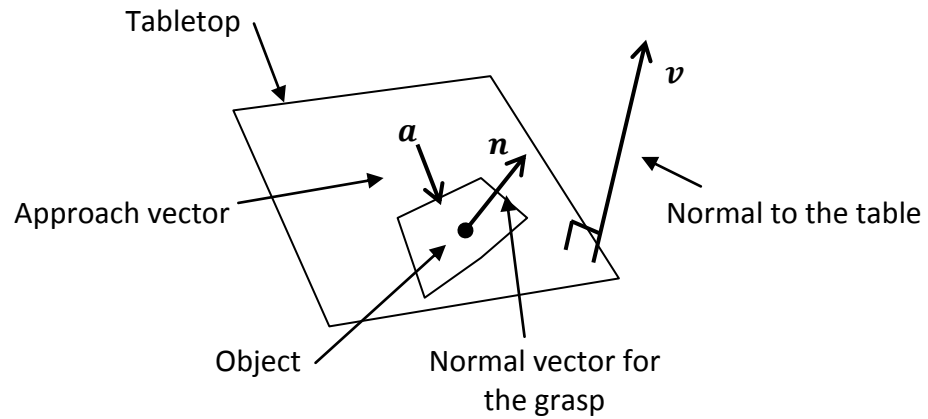


Figure 6-44 The grasp normal vector(n), table normal vector(v), and approach vector(a)

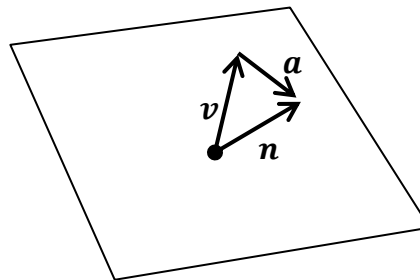


Figure 6-45 Vector a is the negative of the component of v orthogonal to n .

In equation form:

$$\mathbf{a} = -(\mathbf{v} - \text{projection } \mathbf{v} \text{ on } \mathbf{n}) \quad (6-58)$$

$$\mathbf{a} = \frac{\mathbf{v} \cdot \mathbf{n}}{|\mathbf{n}|^2} \mathbf{n} - \mathbf{v} \quad (6-59)$$

Since $\mathbf{v} = (0,0,1)$ in world coordinates the calculation of the approach vector may be simplified as follows:

$$\mathbf{a} = (\mathbf{v} \cdot \mathbf{n})\mathbf{n} - \mathbf{v} \quad (6-60)$$

$$\mathbf{a} = \left((0,0,1) \cdot (n_x, n_y, n_z) \right) \mathbf{n} - (0,0,1) \quad (6-61)$$

$$\mathbf{a} = (n_z) \cdot (n_x, n_y, n_z) - (0,0,1) \quad (6-62)$$

$$\mathbf{a} = (n_z n_x, n_z n_y, n_z^2 - 1) \quad (6-63)$$

With the approach vector known, the next step is to determine the appropriate orientation of the gripper jaws relative to the world frame. Note that the rotation around the x axis (θ_x) is taken as zero since it rotating in that direction is not helpful, see Figure 6-46. If θ_x is non-zero then as the gripper jaws move downwards their lowest corner will make contact with the table and this will increase the minimum height of the object needed for a successful grasp. The rotation also does not help with grasping the object as it is in plane of the jaws.

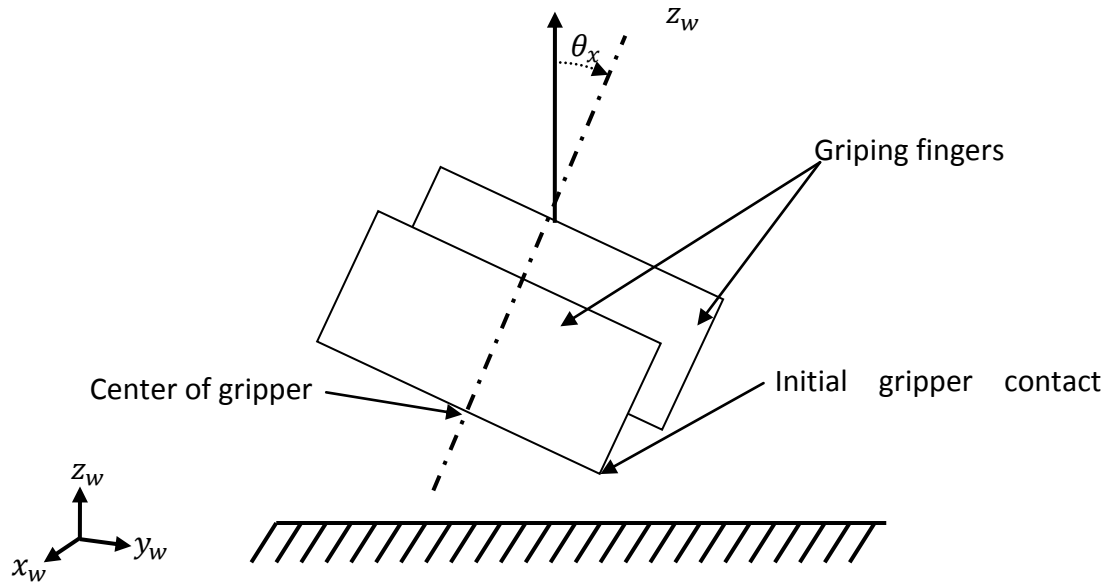


Figure 6-46 The problem with a non-zero θ_x angle. As the gripper moves downwards the bottom right corner of its jaw may hit the table before the center of the gripper is positioned to grasp the object.

The y rotation (θ_y), shown in Figure 6-47, is needed to align the associated gripper jaw with the selected facet. It is based on the projection of \mathbf{a} onto the $x_w z_w$ plane (a_{xz}) shown in Figure 6-48.

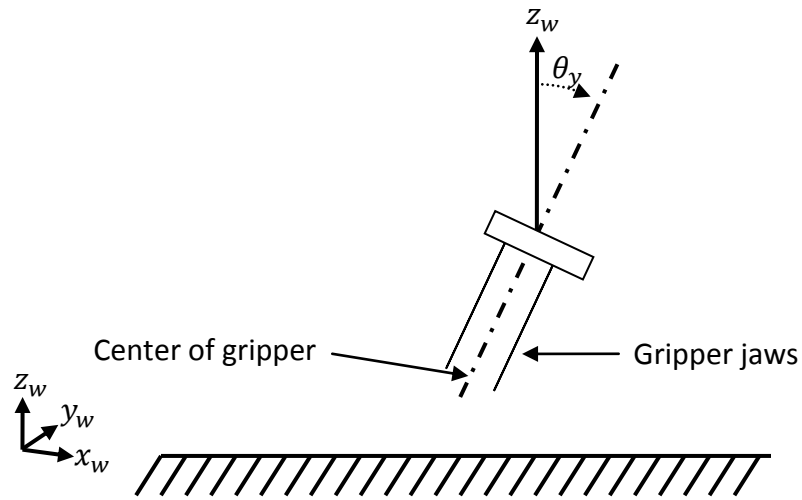


Figure 6-47 Definition of the θ_y angle.

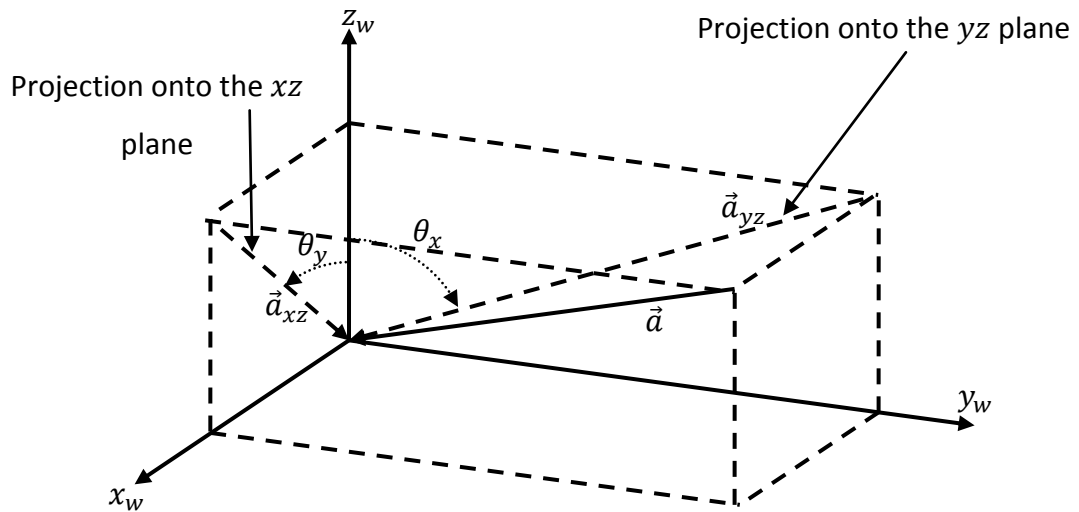


Figure 6-48 Projection of \vec{a} onto the $y_w z_w$ plane.

The angle θ_y can then be calculated as $-\frac{\pi}{2}$ plus the angle between y_w and $-\vec{a}_{yz}$

$$-\theta_y = \frac{\pi}{2} - \frac{(1,0,0) \cdot -\mathbf{a}_{yz}}{|(1,0,0)| |-\mathbf{a}_{yz}|} \quad (6-64)$$

$$\theta_y = \frac{(1,0,0) \cdot -\mathbf{a}_{yz}}{|(1,0,0)| |-\mathbf{a}_{yz}|} - \frac{\pi}{2} \quad (6-65)$$

Recalling Figure 4-1 and Section 4.2 equation (4-1) this may be change from world coordinates to robot base coordinates as follows:

$$\theta_y = \frac{(-1,0,0) \cdot -\mathbf{a}_{yz}}{|(-1,0,0)| |-\mathbf{a}_{yz}|} - \frac{\pi}{2} \quad (6-66)$$

To calculate the z rotation(θ_z), $-\vec{a}$ was projected onto the xy plane as:

$$-\mathbf{a}_{xy} = (-a_x, -a_y, 0) \quad (6-67)$$

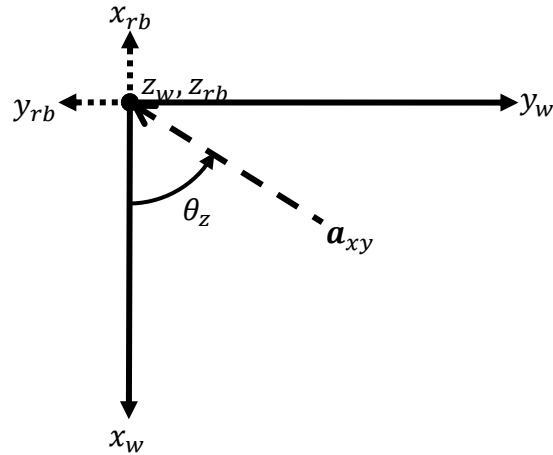


Figure 6-49 Calculation of the z rotation (θ_z)

The z rotation(θ_z) for grasping is then calculated as:

$$\theta_z = \text{atan2}(-a_y, -a_x) \quad (6-68)$$

With a symmetrical parallel jaw gripper its orientation about the Z_w axis can be rotated by $\pm\pi$ without affecting the grasp. To ensure the gripper and robotic arm do not collide an initial gripping angle θ_z is modified such that it lies in the region of 0 to π by adding or subtracting π radians if necessary.

An additional constraint is added to avoid the collision of the laser bracket and robotic arm. As shown in Figure 6-50, the bracket that holds the line laser can collide with the robotic arm (collision shown in red) if the arm rotates to far in the positive direction

around Z_{rb} , and the gripper rotates in the negative direction around Z_{rb} . This is most evident when y_{gw} is positive. To avoid this collision the bracket was oriented to always face away from the robotic arm. A check was added using the y_{gw} coordinate of the gripper, if it was less than -50mm (this allows for the bulk of the arm) the gripping angle would be modified so that it was greater than $-\frac{13}{36}\pi$ by adding π radians the limiting case is illustrated in Figure 6-51.

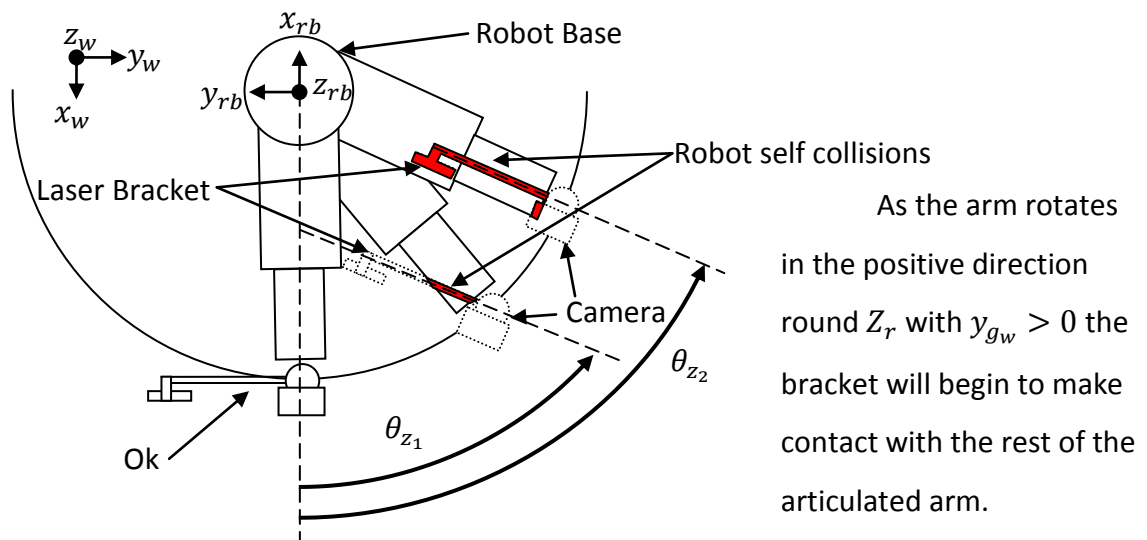


Figure 6-50 Robotic arm physical constraints that must be considered to avoid collision with the laser bracket

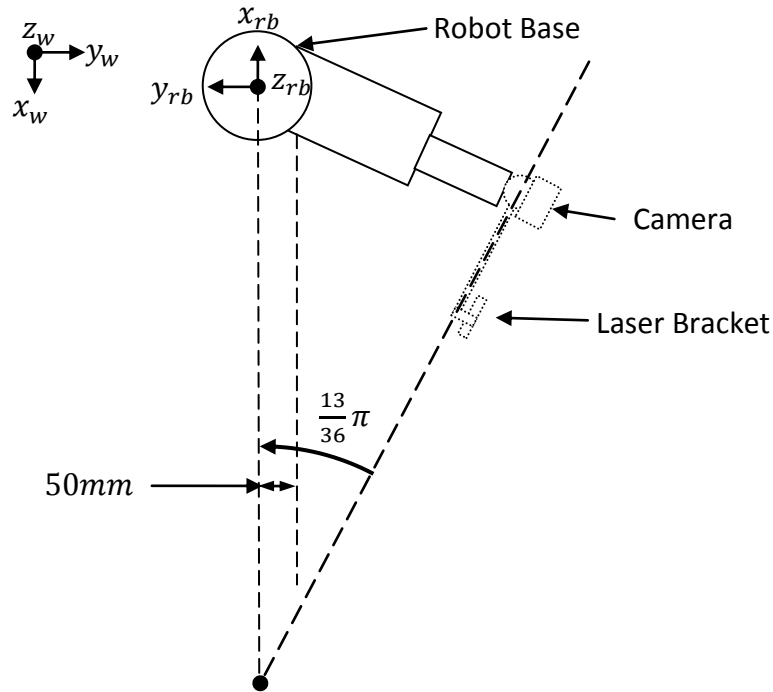


Figure 6-51 Robotic arm physical constraints that are enforced whenever the arms y_r coordinate is above **50mm**

6.5.2 *Calculation of approach point*

The approach point was calculated using the approach vector.

$$p_{approach} = tc - d_{approach} \mathbf{a} \quad (6-69)$$

Where:

$p_{approach}$ is the start point of the approach and,

$d_{approach}$ is the distance away from the the tip centre(tc) that the approach point should be (in our implementation $d_{approach} = 45mm$).

Chapter 7

Evaluation Technique

The following chapter describes the objects and procedure created for grasp evaluation.

The grasp evaluation method was developed as a result of the lack of a common grasping metric used in the current literature as discussed in Section 2.3. This chapter describes the breakdown of the types of evaluation used, the types of objects, and the methods that were created to evaluate the grasps.

7.1 Grasp Evaluation Purpose

The evaluation technique was developed to measure two separate, yet important, aspects of automated grasping. The aspects that were evaluated were: “grasp repeatability” and “graspability”. Grasp repeatability is defined as the ability to grasp the same object in a similar manner, and graspability is defined as the ability of the system to grasp an arbitrary object. High grasp repeatability is useful during the handling of objects since similar objects will be held with a similar pose. For example if a grasping system was tasked with stacking bricks, it would be preferable to have the system grasp the blocks with a similar pose to ease the brick laying calculations with respect to the location of the brick in the gripper(it would in the worst case present a good starting guess for the pose of the brick). Grasp repeatability can be measured by grasping the same object many times and measuring the difference in pose between all grasps. Graspability is useful to evaluate the ability of the grasp planner to grasp random objects

in random poses. This evaluation is useful when an object's shape cannot be predicted or has never been seen before; these objects can exist in the mining or similarly, any other raw material industry. The objects that were used for grasp evaluation will be discussed in Section 7.3. The evaluation procedure will be discussed in the next section.

