



Christina Sherly John Bensam

Visual Grasping Of Unknown Objects

School of Electrical Engineering

Department of Automation and Systems Technology

Thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in Technology

03.08.2016

Instructor:

Joni Pajarinen
D.Sc.(Tech.)
Aalto University
School of Electrical Engineering

Supervisors:

Professor Ville kyrki
D.Sc.(Tech.)
Aalto University
School of Electrical Engineering

Preface

The research for this Thesis was done at the Aalto University School of Science and Technology during the years 2014-2016. The Thesis is based on work performed under Robotic Grasping called ‘Visual Grasping of Unknown Objects ‘.

I thank the Almighty for His Strength and Grace to finish the Thesis. I wish to express my gratitude to Professor Dr. Ville Kyrki who supported this research and encouraged me to prepare the Thesis. Many thanks to my supervisor Dr. Joni Pajarinen for the Guidance in Thesis work, as well for helping and clearing the doubts during the whole tenure. Credits for assistance in Thesis belong also to the whole intelligent robotics team who were a constant support. I thank my parents for their constant support during the whole period.

Espoo, August 3, 2016

Christina Sherly John Bensam

Author:	Christina Sherly John Bensam		
Title of the thesis:	Visual Grasping of unknown Objects		
Date:	August 3, 2016	Number of pages:	41
Department:	Automation and System Technology		
Program:	Master's Degree Programme in Space Science and Technology		
Professorship:	Automation Technology (As-84)		
Supervisors:	Professor Ville Kyrki (Aalto)		
Instructor:	Joni Pajarinen (Aalto)		
<p>The objective of the thesis is to compare and study recent visual grasping techniques which are applied on a robotic arm for grasping of unknown objects in an indoor environment. The novelty of the thesis is that the study has led to questioning the general approach used by researchers to solve the grasping problem. The result can help future researchers in investing more on the problem areas of grasping techniques and can also lead us to question ourselves on the approach we are using to solve the grasping problem.</p>			
Keywords: Grasping, vision, unknown objects, jaco, kinova, objects, learning technique			

Contents

1	INTRODUCTION	1
1.1	Motivation	1
1.2	Objectives.....	2
1.3	Background	2
1.4	Overview of Work.....	4
2	LITERATURE REVIEW	5
2.1	Overview	5
2.2	Grasping Techniques.....	5
3	FACTORS INVOLVED IN GRASPING TECHNIQUE	10
3.1	Introduction	10
3.2	Object Classification	10
3.3	Factors considered while Grasping	10
3.4	Forces present on a object.....	11
4	ALGORITHM 1: SYMMETRY HEIGHT ACCUMULATED FEATURES.....	12
4.1	Overview of SHAF	12
4.2	Implementation of SHAF.....	14
5	ALGORITHM 2: RECTANGLE REPRESENTATION	18
5.1	Overview of Rectangle Representation.....	18
5.2	Implementation of Rectangle Representation	19
6	ALGORITHM 3: PRINCIPAL COMPONENT ANALYSIS.....	23
6.1	Overview of Principal Component Analysis.....	23
6.2	Implementation of PCA	24
7	EXPERIMENTS & RESULTS	26
7.1	Experimental Setup - Hardware.....	26
7.2	Overview of Experiments.....	28

7.3 Results.....	36
7.4 Analyzing the Results.....	39
8 CONCLUSION.....	39
REFERENCES.....	40

List of Graphs

Graph 1. Calculated probability of each of the algorithms.....	36
Graph 2. Calculated population proportion of each of the algorithms for confidence level 95%.....	37

List of Tables

Table 1. Calculated probability of each of the algorithms.....	36
Table 2. Calculated population proportion of each of the algorithms for confidence level 95%.....	37