7.2 Object Grasp Evaluation Procedure

To ensure consistency a constant procedure was used to evaluate the grasps; this procedure was used for both types of evaluations. The steps used to evaluate an attempted grasp begin after the grasp was attempted. The gripper was moved to a predefined location defined as the "pose" position, this position is shown in Figure 7-1. When the robot was stationary in the pose position, the pose evaluation camera captured an image of the gripper jaws, and any object that may have been grasped. The pose location also included two separate halogen illumination lights for consistent lighting of the object from both the top side and bottom; these lights were controlled by the computer through a custom circuit board via an opto-relay (more information on the setup is given in Chapter 3). An example image of a rectangular box with multiple circular targets taken from the pose evaluation camera is shown in Figure 7-2.

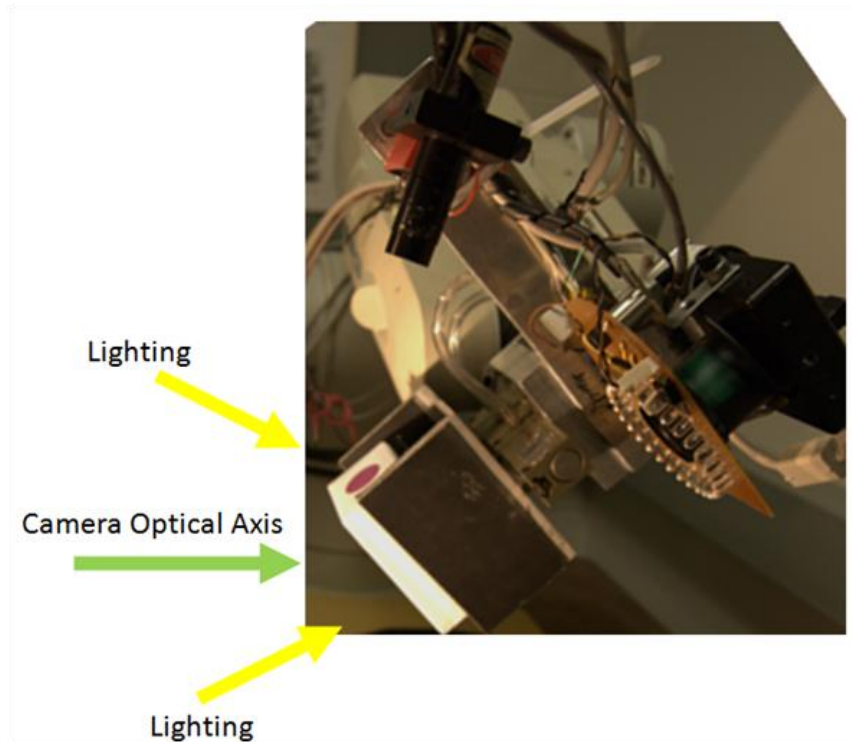


Figure 7-1 The robot in the pose position with and object from set one in the gripper.

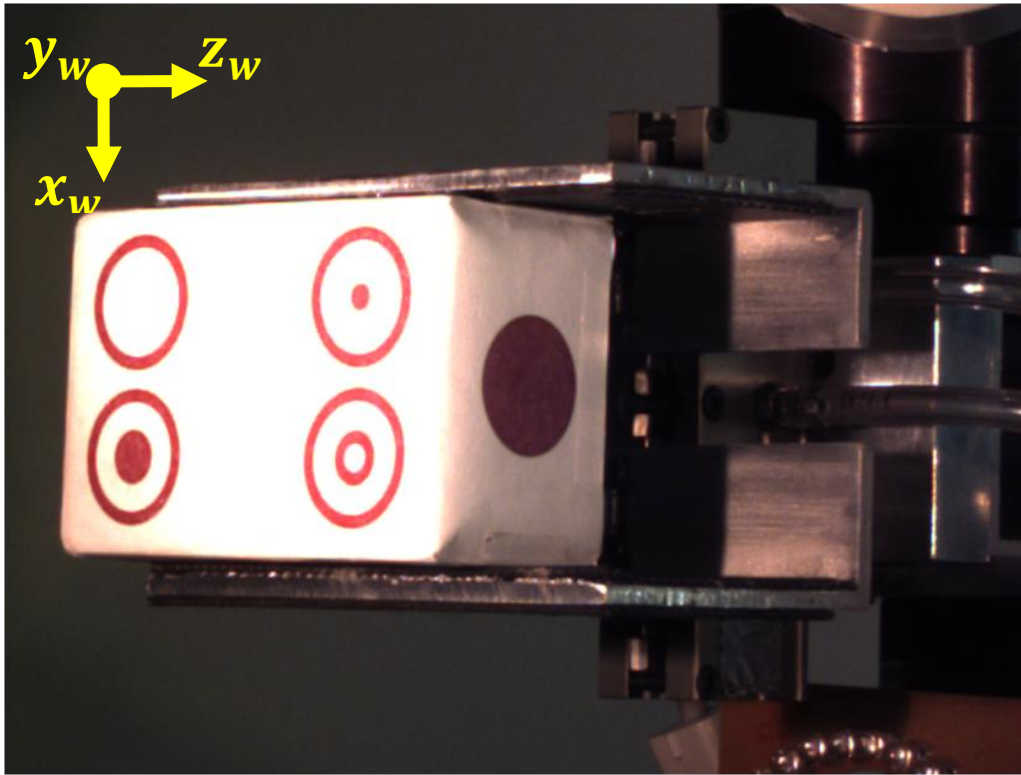


Figure 7-2 An image of a rectangular box taken by the pose evaluation camera.

7.3 Object Categorization

The objects to be grasped were separated into the two evaluation categories and named similar to the grasp evaluation categories as “grasp repeatability objects” and “graspability objects”. Separate experiments were run for each object.

The first set, grasp repeatability objects (where changes in pose were measured), had circular visual targets attached to them. These targets were used for identification of the object pose only and did not play a role in grasp planning or execution. The second set of objects, graspability objects, included real world and concocted objects that were used

to determine if the grasp planner could successfully grasp and lift differently shaped objects. Since the complexity of attaching targets to most of the graspability objects was high, and the likelihood of being able to see the minimum required amount of them to identify its pose could be very low, the graspability objects were only evaluated using a binary pass/fail test.

7.3.1 ***Grasp Repeatability Object***

The grasp repeatability object was used to measure the pose differences for the same object in a set of successive grasps. The object was fitted with optical targets (see Figure 7-3) and then grasped many times on both level and non level surfaces. Each time a grasp was attempted a separate image was taken with the robot in the pose position. Using this pose information a frame was attached to the object, and termed the object frame (o). Each objects pose was computed using POSIT [34], this was compared to a baseline pose of the same object placed in the gripper jaws by a human operator termed the perfect pose frame (p). The difference or offset between the frames was calculated, as

$$offset = o - p \quad (7-1)$$

where

$$offset = \begin{bmatrix} \theta_{offset_x} \\ \theta_{offset_y} \\ \theta_{offset_z} \end{bmatrix} \quad (7-2)$$

see Figure 7-4 and Section 7.3.2.

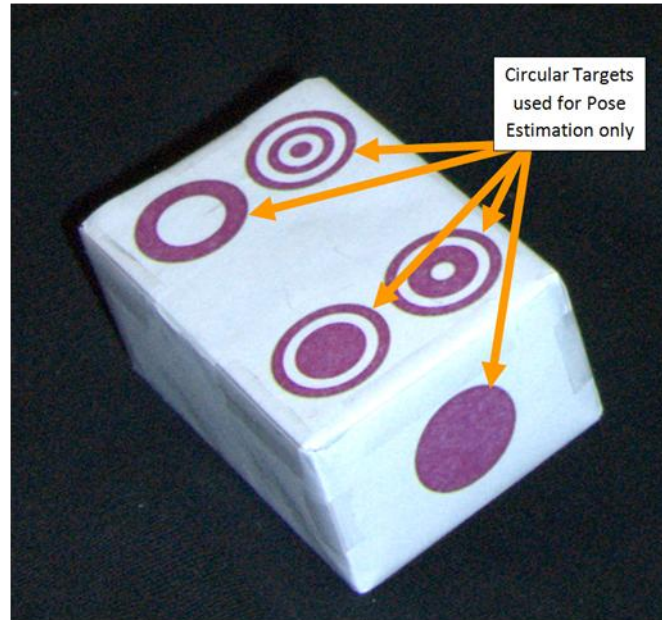


Figure 7-3 The grasp repeatability object, shown with circular targets attached.

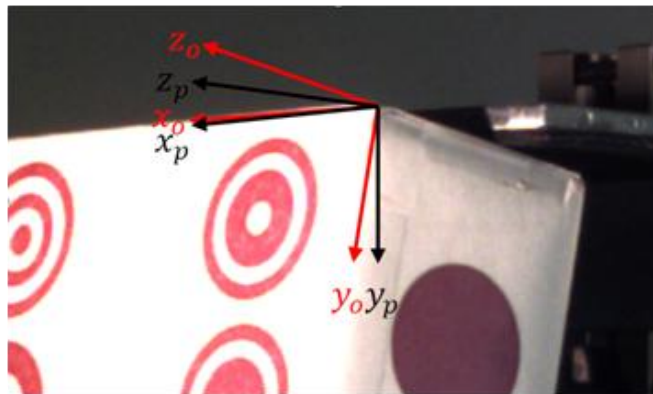


Figure 7-4 The grasp repeatability object with the orientations of the perfect pose(p) frame and object(o) frame shown.

7.3.2 *Offset Calculations*

To calculate the angles between the different axis a homogeneous transformation matrix H_p^o between the frame p and o was calculated using:

$$o = H_p^o p \quad (7-3)$$

$$o(p)^{-1} = H_p^o p(p)^{-1} \quad (7-4)$$

$$H_p^o = o(p)^{-1} \quad (7-5)$$

With the transformation matrix formed, the calculation of the rotational offsets are:

$$\theta_{offset_y} = -\sin^{-1}\left(H_{p,3,1}^o\right) \quad (7-6)$$

$$\theta_{offset_x} = \text{atan2}\left(\frac{H_{p,3,2}^o}{\cos\left(\theta_{offset_y}\right)}, \frac{H_{p,3,3}^o}{\cos\left(\theta_{offset_y}\right)}\right) \quad (7-7)$$

$$\theta_{offset_z} = \text{atan2}\left(\frac{H_{p,2,1}^o}{\cos\left(\theta_{offset_y}\right)}, \frac{H_{p,1,1}^o}{\cos\left(\theta_{offset_y}\right)}\right) \quad (7-8)$$

Another set of offset angles exist for a second θ_{offset_y} where

$$\theta_{offset_y} = \pi + \sin^{-1}\left(H_{p,3,1}^o\right) \quad (7-9)$$

This creates a second θ_{offset_x} , and θ_{offset_z} but as we will show in Section 7.7 we can ignore, due to symmetry, rotations of $\frac{\pi}{2}$ so we just evaluate the first set of rotations.

7.3.3 ***Definition of Graspability Objects***

The second set of objects, graspability objects, included both concocted objects (created to test different basic shapes), and real world objects. The objects were chosen so that they would fit inside the gripper, and would include common and uncommon shapes with both curved and flat sides. Curved sides are important because they limit the area of contact with the gripper jaws making the grasping task more difficult. The graspability objects are shown in Figure 7-5 and include non-convex block like shapes, a half circle, and a hexagonal nut. Each object was evaluated on a pass fail basis; if the object was present in the pose image then it was grasped successfully, if it was absent then the grasp failed.

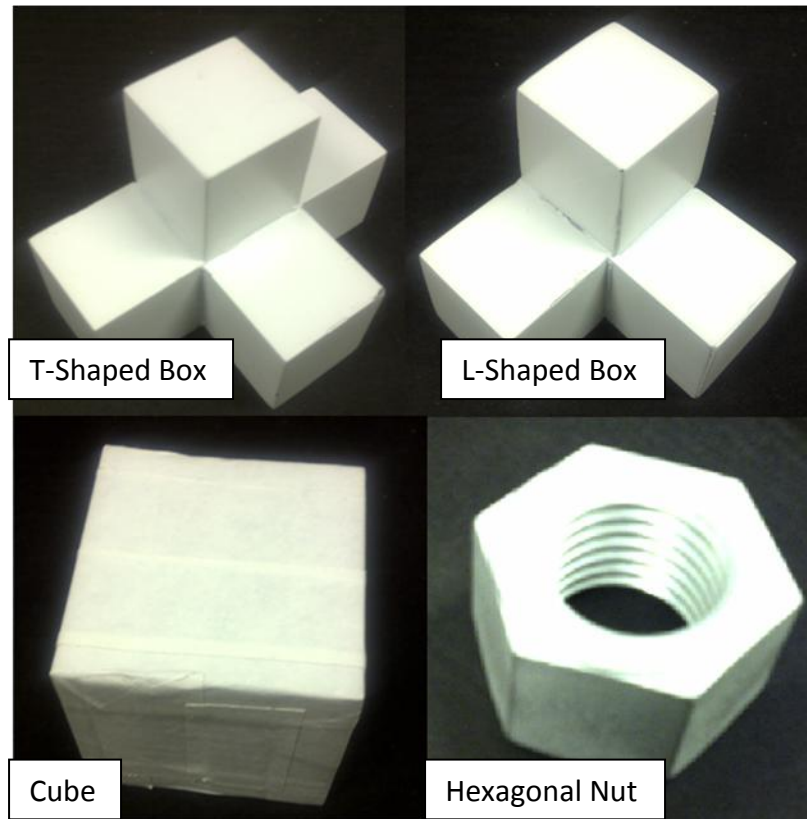


Figure 7-5 Graspability objects.

7.4 Object Pose Identification

To evaluate the pose of the grasp repeatability object, a separate program was written that took the images from the pose position (pose images) as input, found the targets, and computed the pose. The following sections describe this procedure.

7.5 POSIT Computation

The pose estimation was performed using OPENCV [33] and POSIT [34] library.

The POSIT algorithm requires least 3 non-coplanar target points located on the object, in the case of the rectangular block there are targets on the large faces and sides for this

reason. These points are then matched to locations on a model of the object stored on the computer. The pose is then computed from the location of these points on the model in virtual space and the locations of the actual target points on the image plane.

7.6 Target detection and Identification

The following information and algorithm was used to identify the targets attached to the objects in set one. The targets had to be identified to allow the POSIT program to compute the pose of the object. The basic steps for identifying the target were to segment the image in hue, saturation and value (HSV) space (see Appendix C for an explanation of HSV space). The results were then filtered and a boundary following algorithm found the outside of each circle. Finally, an identifier stored in each target was extracted along with the target location on the image plane.

7.6.1 Target Segmentation

To identify the coloured targets, ranges for the hue, saturation and value was set up. Several initial test images of the targets were taken, converted to the HSV colour-space and analyzed to find the initial threshold values. The initial values were then slightly refined during testing if a target was not, or partially found, or if too much background noise was present in the filtered image. The thresholding converted the image from the HSV space into a 2-D binary image matrix. The red targets were found to lie within the range of:

Table 7-2 HSV threshold values for red targets, front set.

Hue	$H > 280, \text{ or } H < 15$
Saturation	$0.4 < S < .87$
Value	$100 < V \leq 255$

Another second set of thresholds were added due to a large difference in lighting of the targets on the side of the object, the difference in apparent colours of the targets can be seen in Figure 7-6. This image was used to determine the thresholds because it had the largest difference in appearance between the front and side targets.

Table 7-3 HSV threshold values for red targets, side set.

Hue	$H > 280, \text{ or } H < 5$
Saturation	$0.25 < S < .7$
Value	$25 < V \leq 120$

The hue has two separate thresholds in each set because Red in the HSV colour-space includes the jump from 360° to 0° . An example of a post threshold image of the object is shown in Figure 7-7. As the image shows, the background included many false positive pixels.

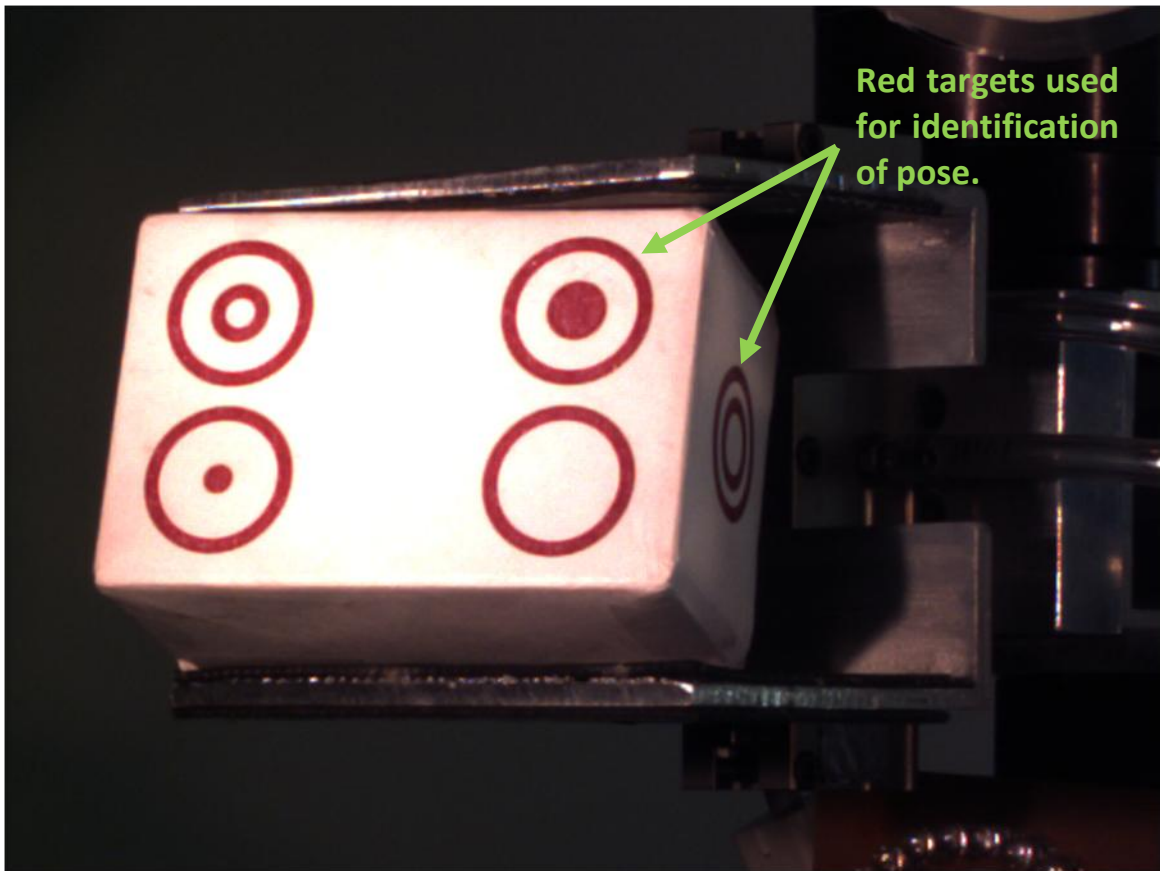


Figure 7-6 This image shows the object and the variations in the apparent colour of the red targets. Note that all of the targets are the same true colour.

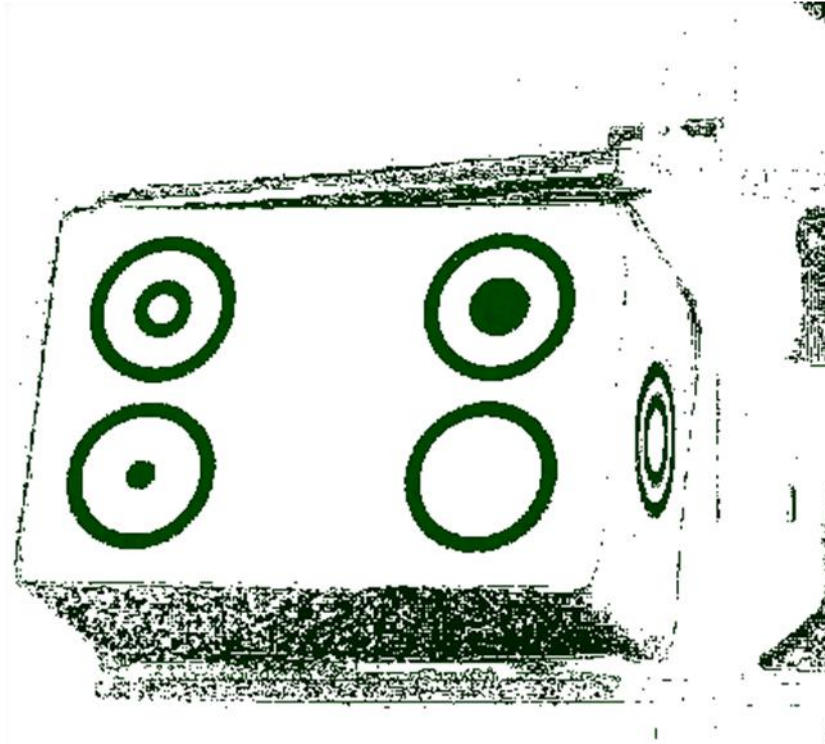


Figure 7-7 Image of the object after thresholding in the HSV colour-space.

To remove some of the noise in the image, it was filtered initially using a morphological open close using the filter element shown in Table 7-4.

Table 7-4 The element used for the morphological open, and morphological close.

0	1	0
1	1	0
0	1	0

After the morphological open close, Figure 7-12, the image was further processed to find the circular targets using the boundary following algorithm in 7.6.2.1.

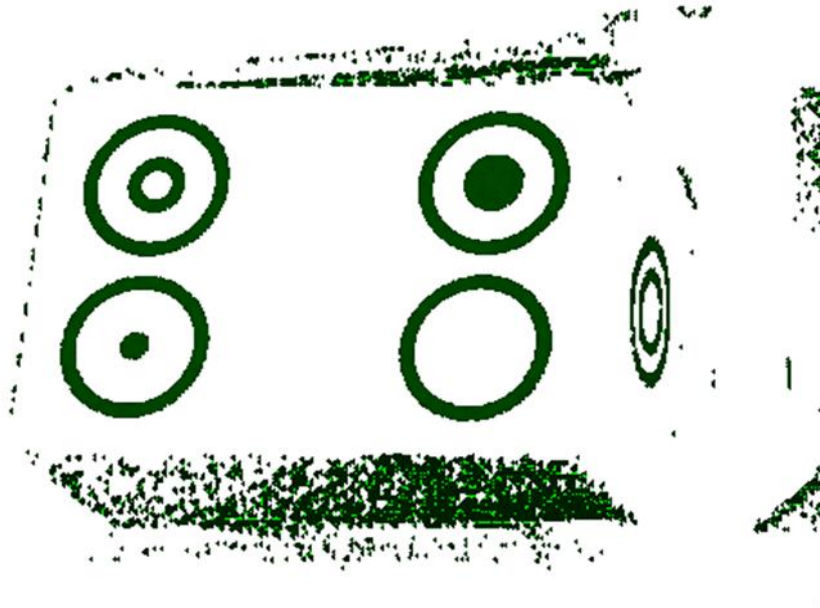


Figure 7-8 The image of the object after the morphological open close filter.

7.6.2 ***Barcode Detection and Code Extraction***

After the initial noise was removed from the image the barcode targets locations and identifiers (ID's) were found and extracted.

7.6.2.1 ***Boundary Following***

To remove more noise and identify image components as a barcode target a boundary following algorithm was written. The initial location to start the boundary finding algorithm was found by scanning horizontally across the image, when a non-background pixel was found the boundary following algorithm began. The boundary following algorithm used the following notation for describing the current pixel, and the surrounding pixels:

P_i , the current pixel being evaluated.

The numbers 0 ... 7, are the surrounding pixel locations (8-neighbors) where 0 is to the left of P and 1 ... 7 are incremented in the clockwise direction.

These are shown in the following table

Table 7-5 Boundary following pixel location map.

1	2	3
0	P_i	4
7	6	5

V_i is defined as the direction to the next pixel using the 0 ... 7 notation

If $V_i > 7$ then $V_i = V_i - 8$

If $V_i < 0$ then $V_i = V_i + 8$,

The initial pixel found from horizontal scanning was labeled P_0 . To find the next pixel on the boundary a search of the surrounding pixels was used. This iterative algorithm started with the pixel in the $V_{i-1} + 3$ direction (always 7 because of the starting criteria) and then searched in a counter clockwise direction radiating out from P_i . This is the same as decrementing the direction value, during this process if a non-background pixel was found it was assigned as P_{i+1} and the current search direction was assigned as $V_i + 3$. The process is shown in Figure 7-9. The yellow box is the current pixel being

evaluated; the green box is the start of the search; the blue boxes are pixels that are on the boundary; and the red box is the next pixel found in the algorithm. The solid straight arrow is from where the last boundary location was to the current boundary location, the curved arrow shows the search direction from the start search pixel location, green box, to the next pixel found on the boundary, red box. The special case where the green and red boxes are together exists because the first box searched is also the next pixel location.

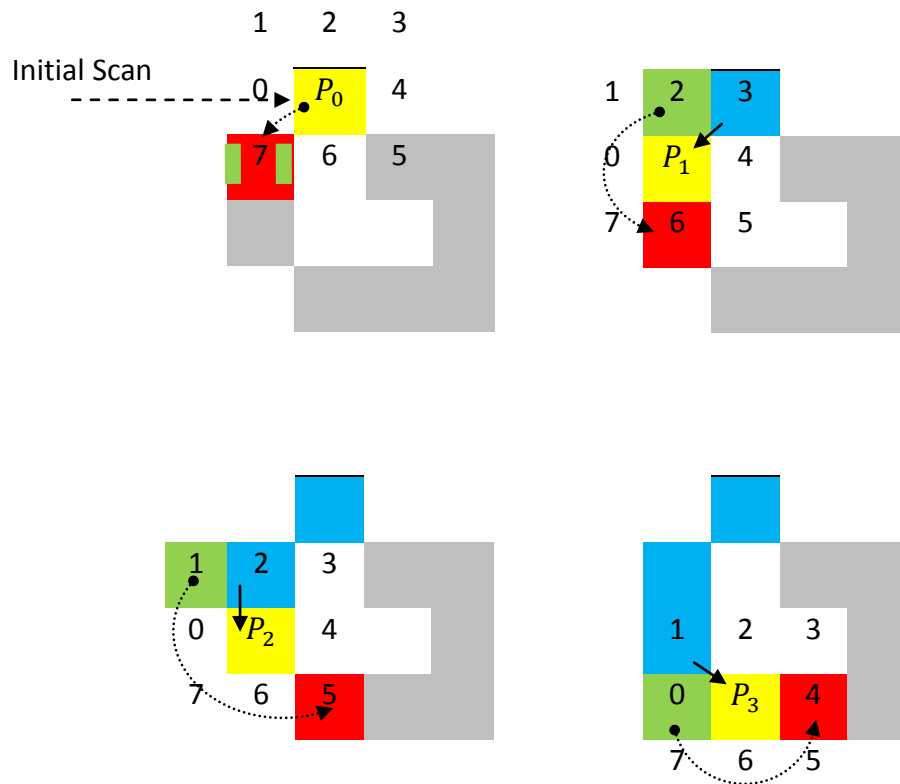


Figure 7-9 The boundary following algorithm showing four iterations.

This iterative process was continued until:

1. $P_{i+n} = P_0$, and the search direction is $V_{i+n} = 2 \vee V_{i+n} = 3$, n an arbitrary positive integer, was reached, this meant that the entire boundary had been found. The condition of $V_{i+n} = 2 \vee V_{i+n} = 3$ is used to ensure that the boundary search doesn't stop if P_0 is the median of a peninsula, this is shown in Figure 7-10, if the condition was not present the boundary would be $P_0 \rightarrow P_1 \rightarrow P_0$ because P_0 and P_2 are the same point, with the direction check the algorithm continues. This can be used because of the constrained starting direction.

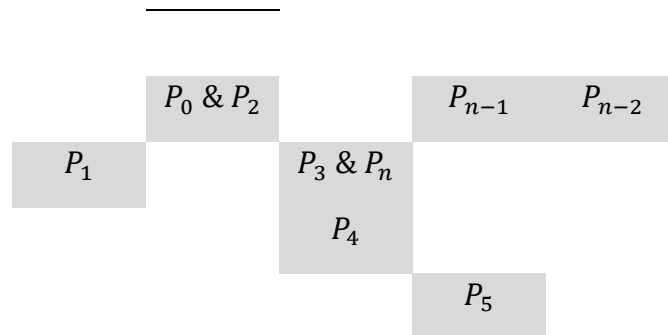


Figure 7-10 An example where the direction ending criteria is needed to ensure that the boundary is not prematurely ended.

2. The search for V_i from 0 ... 7 finds no non-background pixels, this meant that the selected pixel is a single image pixel with no neighbors.

7.6.2.2 *Target Filtering*

Two filters were used to decide if an identified object was a barcode target based on its boundary. The first was based on the length of the boundary. A minimum and maximum length threshold was set for the object. The minimum size for the boundary was set at 170pixels and the maximum size was set at 280pixels, this was based off of the minimum and maximum estimates for rotation of the object. The second condition used the minimum and maximum values in both the X, and Y directions, measuring the width and height of the object. Each circle had to be larger than 20pixels at its maximum width/height, see Figure 7-11. The detected circles are shown in Figure 7-12.

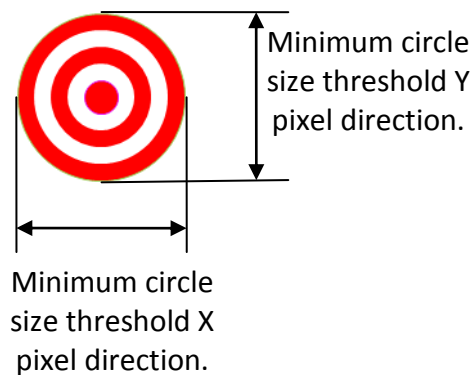


Figure 7-11 Minimum size constraint for X and Y directions.

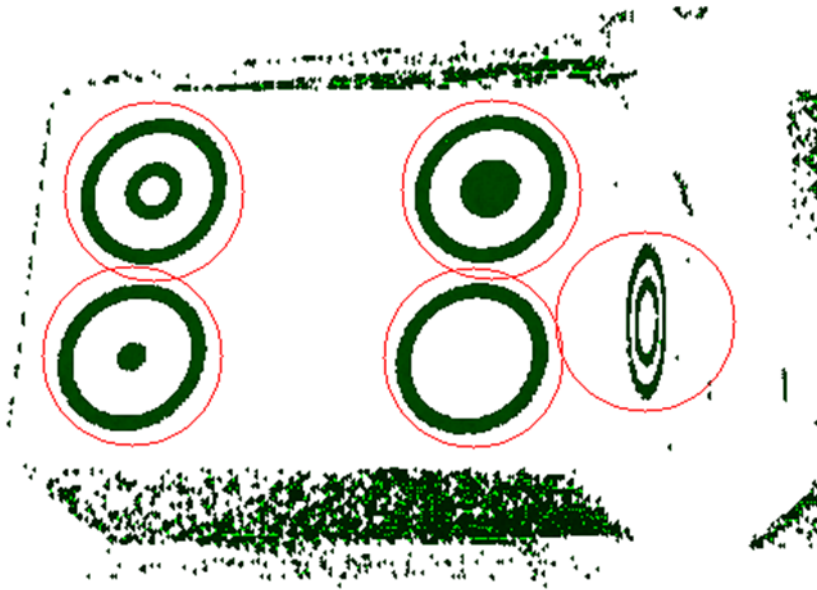


Figure 7-12 The image of the object target identification, red circles indicate target locations.

7.6.2.3 *Circular Barcode Extraction*

Along with the location of the barcode its unique identifier stored in a 1D circular ring barcode had to be extracted. A 1D ring barcode was used because its orientation does not change the ID since it can be read from inside out or outside in, and the limitation in number of IDs that could be stored $2^{number_of_rings-1}$ would not be a limiting factor (the -1 is due to the need for the outermost ring to be one so that it can be identified during boundary following). To store the ID the circle was divided into five separate rings, each ring was encoded with either a one or a zero, where the presence of a non-background pixels was represented as a one. The centre ring was labeled as ring zero

and the value was incremented while moving outwards. The encoding can be seen in Figure 7-14, and a test set of barcodes is shown in Figure 7-13.

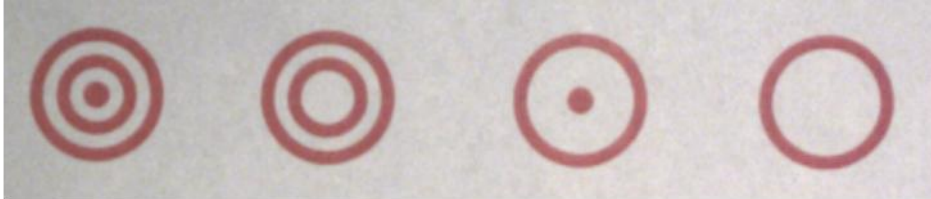


Figure 7-13 A test set of circular red targets.

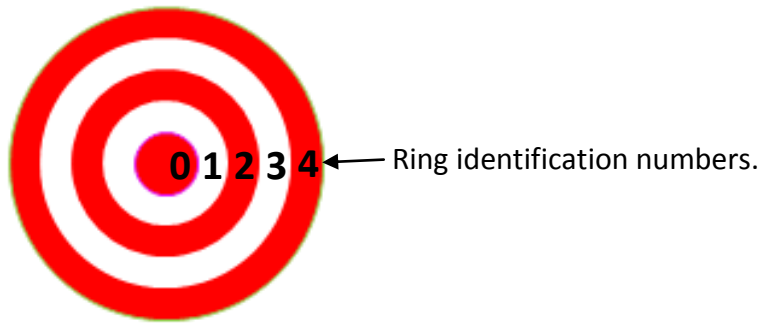


Figure 7-14 Barcode identification numbers.

The barcode ID was encoded using the ring identification number as:

$$ID = \sum_{i=0}^5 (b_{v_i}) * 2^i \quad (7-10)$$

Where

i is the barcode identification number.

b_{v_i} is the barcode value for ring i

The barcodes from Figure 7-13 are from left to right:

1. $2^0 + 2^2 + 2^4 = 21$
2. $2^2 + 2^4 = 20$
3. $2^0 + 2^4 = 1 + 16 = 17$
4. $2^4 = 16$

To determine the value of each ring the centre point of each target along with the minimum and maximum values in both the X, and Y directions were used. Each target was sampled at the centre and then four other times once for each ring, this was then repeated along the four directions (up(Y+), down(Y-), left(X-), right(X+)). Barcodes that are not nearly perpendicular to the camera's optical axis provide uneven width to height ratios that must be accounted for. When the targets were tilted by large amounts the barcode becomes skewed and hard for the vision system to read (see Figure 7-15), and some of the scanned directions reported erroneous barcodes. To combat this problem a weighting system was implemented based on the width and height of the object. Each sample along the X axis was multiplied by the width, and each sample along the Y axis was multiplied by the height. As an example when the object was tilted along its horizontal axis such as in Figure 7-15 the horizontal barcodes are easier to read and the width is larger than the height. When the barcode was tilted along the vertical axis then

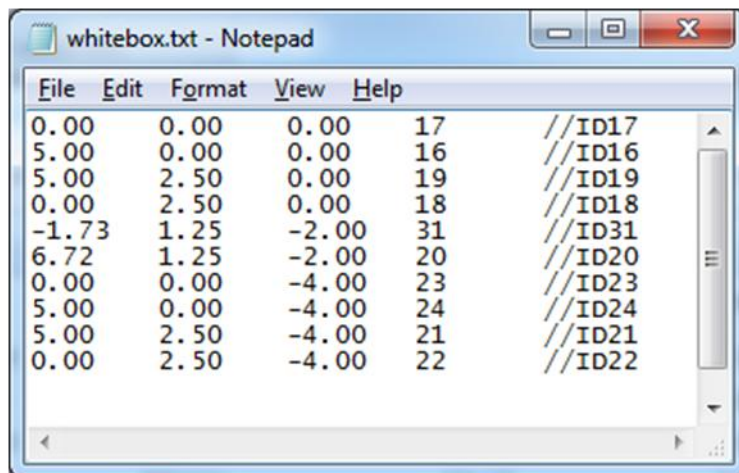
the height was larger than the width. Empirical measurements with this method allowed barcodes up to approximately 70° to be read successfully.



Figure 7-15 A circular target tilted to approximately 70° .

7.7 Calculation of Pose from Data

Once the target locations and barcodes from the image plane (X,Y) were extracted from the image, they were matched to an internal model of the object stored in the program; this model had the physical locations (x,y,z) and barcodes for each target on the object. A sample stored model is shown in Figure 7-16 where the first column is the x coordinate, the second column is the y coordinate, the third column is the z coordinate, and the last column is the barcode ID of the target.



File	Edit	Format	View	Help
0.00	0.00	0.00	17	//ID17
5.00	0.00	0.00	16	//ID16
5.00	2.50	0.00	19	//ID19
0.00	2.50	0.00	18	//ID18
-1.73	1.25	-2.00	31	//ID31
6.72	1.25	-2.00	20	//ID20
0.00	0.00	-4.00	23	//ID23
5.00	0.00	-4.00	24	//ID24
5.00	2.50	-4.00	21	//ID21
0.00	2.50	-4.00	22	//ID22

Figure 7-16 The stored model of a box, and its x, y, z components and barcode ID's.

After the matching was complete the posit algorithm generated a pose of the object in reference to the target ID17 in the pose estimate camera frame(X_p, Y_p, Z_p). This homogeneous transformation matrix can be seen in Figure 7-17. The axis of the objects computed pose was also shown on image of the object for reference, see Figure 7-18.

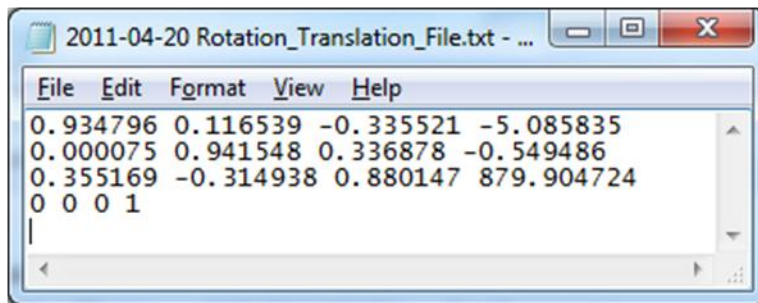


Figure 7-17 The generated homogeneous transformation matrix file for the box object.

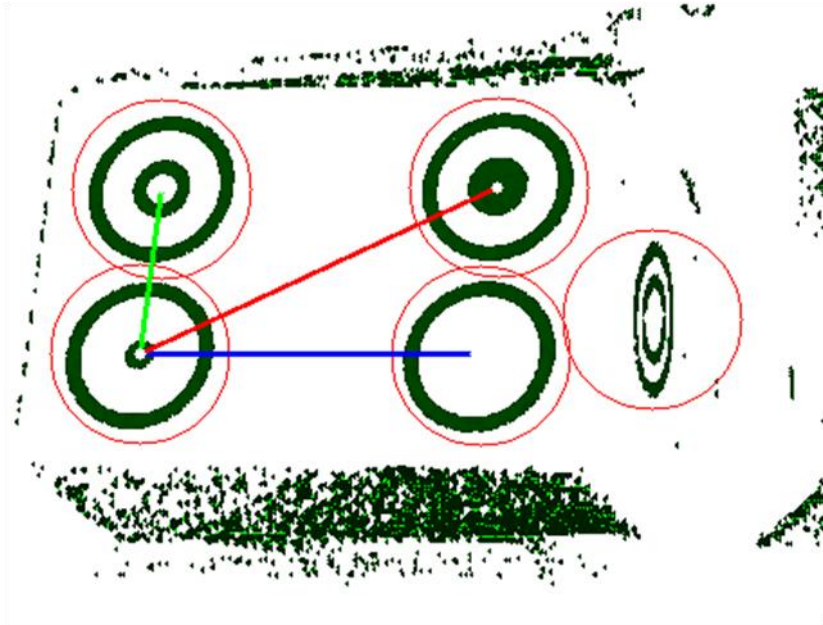


Figure 7-18 The image of the object with the targets, and its computed pose from POSEIT.

Modifications of to the box were performed such as not to unfairly skew the evaluation, because the rectangular object was symmetric around the XY , XZ and YZ planes rotations of 180 degrees along the major axis were modified such that they did not negatively alter the distribution of pose errors, because of this all rotations fell in the range of $(-\pi, \pi)$. The origin of the object frame was also set as the centre of the object instead of the base target ID17 so that rotations about any of its axis would not modify the translational measurements.

Chapter 8

Experimental Procedure

The experimental procedure describes the exact method that was used to test each grasp. It also describes in detail the grasp repeatability and graspability objects (introduced in Chapter 7).

8.1 Test Objects

The following are the objects chosen for grasp evaluation. Each object is described along with the number of unique possible parallel jaw grasps that could be successfully formed with the algorithm. For a grasp to be added to the set of possible grasps, the two contact areas would have to be unique, e.g. any rectangular prism would only have 3 unique grasps, one for each axis perpendicular to its faces. This eliminates the redundant grasps due to the rotation of the object about the jaw projected normals.

8.2 Grasp Repeatability Object

The object was a rectangular block measuring $85mm \times 57mm \times 40mm$ with a mass of $0.057kg$. The visual circular targets were located on the two larger rectangular faces as well as the two end faces. The smaller rectangular faces did not have targets attached to them because the geometry of the object and configuration of the gripper jaws did not allow for the presentation of these sides to the pose camera. An example target set for this object is shown in Appendix B.

8.3 Graspability Objects

These objects were evaluated using only the pass/fail criteria. The goal was to choose complex and non-complex shapes that would be both easy and hard for a parallel jaw gripper to grasp. The only selection criteria for these objects was that they in at least one way would be graspable by the gripper.

8.3.1 *Cube*

The first object is a cube with a 50.5mm side length, as shown in Figure 8-1. This object has 3 unique grasps, and a mass of 0.1376kg .

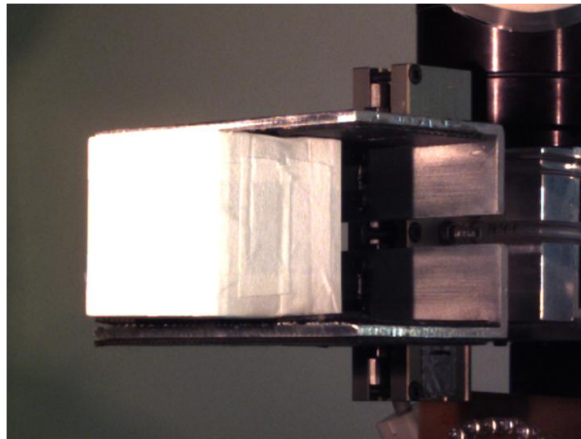


Figure 8-1 The cube graspability object.

8.3.2 *L-Shaped Block*

The second object is an L-shaped block, as shown in Figure 8-2. This object has 6 unique grasps, and a mass of 0.0164kg .

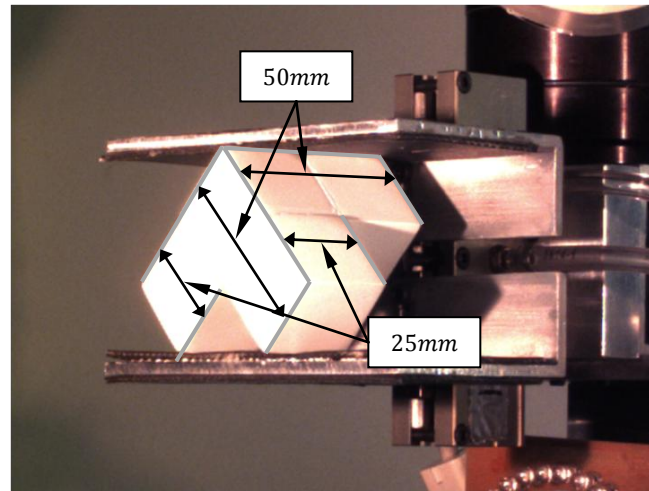


Figure 8-2 The L-shaped block graspability object.

8.3.3 *T-Shaped Block*

The third object is a T-shaped block, as shown in Figure 8-3. This object has 3 unique grasps, and a mass of $0.0204kg$.

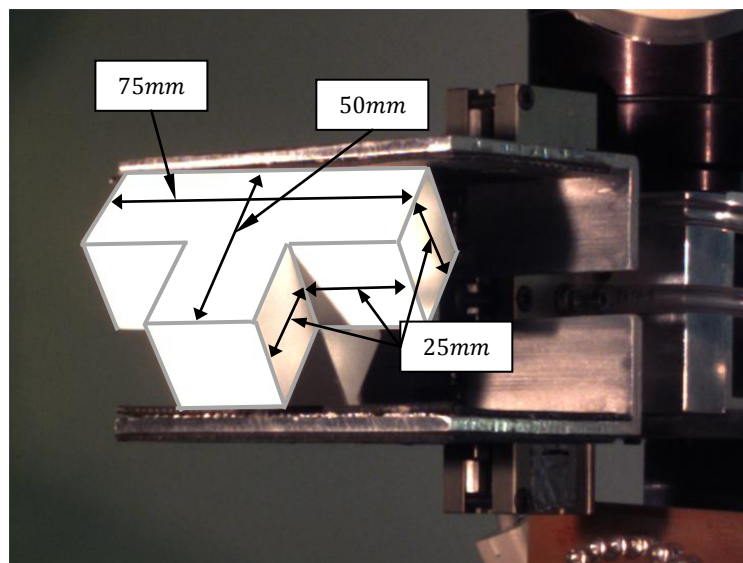


Figure 8-3 The T-shaped block graspability object.

8.3.4 *Hexagonal Nut*

The fourth object is a hexagonal nut, as shown in Figure 8-4. The hexagonal nut had 3 unique grasps, and a mass of 0.42 kg.

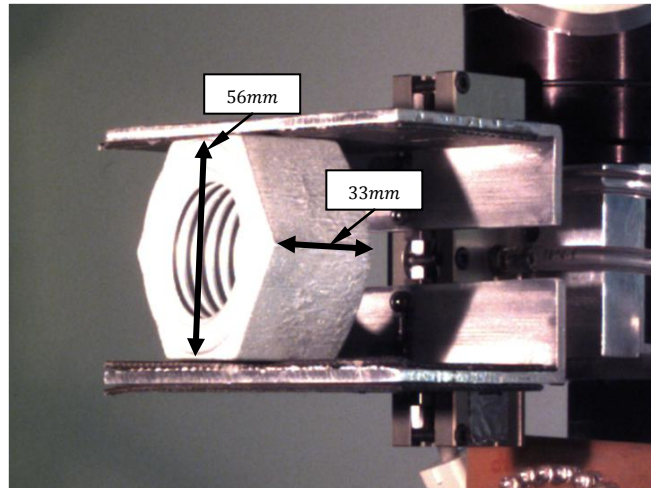


Figure 8-4 The hexagonal nut graspability object.

8.4 **Baseline System Errors**

To allow comparison of the relative sizes of errors between the robot, cameras, and grasp planner, initial testing was done to evaluate the robot and camera errors. These errors were only used the analysis of the grasp repeatability object one because only its variations in pose could be measured.

8.4.1 *Imager Pixel And Stationary Robot Errors*

The sizes of the errors from the pose camera imagers CCD due to changing lighting conditions, robot vibration/robotic control system errors and pixel interpolation were

estimated as follows. The robot was held stationary with the rectangular block object grasped in the pose position, 200 images of the rectangular block were then taken. The optical targets positions were then evaluated and the variations in target pose were recorded and stored.

8.4.2 ***Robot Repeatability Errors***

To quantify the robot repeatability errors the rectangular block was again inserted into the gripper jaws and the robotic arm was moved from random locations just above the table, to simulate a grasp attempt of the object, to the pose position. An image was then taken by the pose camera and stored. This entire procedure was repeated 200 times. The variations in target poses were recorded.

8.5 **Grasping**

After the errors of the camera system, and robot were measured, two sets of grasping trials were performed for the objects. The first used a flat tabletop robotic workspace, and the second included the addition of the playfield to the robotic workspace.

8.5.1 ***Flat Tabletop Workspace***

The flat tabletop was used to evaluate the grasping algorithm when the test object was not tilted, and was resting flat on the table. This allowed the object's initial pose to vary with 3 DOF, i.e. translation along the X_w and Y_w , and rotation about Z_w .

8.5.2 ***Playfield***

The playfield was used to evaluate the grasping algorithm so that the object would always be in a random pose and not resting flat on the tabletop. This allowed the object's initial pose to vary with all 6 DOF.

8.5.3 ***Testing procedure for grasps***

The following standard test procedure was used for every object:

1. Find the object to be grasped from an overhead image.
2. Capture images from multiple viewpoints.
3. Image processing and preliminary model generation.
4. Laser scanning, and model refinement.
5. Grasp solution computation.
6. If no grasp exists, then perform the tumbling routine (described in Section 6.4.6) forcing the object into a new pose. Continue with step 1.
7. If a grasp does exist then execute the selected grasp.
8. After the grasp attempt move to the predefined pose position
9. Capture pose position image.
10. Place object into a random pose located on the table.
11. If the desired number of grasp attempts has not been achieved then continue with step 1, else stop.

The random placement of the object ensured that the next grasp attempt would be performed in a different location on the tabletop. When the object was placed on the table it was also randomly rotated about the Z_w axis so that it came to rest in a random orientation. The playfield provided rotations of the object along the X_w and Y_w axis. A uniform random generator was used for both the location and rotation of the objects so that no bias could be given to grasps located at specific locations on the playfield. The random number generator was seeded with the current time, and gave a number in the following ranges:

$$510 < x_w < 580 \quad (8-1)$$

$$-100 < y_w < 100 \quad (8-2)$$

$$230 < z_w < 250 \quad (8-3)$$

$$-0.873 < \theta_{z_w} < 0.873 \quad (8-4)$$

This allowed the object, after hitting the playfield, to come into rest in the robots dexterous workspace.

When the grasping evaluation algorithm finished its series of grasp attempts the set of pose images, one for each grasp attempt, were stored on the PC. The images were then post processed to find the pose offset of the grasp repeatability object, and to determine the binary pass/fail criteria for each of the graspability objects. All of the experimental results will be presented and discussed in the next chapter.

Chapter 9

Experimental Results

This section describes the results of the grasping experiments. It also discusses the base non grasp related errors in the system that occur from robot and camera repeatability.

9.1 Stationary Robot Test Results

For this test, the robot was held stationary with the rectangular block object grasped in the pose position, and 200 images were taken. These were post processed to obtain the target centroid positions and object pose results whose histograms are plotted in Figure 9-1 and Figure 9-2 respectively. Means and deviations are presented in Table 9-1 and Table 9-2.

9.1.1 Stationary Robot Test Results Data

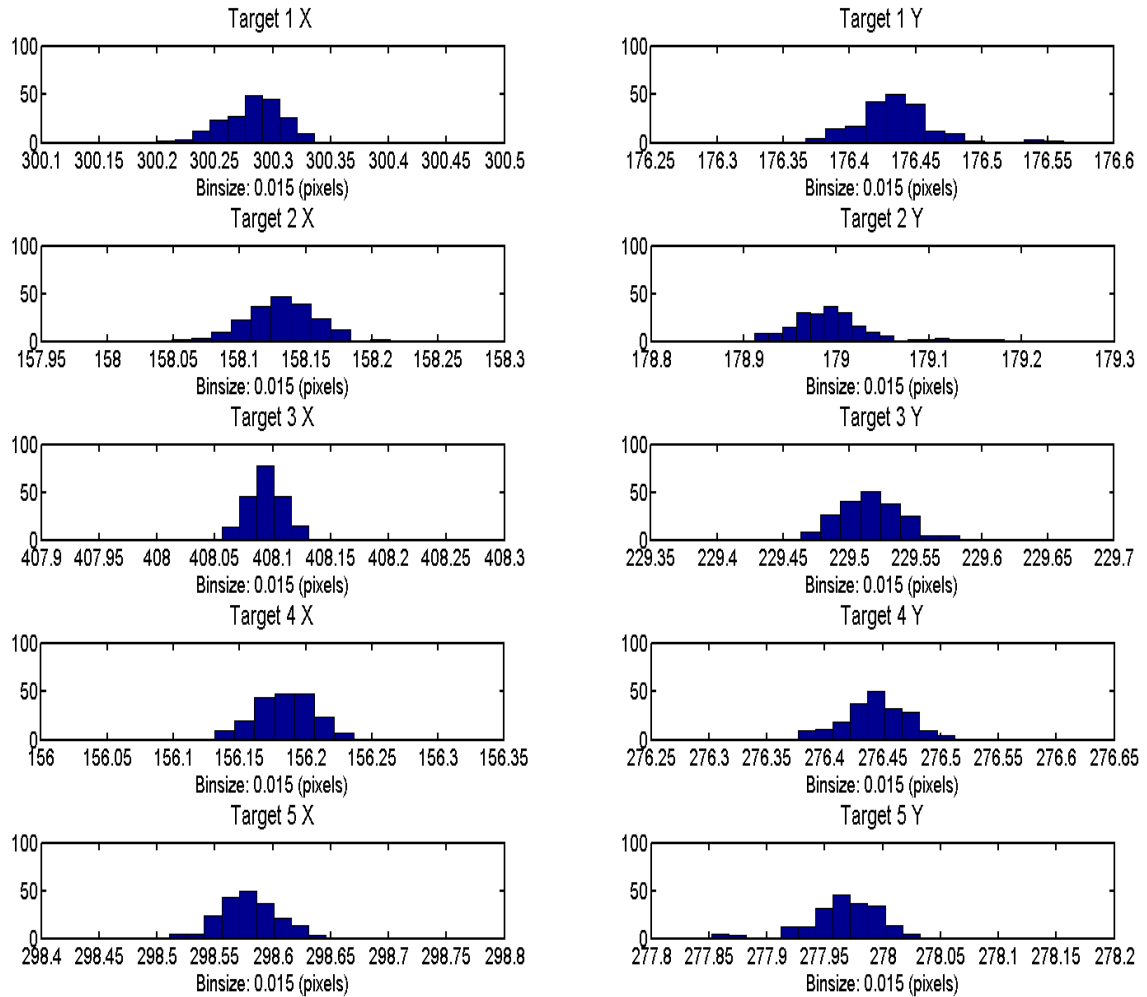


Figure 9-1 Histograms of the object centroids from the stationary robot test.

Table 9-1 Means and standard deviations of target centroids

Variable	Mean (pixels)	Standard Deviation (pixels)
Target 1, (X_o)	300.28	0.026
Target 1, (Y_o)	176.43	0.032

Target 2, (X_o)	158.13	0.026
Target 2, (Y_o)	178.99	0.043
Target 3, (X_o)	408.09	0.015
Target 3, (Y_o)	229.52	0.022
Target 4, (X_o)	156.18	0.021
Target 4, (Y_o)	276.44	0.027
Target 5, (X_o)	298.58	0.025
Target 5, (Y_o)	277.96	0.033

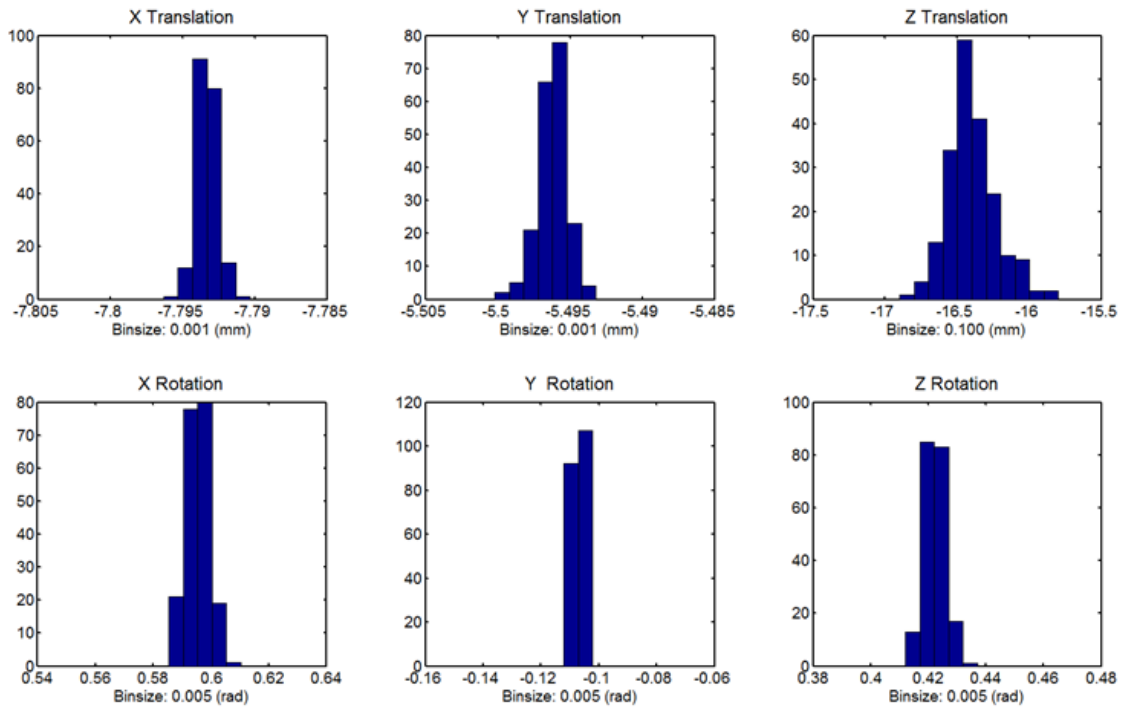


Figure 9-2 Histograms of the object pose frames from the stationary robot test.

Table 9-2 Means and standard deviations of the pose frame.

Variable	Mean	Standard Deviation
$X_{translation} (mm)$	-7.793	0.001
$Y_{translation} (mm)$	-5.496	0.001
$Z_{translation} (mm)$	-16.398	0.167
$\theta_{Xrotation} (rad)$	0.595	0.004
$\theta_{Yrotation} (rad)$	-0.107	0.0005
$\theta_{Zrotation} (rad)$	0.422	0.003

9.1.2 ***Stationary Robot Test Results Discussion***

The standard deviation for the mean of each of the centroids in both the vertical Y and horizontal X direction are small, between 15 and 43 thousands of a pixel and are close to each other (relative magnitude). It can be seen from Table 9-1 that the deviations in the Y direction are always larger than the X direction. This is attributed to the pose of the object when the pose image is taken, because the object is tilted by some amount along the X_o axis the optical targets will always appear to be flattened. With the height being less any errors estimating the boundary near the top and bottom are exaggerated compared to any errors on the left and right, this leads to less consistent vertical position results, creating a larger standard deviation.

The object pose estimate using the information from Table 9-1 shows that when the individual target centroids were added together their uncertainty decreases, this can be seen in the $X_{translation}$ and $Y_{translation}$ pose estimates. The better estimate of optical target positions in the X direction can be seen in the object pose as well. In the $Z_{translation}$ case the errors of each optical target will combine creating a worse estimate. Using the static data, without the information gained from the model, for the $X_{translation}$ and $Y_{translation}$ the $Std_{new} = \frac{1}{\sum \frac{1}{Std_i}}$, is approximately 0.004 for $X_{translation}$ and 0.007 for $Y_{translation}$, summing the deviations $Std_{new} = \sum Std_i$ gives 0.237 for $Z_{translation}$, this shows that the measured trends match the expected results.

The rotation Standard deviations were similar to the translation vectors, this time the better estimate of $\theta_{Y_{rotation}}$ compared to $\theta_{X_{rotation}}$ is attributed to the better estimate of $X_{position}$ because rotation around the Y_o axis changes the optical targets width(X), and rotations around the X_o axis change the optical targets height(Y). The $\theta_{Z_{rotation}}$ standard deviation is similar to the other two because all targets are reducing the uncertainty in rotation.

9.2 Robot Repeatability Test Results

A robot repeatability test was performed with the rectangular block using the procedure presented in Section 8.5.3. The resulting histograms are shown in Figure 9-3.

9.2.1 *Robot Repeatability Test Results Data*

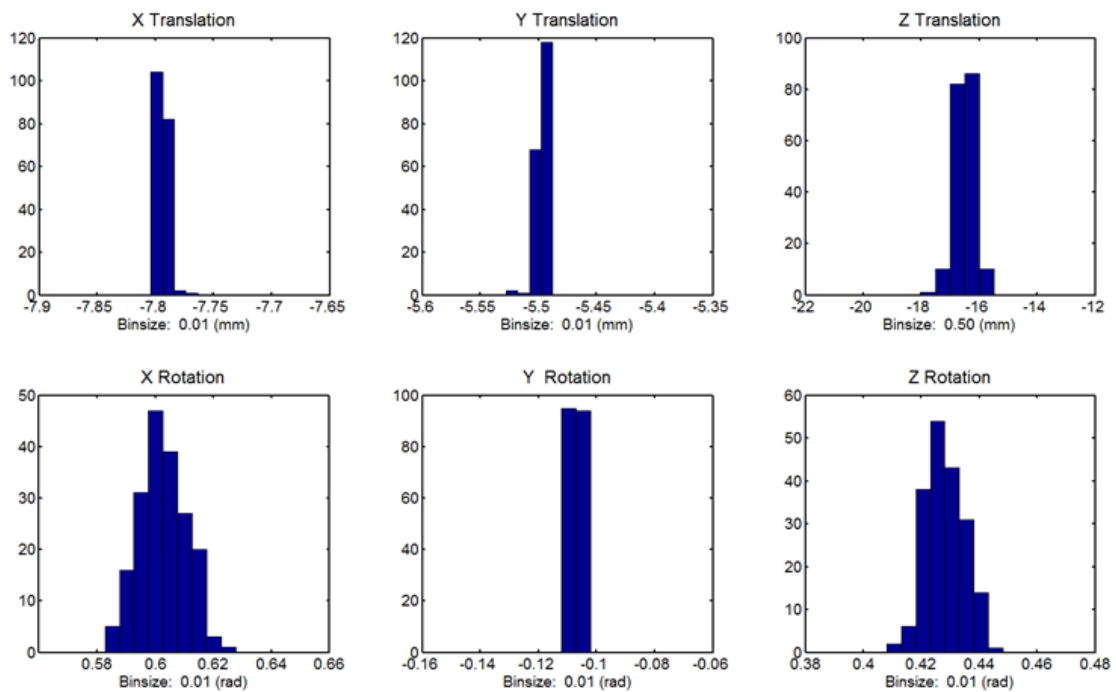


Figure 9-3 Histograms of the robot repeatability from the robot repeatability test.

Table 9-3 Means and standard deviations of the object pose from the robot repeatability test.

Variable	Mean	Standard Deviation
$X_{translation} (mm)$	-7.794	0.004
$Y_{translation} (mm)$	-5.498	0.003
$Z_{translation} (mm)$	-16.498	0.318
$\theta_{Xrotation} (rad)$	0.603	0.008
$\theta_{Yrotation} (rad)$	-0.107	0.001
$\theta_{Zrotation} (rad)$	0.428	0.006

9.2.2 Robot Repeatability Test Results Discussion

The means of the Robot Repeatability Test and the Stationary robot test are almost identical, they are in the worst case of $Z_{translation}$ within $0.1mm$. The important difference is that the Standard Deviation almost doubles with respect to each measured and calculated value. Each value, with the exception of $Z_{translation}$ is still very small, and hence should not negatively affect our results.

9.3 Grasping Results for Grasp Repeatability Testing

The following results are for the Grasp Repeatability Object presented in Section 7.3.1.

9.3.1 ***Robot Grasping Rectangular Box Initially on Flat Tabletop***

This test was performed using the procedure described in sections 8.4.1 and 8.4.3. The only variations in the object's initial pose were along X_w and Y_w , and about Z_w . The results are presented in Figure 9-4 and Table 9-4.

9.3.1.1 ***Robot Grasping Rectangular Box Initially on Flat Tabletop Data***

Presented below is the histogram of results as well as the mean and standard deviation of the measured values.

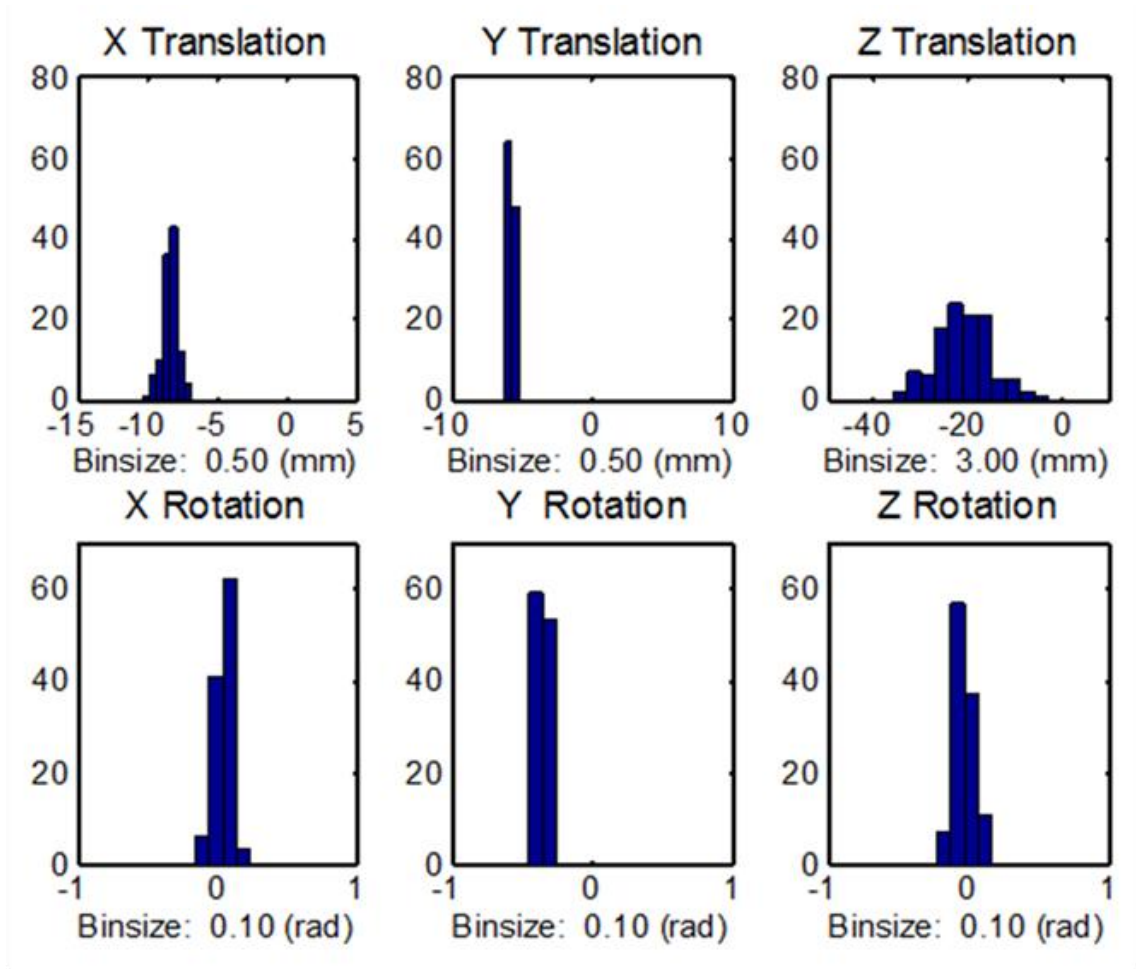


Figure 9-4 Histograms of the pose errors from the grasp repeatability test with the rectangular box object initially on the flat tabletop.

Table 9-4 Means and standard deviations of the pose errors from the grasp repeatability test with the rectangular box object initially on the flat tabletop.

Variable	Mean	Standard Deviation
$X_{translation} (mm)$	-8.324	0.571
$Y_{translation} (mm)$	-5.739	0.061
$Z_{translation} (mm)$	-20.937	5.850

$\theta_{x_{rotation}} (rad)$	0.048	0.048
$\theta_{y_{rotation}} (rad)$	-0.345	0.013
$\theta_{z_{rotation}} (rad)$	-0.027	0.064

9.3.1.2 *Robot Grasping Rectangular Box Initially on Flat Tabletop Discussion*

With the object always resting flat on the tabletop the object model is more accurate, as the laser scanner can model a box by itself accurately. With a good model the amount of rotations about the X_g , and Y_g axis are minimal. The grasp planner is focused on finding the correct X_w, Y_w, θ_{z_w} although it does not know that the object is resting flat so it is still computing the other three pose variables ($Z_w, \theta_{x_w}, \theta_{y_w}$). Comparing the relative standard deviation of the translations we see that the $Z_{translation}$ is the least consistent, this makes sense as it is the least constrained of all of the grasping axis, the $X_{translation}$ and $Y_{translation}$ are based on the size and shape of the largest facet, where as the $Z_{translation}$ is based $P_{opposite}$ of which there are many more choices. The most consistent translation ($Y_{translation}$) occurs because the object is always located within/by the parallel jaws when they close. If we analyze the rotations the object was always located flat on the tabletop and our gripping algorithm does not create any rotation along the X_g axis see Figure 6-46 and rotation along X_g causes a change in the rotation along Y_o . This leaves $\theta_{x_{rotation}}$ and $\theta_{z_{rotation}}$ and both are constrained by the gripper jaws in a similar manner, helping to rotate the object to a consistent position.

9.3.2 *Robot Grasping Rectangular Block Initially on Playfield*

This test was performed using the procedure described in sections 8.5.2 and 8.4.3. The variations in the objects pose could include all 6 DOF. The results are presented in Figure 9-5 and Table 9-5.

9.3.2.1 *Robot Grasping Rectangular Box Initially on Playfield Data*

Presented below is the histogram of results as well as the mean and standard deviation of the measured values.

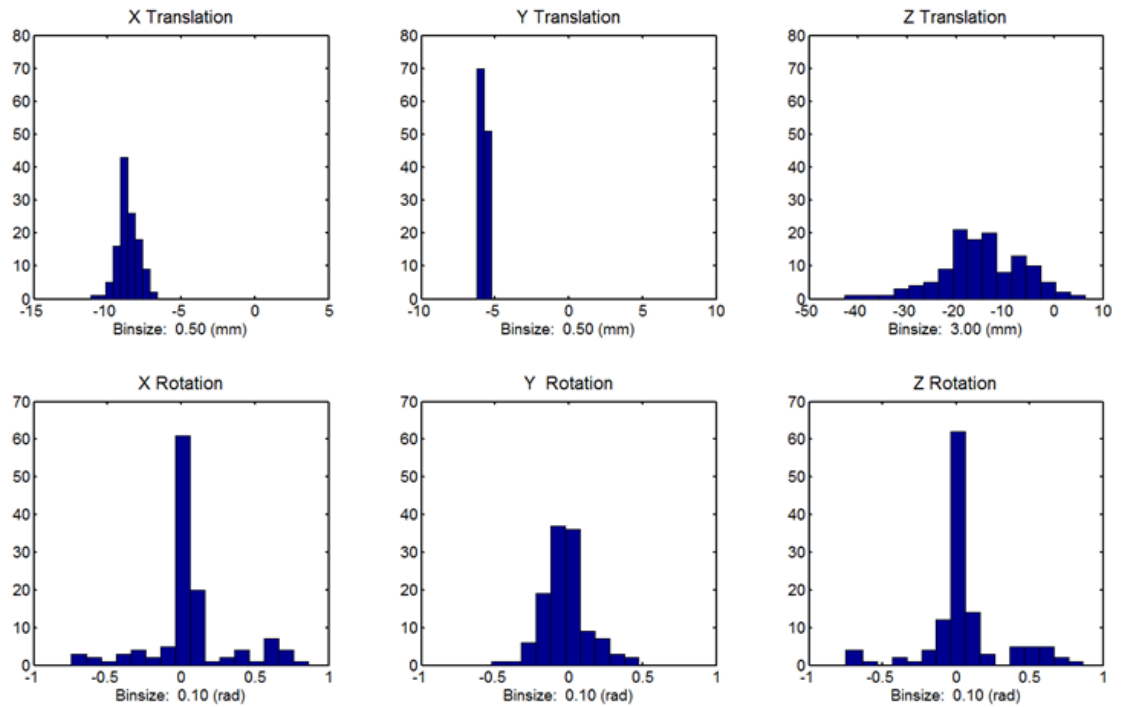


Figure 9-5 Histogram of the pose errors from the grasp repeatability test with the rectangular box on the playfield.

Table 9-5 Means and standard deviations of the pose errors from the grasp repeatability test with the rectangular box object initially on the playfield.

Variable	Mean	Standard Deviation
$X_{translation} (mm)$	-8.642	0.692
$Y_{translation} (mm)$	-5.720	0.051
$Z_{translation} (mm)$	-14.669	8.196
$\theta_{x_{rotation}} (rad)$	0.062	0.286
$\theta_{y_{rotation}} (rad)$	0.024	0.148
$\theta_{z_{rotation}} (rad)$	0.066	0.265

9.3.2.2 *Robot Grasping Rectangular Box Initially on Playfield Discussion*

The playfield data is similar to the Flat Tabletop results in terms of why some standard deviations like $Z_{translation}$ is larger, why $Y_{translation}$ is smaller. The $X_{translation}$ has increased because on an angle the horizontal size (relative to the X_w) axis has decreased and there are now more options for facet selection. The model of the object also now includes rotations along all axis, and is more erroneous then the previous experiment. The $\theta_{x_{rotation}}$ now also includes grasps where the object and the grasp planner were far enough apart that the friction from rubber pads on the gripper jaws held the object at an angle. $\theta_{y_{rotation}}$ accounts for tilting of the object across its length(X_o), this was only randomized by the object coming to rest on the playfield, and as before not adjusted by

the grasp planner. The standard deviation for the $\theta_{x_{rotation}}$ and $\theta_{z_{rotation}}$ has quadrupled from the static test, this can easily be seen by observing both histograms, Figure 9-4 and Figure 9-5. There are a number rotations in the playfield case that lie outside of the main peak. The distribution of these points is also more uniform than normal this could be attributed to the grasp repeatability object aligning itself with the parallel jaws when the difference in the angle between the sides of the object and the gripper jaws is low. In this case as the offset moves away from the mean its occurrence probability would drop but the probability of it aligning its sides with the gripper jaws would also drop. A normal grasp and some of these extraneous cases can be seen in Figure 9-8.

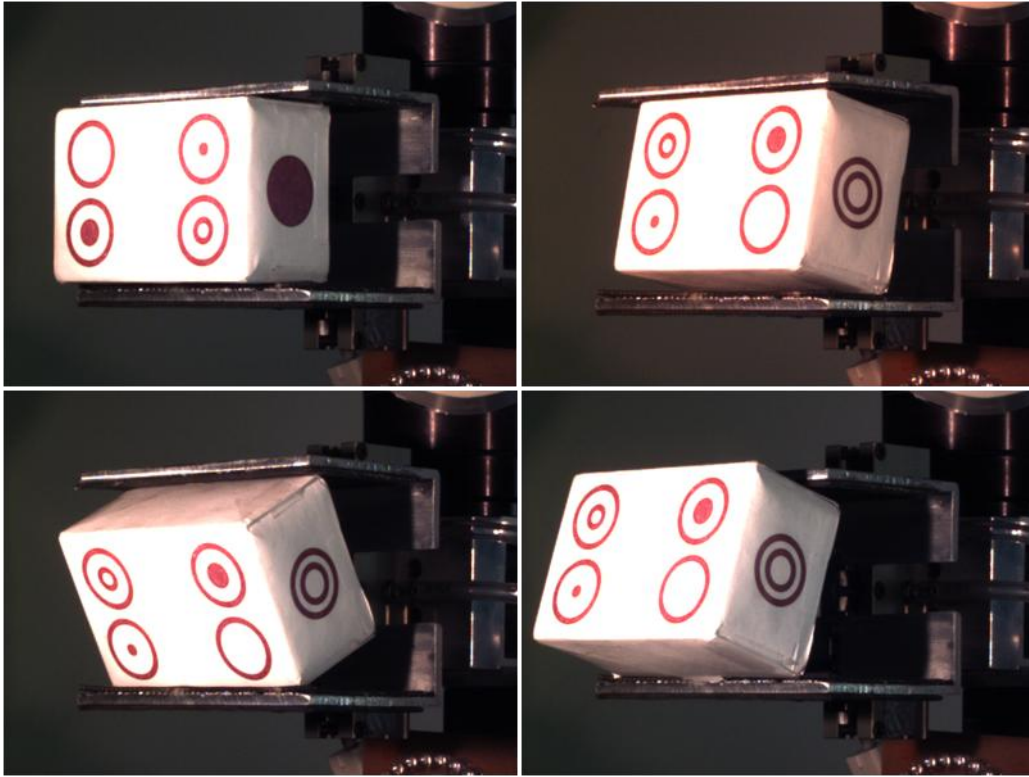


Figure 9-6 Top Left: Standard Grasp, Top Right: Small amount of offset along Y_o , Bottom Left, and Bottom Right: Large amount of offset along Y_o .

9.4 Grasping Results for Graspability Testing

Graspability tests were performed for all 4 graspability objects plus the rectangular block grasp repeatability object using the procedure described in sections 8.5.2 to 8.43. The results are summarized in Table 9-6.

9.4.1 *Robot Graspability Testing Data*

Table 9-6 Object graspability results

Object	Grasp Attempts	Grasp Successes	Success Percentage
Rectangular block	127	125	98.43%

on flat surface			
Rectangular block on playfield	137	128	80.30%
T-shaped block on playfield	152	148	97.37%
L-shaped block on playfield	110	106	96.36%
Cube on playfield	204	198	97.06%
Hexagonal nut on playfield	173	169	97.69%

9.4.2 *Robot Graspability Testing Discussion*

The grasp experiments performed for all of the objects were deemed successful. The most successful grasp object was the Rectangular block on flat surface, it was also the only time the playfield was not used, for a specific discussion about why it was better refer to 9.3.1. The worst object was the rectangular block on the playfield. This time the robot was forced to make an estimate of the objects θ_{x_o} and θ_{y_o} with a model that had more error. The gripper jaws had a maximum opening of $66mm$ with a resolution from the laser scanner (that can only affect the top of the object) of $1mm$ and the object was $57mm$ wide leaving a maximum of a possible $4mm$ for the gripper model, and grasp planner. The other test objects in graspability set were slightly smaller (from $50mm$ – $52mm$) and hence gave a little more room for grasp planner and modeling error.

Comparing the T shaped block and the L shaped block it can be seen that the T Shaped Block was able to be grasped 1% more often than the L shaped block. This initially feels unintuitive since the L Shaped Block can be grasped in more orientations than the T

Shaped Block, the L Shaped Block had 6 unique grasps compared to the 2 for the T Shaped Block. However the T Shaped Block, unlike the L Shaped Block has one side that is much larger than the others, this creates a natural location for the Grasping plane because after the convex hull is taken the size of the facet is much larger than any other in the model. This removal of ambiguity from the L Shaped Block could have given it the consistency needed for more consistent grasps.

Note that for the Hexagonal Nut, is that after grasp attempt 134 failed during the automated test the next grasp attempt with the object in the same pose was successful without any complications, although visually similar the models of the object were found to have slight differences on the estimate of the size of the nut, the successful grasp estimated the distance between jaws would be $1.2mm$ further apart than the failed grasp.

9.5 Timing information

Timing information was collected from all grasp attempts for the processing and grasping of the hexagonal nut object. The timing information for the computation of the grasp is shown in Figure 9-7. The detailed set of timing information for the hexagonal nut is shown in Figure 9-8, where:

9.5.1 *Timing data for the Hexagonal Nut*

- “imageTime” was the time it took to capture all of the images.

- “modelTime” was the time it took to create the model after silhouettes had been processed.
- “graspFlatFaceTime” was the time it took the grasp planner from [16] to create a grasp, if it was able to.
- “graspConvexTime” was the time the convex hull grasp planner developed in this thesis took to create a grasp.
- “graspPoseTime” was the time it took to move to the pose position and compute the pose of the object.
- “LaserScanTime” was the time it took to capture and process the laser scan.
- “LaserModificationTime” was the time it took to modify the model based on the laser scan.

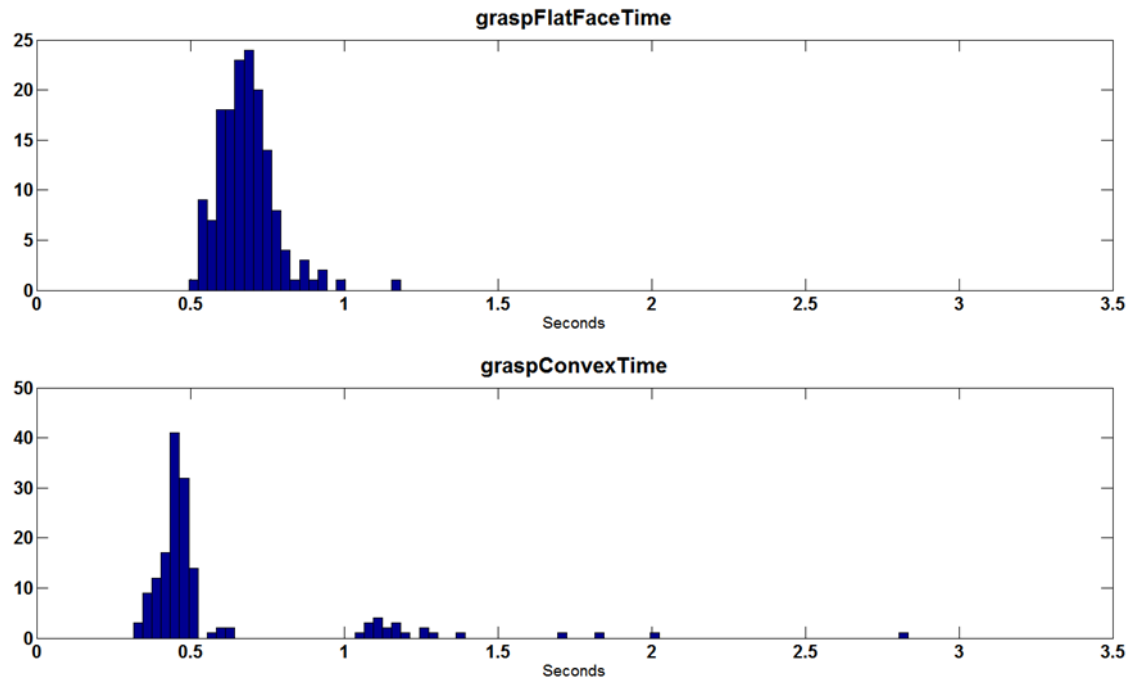


Figure 9-7 Histograms of the time the algorithm presented in [16] (top) and the time the algorithm presented in this thesis (bottom) required to complete the grasp planning for the hexagonal nut object.

Table 9-7 Means and Standard Deviations for the two histograms presented above.

Variable	Mean (Seconds)	Standard Deviation
graspFlatFaceTime	0.687	0.092
graspConvexTime	0.574	0.349

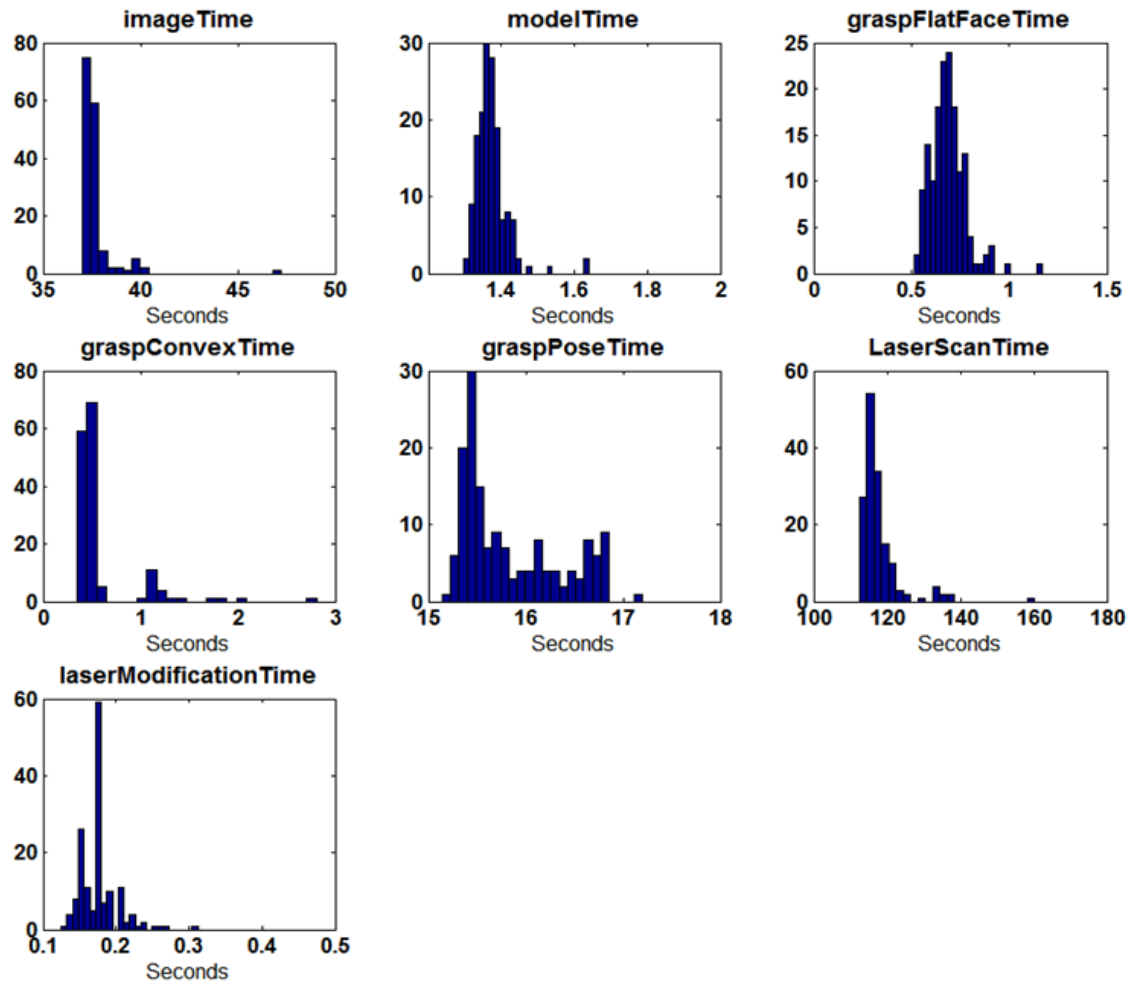


Figure 9-8 Histograms of the times the algorithm required to complete each step for the hexagonal nut object.

Table 9-8 Means and standard deviations of the times it took to complete each of the actions during the grasping routine.

Variable	Description	Mean (s)	Standard Deviation (s)
imageTime	Time taken to capture input images	37.67	0.974
modelTime	Time taken to create the model from the images	1.37	0.048
graspFlatFaceTime	Time taken for existing algorithm to compute a grasp	0.69	0.092
graspConvexTime	Time taken for algorithm presented in this thesis to compute a grasp	0.57	0.349
graspPoseTime	Time taken to capture the pose image	15.84	0.510
LaserScanTime	Time taken for laser scan and processing	117.94	5.900
LaserModificationTime	Time taken to refine the model based on the laser scan	0.18	0.026

9.5.2 *Timing data discussion for the Hexagonal Nut*

The most valuable time comparison is the comparison of the grasp generation developed in this thesis and the one previously completed in [16]. It can be seen that the previous grasp planner completed all of its computations in the 0.5s to 1s range, while the grasp planner developed in this thesis completed the majority of its grasps in under 0.5s. The real difference is the distribution of the two different grasp planners. The standard deviation of the new grasp planner was over three and a half times the existing planner. As previously stated the existing grasp planner times spanned a range of just only one half of a second and consisted of single peak. The grasp planner in developed in this thesis has at least two and potentially up to four distinct regions

centered approximately around (0.062s, 1.12s, 1.75s, 2.8s). The largest set, around 0.62s can be attributed to the convex hull algorithm reducing the complexity of the object. It was found that the first peak correlated to the object resting "flatter" (on its larger face) The objects were reduce down to around 44 facets for anaysis. The second, third and fourth have the object tilted more, or resting on its side this created more facets that had to be analyzed in the grasp planner. The worst grasp in terms of time (over 2.8s) had 66 facets that needed to be analyzed.

9.6 Discussion

The grasp algorithm developed in this thesis was tested extensively, it was found that optical targets were sufficient to track the pose of the grasp repeatability object and that the algorithm was able to successfully grasp a number of graspability objects with a high success rate. It was shown that the accuracy of the model created for grasping is critical in terms of grasp repeatability and overall graspability. Finally the size and pose of the object to be grasped were shown to significantly affect the convex grasp computational time.

Chapter 10

Conclusions and Recommendations

10.1 Summary and Key Achievements.

This thesis presented a method for grasping unmolded unknown objects using vision. The results presented show that the grasping algorithm that was developed is very repeatable for a variety of objects. The thesis was broken down into two main sections: grasp planning and grasp evaluation. Experimental results of grasp evaluation and of the combined evaluation and grasp system were presented. Grasp computational time data from a previous competing version of the planner compared to the presented version was also presented.

Key achievements include:

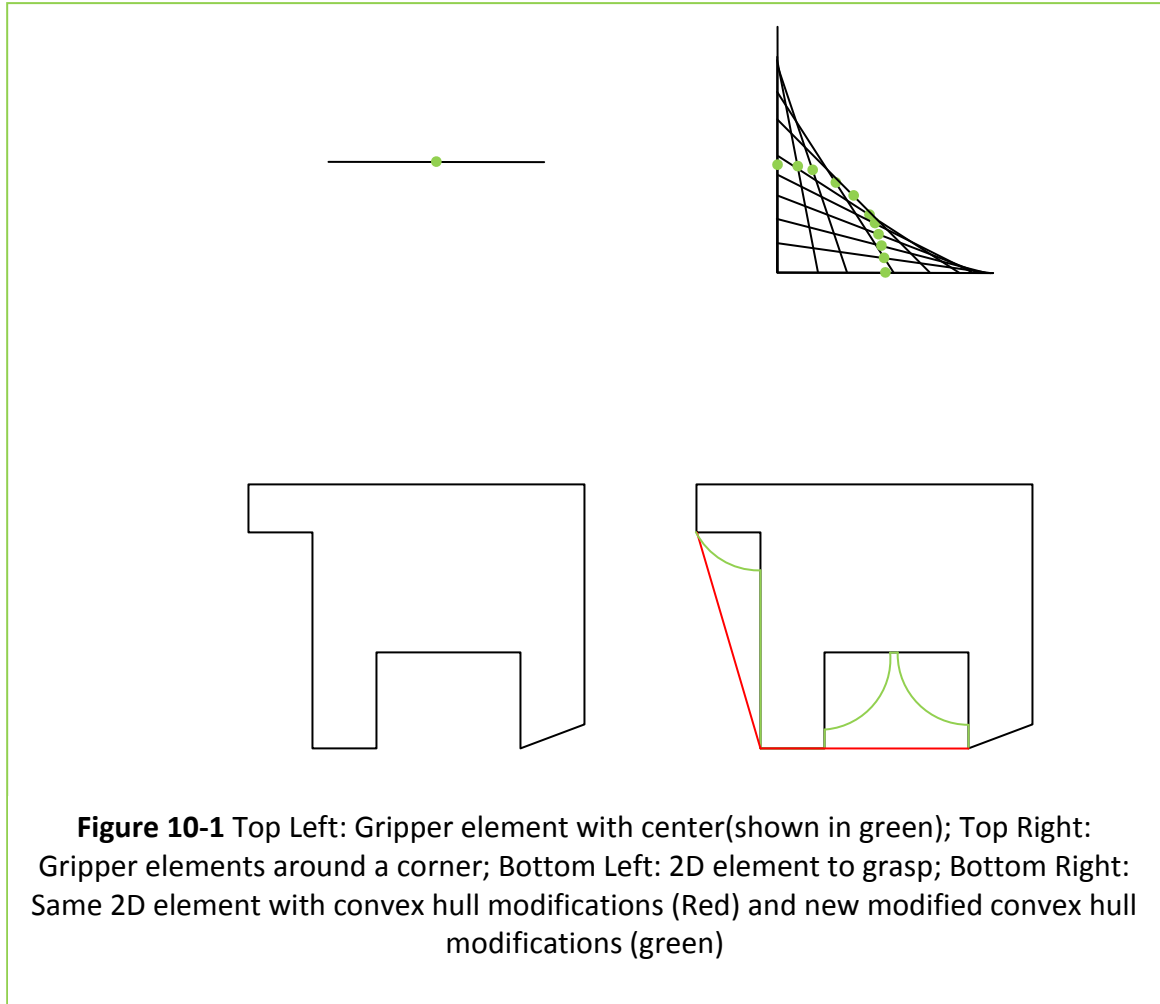
- The developed grasping algorithm was able to grasp a variety of untaught unknown objects. The algorithm also did not require previous grasp training.
- In the best case the system was able to successfully grasp a Hexagonal Nut with a 97.69% success rate over 173 attempts.
- In the worst case the system was able to successfully grasp a Rectangular Block On Playfield with a 80.30% success rate over 137 attempts.
- The system was able to grasp objects in any pose possible while they were lying on the playfield.

- The average grasp time from the existing algorithm presented by [16] was reduced by $0.1s$ to $0.57s$ and the median grasp time was reduced by $0.23s$ to $0.45s$.
- A evaluation method involving two distinct sets of objects was designed, developed, tested, and presented.
- With respect to the most relevant, in terms of being actively fed and modified by the grasp planner, components of the pose($X_{translation}, Y_{translation}, Z_{translation}, \theta_{X_{Rotation}}, \theta_{Z_{Rotation}}$) the ratio of the size of the distribution of the points (6 standard deviations) in the X_o direction to the length of the object is 5.0% , in the Y_o direction to the width is 0.5% and the standard deviations on the X and Y rotations were under a spread of 0.286 radians.

10.2 Future Work

Future work in this area of research can include extensions on the use of the convex hull. From the definition of the convex hull, its edges have no limit on size, the parallel jaw gripper also was assumed to be larger than the object. If these are changed such that the parallel jaws are not assumed to be larger, so that they have a finite length, and the convex hull edges are changed so that they also have the same max length then when the convex hull is taken it is no longer bound to just the exterior edges, but includes features that are larger than the gripper jaws. In this way, additional candidate grasps

will be generated. These grasps would have been eliminated from earlier trials if the convex hull was not modified. This idea is shown in 2D in Figure 10-1.



Another option for future work would be designing a custom gripper, since the main focus of this algorithm is on defining a plane and point for grasping, gripper jaws that include a single plane on one side, and a smaller adjustable point for the other side could be constructed. This gripper during, and after grasping would give the best view of

the object being grasped to a machine vision system that could be used to classify and potentially sort the object.

Modifications on gripper jaw opening and closing distance can also be attempted. The algorithm has been tested with objects that can fit inside and be grasped by the gripper, approximately 42mm to 70mm. With the addition of a larger or smaller gripper and/or larger or smaller articulated arm, many different types and sizes of objects could be grasped.

Image processing and grasping time could be optimized through the use of hardware accelerators or other embedded devices. The image processing, for example, could be done on a Field Programmable Gate Array (FPGA) or a Graphics Processing Unit (GPU), using Compute Unified Device Architecture (CUDA).

A stereo vision system such as Bumblebee stereo camera or Microsoft Kinect could replace the Dragonfly2 camera to create a better 3D estimate of the object, alternatively the current setup could be used with a modification of the image capture routine to use the laser scanner in more than one plane.

The system could be expanded to work with many objects at once, where each object would be grasped independently after some type of segmentation operation. The grasping order, type of grasp, and other constraints could be fed into the grasping

planner. This could for example simulate clearing a mine shaft of debris after blasting had occurred.

References

- [1] G. Webster, "Strong Robotic Arm Extends From Next Mars Rover," Jet Propulsion Laboratory, 16 09 2010. [Online]. Available: <http://mars.jpl.nasa.gov/msl/news/index.cfm?FuseAction=ShowNews&NewsID=1056>. [Accessed 05 01 2013].
- [2] ABB, "IRB 360," ABB, 2013. [Online]. Available: <http://www.abb.ca/product/seitp327/cf1b0a0847a71711c12573f40037d5cf.aspx>. [Accessed 12 2012].
- [3] rethink robotics, "Baxter Research Robot," rethink robotics, 2012. [Online]. Available: http://www.rethinkrobotics.com/index.php/products/baxter_research/. [Accessed 03 2013].
- [4] rethink robotics, "Baxter Robot: Conveyor Task with Two Arms," YouTube, 11 10 2012. [Online]. Available: <http://www.youtube.com/watch?v=6Eko3zw0eqs>. [Accessed 07 03 2013].
- [5] cognex, "A 3D Machine Vision Tool for VisionPro and CVL," Cognex Corporation,

2011. [Online]. Available: <http://www.cognex.com/3d-machine-vision.aspx?pageid=11502&langtype=1033>. [Accessed 2012].
- [6] S. Stansfield, "Robotic grasping of unknown objects: A knowledge-based approach," *The International journal of robotics research*, vol. 10, no. 4, pp. 314-326, 1991.
- [7] M. Taylor, A. Blake and A. Cox, "Visually guided grasping in 3D," in *Proc. Conf. IEEE Int Robotics and Automation*, 1994.
- [8] A. Bendiksen and G. Hager, "A vision-based grasping system for unfamiliar planar objects," in *Proc. Conf. IEEE Int Robotics and Automation*, 1994.
- [9] C. L. Jackins and S. L. Tanimoto, "Oct-trees and their use in representing three-dimensional objects," *Computer Graphics and Image Processing*, vol. 14, no. 3, pp. 249-270, 1980.
- [10] C. Bard, C. Laugier, C. Milési-Bellier, J. Troccaz, A. F. Viallet, B. Triggs and G. Vercelli, *Achieving Dextrous Grasping By Integrating Planning and Vision Based Sensing*, 1995.
- [11] M. Trobina and A. Leonardis, "Grasping arbitrarily shaped 3-D objects from a pile," in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International*

Conference on, 1995.

- [12] C. Borst, M. Fischer and G. Hirzinger, "A fast and robust grasp planner for arbitrary 3D objects," in *Proc. IEEE Int Robotics and Automation Conf*, 1999.
- [13] P. J. Sanz, A. Requena, J. M. Inesta and A. P. Del Pobil, "Grasping the not-so-obvious: vision-based object handling for industrial applications," *IEEE Robotics \& Automation Magazine*, vol. 12, no. 3, pp. 44-52, 2005.
- [14] A. T. Miller, S. Knoop, H. I. Christensen and P. K. Allen, "Automatic grasp planning using shape primitives," in *Proc. IEEE Int. Conf. Robotics and Automation ICRA '03*, 2003.
- [15] A. T. Miller and P. K. Allen, "Graspit! A versatile simulator for robotic grasping," *IEEE Robotics \& Automation Magazine*, vol. 11, no. 4, pp. 110-122, 2004.
- [16] G. M. Bone, A. Lambert and M. Edwards, "Automated modeling and robotic grasping of unknown three-dimensional objects," in *Proc. IEEE Int. Conf. Robotics and Automation ICRA 2008*, 2008.
- [17] M. Richtsfeld and M. Vincze, "Grasping of Unknown Objects from a Table Top," in *Workshop on Vision in Action: Efficient strategies for cognitive agents in complex environments*, Marseille, France, 2008.

- [18] A. Saxena, J. Driemeyer and A. Y. Ng, "Robotic Grasping of Novel Objects using Vision," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157-173, 2008.
- [19] K. Huebner and D. Kragic, "Selection of robot pre-grasps using box-based shape approximation," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems IROS 2008*, 2008.
- [20] C. Goldfeder, M. Ciocarlie, J. Peretzman, H. Dang and P. K. Allen, "Data-driven grasping with partial sensor data," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems IROS 2009*, 2009.
- [21] R. Ohbuchi, K. Osada, T. Furuya and T. Banno, "Salient local visual features for shape-based 3D model retrieval," in *Proc. IEEE Int. Conf. Shape Modeling and Applications SMI 2008*, 2008.
- [22] M. Popovic, D. Kraft, L. Bodenhagen, E. Baseski, N. Pugeault, D. Kragic, T. Asfour and N. Krüger, "A strategy for grasping unknown objects based on co-planarity and colour information," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 551-565, 2010.
- [23] L. Bodenhagen, D. Kraft, M. Popovic, E. Baseski, P. E. Hotz and N. Kruger, "Learning to grasp unknown objects based on 3D edge information," in *Proc. IEEE*

Int Computational Intelligence in Robotics and Automation (CIRA) Symp, 2009.

- [24] J. Bohg, M. Johnson-Roberson, B. Leon, J. Felip, X. Gratal, N. Bergstrom, D. Kragic and A. Morales, "Mind the gap - robotic grasping under incomplete observation," in *Proc. IEEE Int Robotics and Automation (ICRA) Conf*, 2011.
- [25] S. Dragiev, M. Toussaint and M. Gienger, "Gaussian process implicit surfaces for shape estimation and grasping," in *Proc. IEEE Int Robotics and Automation (ICRA) Conf*, 2011.
- [26] Y. Jiang, S. Moseson and A. Saxena, "Efficient grasping from RGBD images: Learning using a new rectangle representation," in *Proc. IEEE Int Robotics and Automation (ICRA) Conf*, 2011.
- [27] J. Aleotti and S. Caselli, "Part-based robot grasp planning from human demonstration," in *Proc. IEEE Int Robotics and Automation (ICRA) Conf*, 2011.
- [28] S. Biasotti, D. Giorgi, M. Spagnuolo and B. Falcidieno, "Reeb graphs for shape analysis and applications," *Theoretical Computer Science*, vol. 392, no. 1-3, pp. 5-22, 2008.
- [29] S. Berretti, A. D. Bimbo and P. Pala, "3D Mesh decomposition using Reeb graphs," *Image and Vision Computing*, vol. 27, no. 10, pp. 1540-1554, 2009.

- [30] A. M. Dollar and R. D. Howe, "Towards grasping in unstructured environments: optimization of grasper compliance and configuration," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS 2003)*, 2003.
- [31] A. M. Dollar and R. D. Howe, "Simple, Robust Autonomous Grasping in Unstructured Environments," in *Proc. IEEE Int Robotics and Automation Conf*, 2007.
- [32] J. Su, H. Qiao and C. Liu, "A vision-based 3D grasp planning approach with one single image," in *Proc. Int. Conf. Mechatronics and Automation ICMA 2009*, 2009.
- [33] G. Bradski, "{The OpenCV Library}," *Dr. Dobb's Journal of Software Tools*, 2000.
- [34] L. S. D. Daniel F. DeMenthon, *Model-Based Object Pose in 25 Lines of Code*, 1995.
- [35] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *#IEEE_J_JRA#*, vol. 3, no. 4, pp. 323-344, 1987.
- [36] R. Willson, "Tsai Calibration C Code," [Online]. Available: <http://www-cgi.cs.cmu.edu/afs/cs.cmu.edu/user/rgw/www/TsaiCode.html>.
- [37] C. R. Dyer, "Volumetric Scene Reconstruction From Multiple Views," in

Foundations of Image Understanding, 2001.

- [38] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Upper Saddle River, New Jersey: Pearson Prentice Hall, 2008.
- [39] U. H. Wolfgang Niem, *Robust and fast modelling of 3D natural objects from multiple views*, 1994.
- [40] J. O'Rourke, "Computational Geometry in C," in *Computational Geometry in C*, Cambridge (NY), Cambridge University Press, 1998, p. 72.
- [41] D. R. Chand and S. S. Kapur, "An Algorithm for Convex Polytopes," *J. ACM*, vol. 17, no. 1, pp. 78-86, #jan# 1970.
- [42] C. B. Barber, D. P. Dobkin and H. Huhdanpaa, *The Quickhull algorithm for convex hulls*, 1996.
- [43] A. Bicchi, "On the Closure Properties of Robotic Grasping," *The International Journal of Robotics Research*, vol. 14, no. 4, pp. 319-334, 1995.
- [44] B. DizioÄŸlu and K. Lakshiminarayana, "Mechanics of form closure," *Acta Mechanica*, vol. 52, pp. 107-118, 1984.
- [45] Thermo CRS, Ltd., F3 Robot System User Guide, Burlington: Thermo CRS, Ltd,

2002.

- [46] P. Bose, D. Bremner and G. Toussaint, *All convex polyhedra can be clamped with parallel jaw grippers*, 1994.
- [47] V.-D. Nguyen, "Constructing force-closure grasps," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1986.
- [48] V.-D. Nguyen, "Constructing force-closure grasps in 3D," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1987.
- [49] A. M. Dollar and R. D. Howe, "Designing robust robotic graspers for unstructured environments," in *proceedings of the Workshop on Manipulation for Human Environments 2006 Robotics Science and Systems Conference*, Citeseer, 2006.
- [50] F. Ababsa and M. Mallem, "A Robust Circular Fiducial Detection Technique and Real-Time 3D Camera Tracking," *Journal of Multimedia*, vol. 3, no. 4, 2008.
- [51] W. Zhang, L. Wang, L. Hao, Q. Chen and D. Du, "An indirect style linkage under-actuated humanoid robot hand," in *Proc. 2nd Int Informatics in Control, Automation and Robotics (CAR) Asia Conf*, 2010.
- [52] G.-L. Zhang, Y. Wang and C. W. de Silva, "Knowledge-based grasping of unknown objects in unstructured urban environments," in *Proc. IEEE Int. Conf. Automation*

and Logistics ICAL 2008, 2008.

- [53] J. Sin and H. Stephanou, "Semi-distributed manipulation on a friction force field," in *Proc. IEEE/RSJ Int Intelligent Robots and Systems Conf*, 2001.
- [54] Y. Song, M. Li, Q. Li and L. Sun, "A Visual Guided 4 D.O.F Miniature Robot System for Micro-Handling Purpose," in *Proc. IEEE Int. Conf. Robotics and Biomimetics ROBOTICS '06*, 2006.
- [55] D. Sun and J. K. Mills, "Manipulating rigid payloads with multiple robots using compliant grippers," *#IEEE_J_MECH#*, vol. 7, no. 1, pp. 23-34, 2002.
- [56] T. Tomizawa, K. Ohba, A. Ohya and S. Yuta, "Remote Food Shopping Robot System in a Supermarket -Realization of the shopping task from remote places," in *Proc. Int. Conf. Mechatronics and Automation ICMA 2007*, 2007.
- [57] A. M. Zaki, A. M. Soliman, O. A. Mahgoub and A. M. El-Shafei, "Design and implementation of efficient intelligent robotic gripper," in *Proc. Int Modelling, Identification and Control (ICMIC) Conf*, 2010.
- [58] H. Ahmadi and M. J. Sadigh, "Safe grasping in presence of multi phase friction," in *Proc. 10th Int. Conf. Control, Automation, Robotics and Vision ICARCV 2008*, 2008.

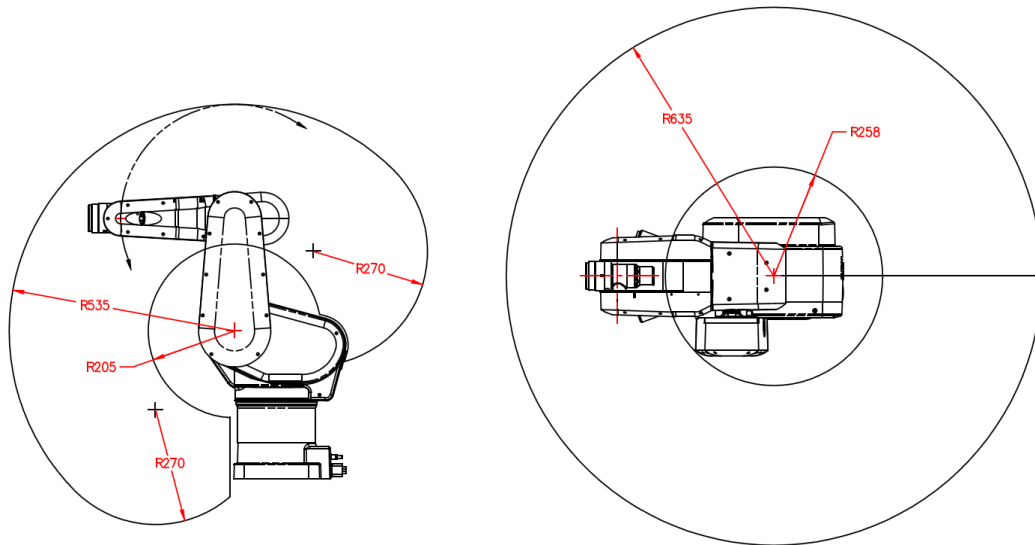
- [59] H. Bay, A. Ess, T. Tuytelaars and L. V. Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346-359, 2008.
- [60] N. Bergstrom and D. Kragic, "Partial 3D Reconstruction of Objects for Early Reactive Grasping.," 2009.
- [61] C. Borst, M. Fischer and G. Hirzinger, "Grasping the dice by dicing the grasp," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS 2003)*, 2003.
- [62] D. Ding, Y.-H. Liu, J. Zhang and A. Knoll, "Computation of fingertip positions for a form-closure grasp," in *Proc. ICRA Robotics and Automation IEEE Int. Conf*, 2001.
- [63] A. M. Dollar and R. D. Howe, "Simple, reliable robotic grasping for human environments," in *Proc. IEEE Int. Conf. Technologies for Practical Robot Applications TePRA 2008*, 2008.
- [64] A. M. Dollar and R. D. Howe, "Joint coupling and actuation design of underactuated hands for unstructured environments.," in *IEEE Transactions on Robotics*, 2006.
- [65] S. Ekvall and D. Kragic, "Learning and Evaluation of the Approach Vector for Automatic Grasp Generation and Planning," in *Proc. IEEE Int Robotics and Automation Conf*, 2007.

- [66] E. G. M. Holweg, H. Hoeve, W. Jongkind, L. Marconi, C. Melchiorri and C. Bonivento, "Slip detection by tactile sensors: algorithms and experimental results," in *Proc. Conf. IEEE Int Robotics and Automation*, 1996.
- [67] K. Huebner, S. Ruthotto and D. Kragic, "Minimum volume bounding box decomposition for shape approximation in robot grasping," in *Proc. IEEE Int. Conf. Robotics and Automation ICRA 2008*, 2008.
- [68] K. Koganezawa and Y. Ishizuka, "Novel mechanism of artificial finger using double planetary gear system," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems IROS 2008*, 2008.
- [69] R. Kolluru, K. P. Valavanis, S. S. Smith and N. Tsourveloudis, "Design fundamentals of a reconfigurable robotic gripper system," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 30, no. 2, pp. 181-187, 2000.
- [70] R. Kouskouridas, A. Amanatiadis and A. Gasteratos, "Guiding a robotic gripper by visual feedback for object manipulation tasks," in *Proc. IEEE Int Mechatronics (ICM) Conf*, 2011.
- [71] S. M. LaValle, "Rapidly-Exploring Random Trees: A New Tool for Path Planning," *In*, vol. 129, no. 98-11, pp. 98-11, 1998.

- [72] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91-110, 2004.
- [73] W. N. Martin and J. K. Aggarwal, "Volumetric Descriptions of Objects from Multiple Views," *#IEEE_J_PAMI#*, no. 2, pp. 150-158, 1983.
- [74] Z. Marton, D. Pangercic, N. Blodow, J. Kleinhellefort and M. Beetz, "General 3D modelling of novel objects from a single view," in *Proc. IEEE/RSJ Int Intelligent Robots and Systems (IROS) Conf*, 2010.
- [75] A. Miller, P. Allen, V. Santos and F. V. cuevas, "From Robot Hands to Human Hands: A Visualization and Simulation Engine for Grasping Research," *Industrial Robot*, vol. 32, pp. 55-63, 2005.
- [76] K. Morimoto, Y. Tada, H. Takashima, K. Minamino, R. Tahara and S. Konishi, "Design and characterization of high-performance contactless gripper using spiral air flows," in *Proc. Int Micro-NanoMechatronics and Human Science (MHS) Symp*, 2010.
- [77] R. K. R. Jain and B. G. Schunk, *Machine Vision*, McGraw-Hill, 1995.
- [78] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. Third Int 3-D Digital Imaging and Modeling Conf*, 2001.

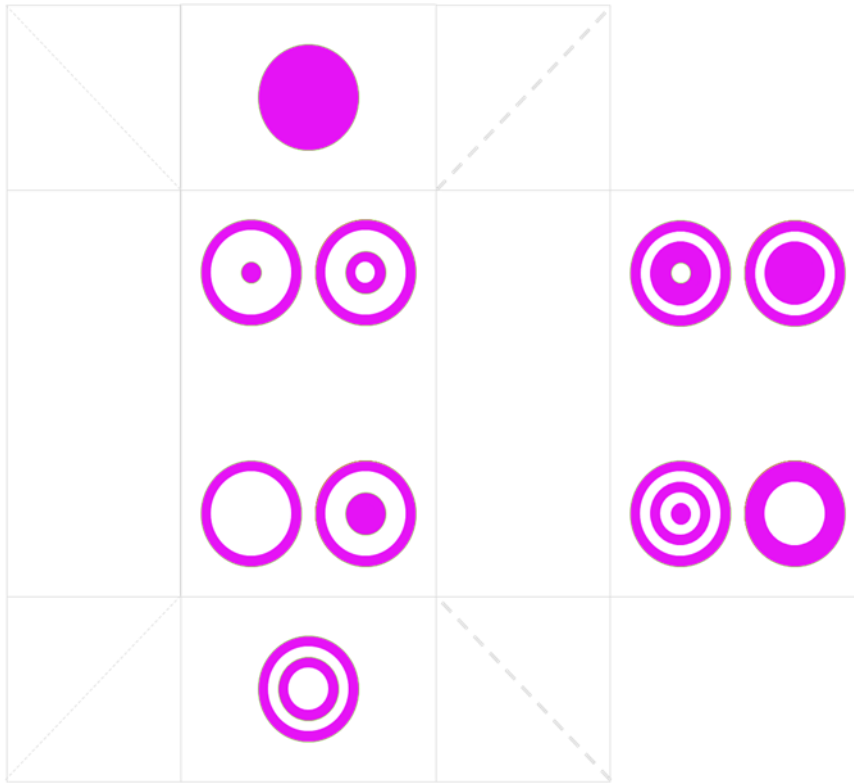
- [79] R. B. Rusu, N. Blodow, Z. C. Marton and M. Beetz, "Close-range scene segmentation and reconstruction of 3D point cloud maps for mobile manipulation in domestic environments," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems IROS 2009*, 2009.
- [80] M. J. Sadigh and H. Ahmadi, "Safe grasping with multi-link fingers based on force sensing," in *Proc. IEEE Int Robotics and Biomimetics (ROBIO) Conf*, 2009.
- [81] R. Sam and S. Nefti, "Design and development of flexible robotic gripper for handling food products," in *Proc. 10th Int. Conf. Control, Automation, Robotics and Vision ICARCV 2008*, 2008.
- [82] SharkD, "The HSV color model mapped to a cylinder. POV-Ray source is available from the POV-Ray Object Collection.," 12 11 2009. [Online]. Available: http://commons.wikimedia.org/wiki/File:HSV_color_solid_cylinder_alpha_lowgamma.png.
- [83] msdn, "RGB Color Spaces," 16 11 2010. [Online]. Available: [http://msdn.microsoft.com/en-us/library/windows/desktop/dd372185\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/dd372185(v=vs.85).aspx).

Appendix A **F3 Robot Arm Workspace** **[45]**



UNITS: MILLIMETERS

Appendix B **Example visual targets for rectangular box object**



Appendix C Hue Saturation Value (HSV) colour-space.

The HSV colour-space is a different representation of the standard Red Green Blue (RGB) colour-space. In the RGB colour-space each pixel in the image receives a different value for red, green, and blue pixels. These values add together and form a colour; this is shown in Figure C-2. An issue with the RGB colour-space is that different shades of the same colour, e.g. light and dark red, can require modification of all the RGB values. HSV colour-space is cylindrical coordinate representation of the RGB cube. In this coordinate system colours like red are represented as a set range of angles. The angle around the centre axis is represented as the hue, only one component is needed to represent it. The saturation is a measure, from zero to one, of how much white is in the colour, at zero saturation the colour is full of white while at a saturation of one the colour has no white. Value is another measure that goes from zero to one and it can be thought of as the amount of light that the colour receives.

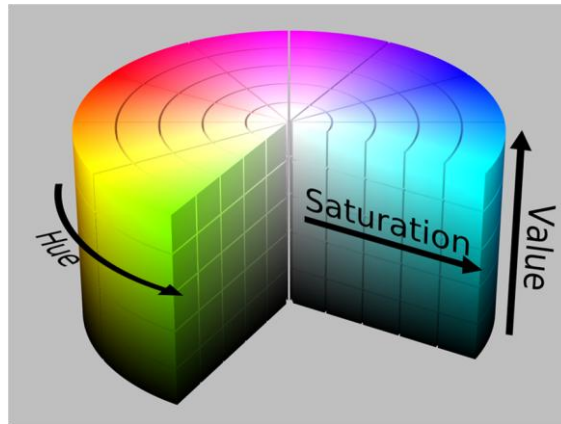


Figure C-1 The representation of the HSV colour-space [82].

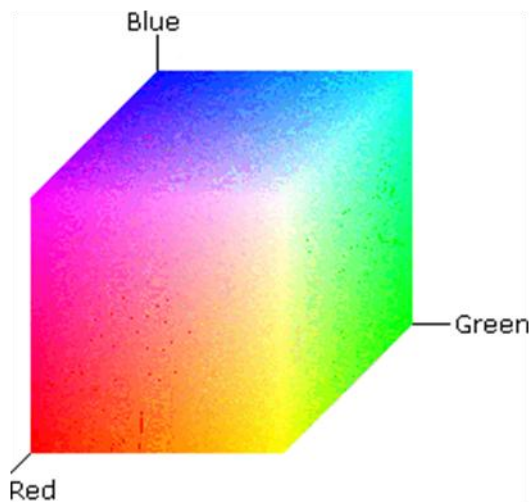


Figure C-2 The representation of the RGB colour-space, shown as a cube [83].