List of Figures

Fig. 1. Robot performing grasping activity	1
Fig. 2. Claw Arcade Game.....	3
Fig. 3. Breaking down SHAF: Flow chart	12
Fig. 4. Flow chart of the implementation of SHAF	15
Fig. 5. Flow chart of the modified implementation of SHAF.....	16
Fig. 6. Top Ten Grasping Rectangle For Toy Train.....	19
Fig. 7. Top One Grasping Rectangle.For Toy Train.....	20
Fig. 8. Flow chart of the implementation of rectangle representation.....	21
Fig. 9. Flow chart of the implementation of PCA.....	24
Fig.10. Experimental Setup.....	26
Fig. 11. Objects used in Experiments.....	29
Fig. 12. Jaco grasping truck.	30
Fig. 13. Jaco grasping a four wheeler.	30
Fig. 14. Jaco grasping a toy plane.	31
Fig. 15. Jaco grasping a toy car.....	31
Fig. 16. Jaco grasping an empty cardboard box.....	32
Fig. 17. Jaco grasping a box.....	32
Fig. 18. Jaco grasping a toy plane.	33
Fig. 19. Jaco grasping a bottle.....	34
Fig. 20. Jaco grasping a block of lego.....	34
Fig. 21. Jaco grasping a ball.....	35
Fig. 22. Jaco grasping a soft toy train.	35

Symbols and Abbreviations

Aalto	Aalto University
SHAF	Symmetry Height Accumulated Features
SVM	Supervised vector Machine
PCA	Principal Component Analysis

Foreword

To compare recent visual grasping techniques which are applied on a robotic arm for grasping of unknown objects in an indoor environment. This thesis compares the algorithm written by David Fischinger on Symmetry Height Accumulated Features(SHAF) and the same has been compared with Principal Component Analysis(PCA) written by Joni Pajarinen. The work was performed at Intelligent Robotics Laboratory of Aalto University and the results were reported in the final Presentation delivered on March, 29th 2016.

Chapter 1

Introduction

1.1 Motivation

As humans, we always interact with the surrounding environment and take delight in it. Our interactions include grasping objects in the environment, holding a hot coffee cup or lifting a chair that seem effortless. However, research is being done in the field of robotics to perform such tasks. The figure below demonstrates a robot performing grasping activity.



Fig.1.Robot performing grasping activity

Grasping of objects is a learned process for humans. By time, we acquire the necessary skills to grasp any object. The learning process takes place in children, they start with grasping the object using two hands, later with one hand and then if the object is light, they are able to lift it up easily with few fingers. Grasping based on human experience can be used to imitate in robotics but at the same time we shall not limit ourselves to that.

1.2 Objectives

In the thesis, visual grasping of unknown objects, a small attempt has been made to solve the grasping problem by imitating human experience in robotics. This solving of visual grasping problem is limited to theoretical Idea and may or may not be exact in its details and may be subject to error. However, the main Objective of the thesis is to compare and study the recent visual grasping techniques which are applied on a robotic arm. These techniques are applied for grasping of unknown objects in an indoor environment and prove some of the assumptions made in the theoretical idea.

1.3 Background

The section below presents the theoretical idea discussed in the objectives. Grasping through vision is considered in the thesis. It is a fact that grasping without a sense of touch is next to impossibility for humans themselves. To prove it, it is very hard for a human to perform grasping action with a numb hand. So, the task of grasping only through vision in a robot raises a lot of questions. It is like playing a game of chance. It can be related to the claw arcade game shown in the figure below. It is usually played during fairs. No matter what point is chosen for grasping, it tends to be a game of probability. The only difference between the claw arcade game and Jaco robot [7.1] is that the fingers apply more pressure in the case of Jaco robot but not for the manipulator in the claw arcade game.



Fig.2.Claw Arcade Game

Now, the question arises how to distinguish between unknown and known objects. The known factors of the object through vision are:

1. Its material (from color and texture) which gives the estimation of the mass of the object.
2. The length, breadth and height of the object which gives estimation about the shape of the object. However, the shape of the object was not able to be determined fully due to occlusions and self-occlusions.
3. The surface in which the object is present gives an idea about the frictional force present between the surface and the object

Usually a Grasping method learns from the database about the object and performs the necessary grasping but when it comes to unknown object, there is no question of learning from a database as the object is itself unknown.

Another important factor to be considered in grasping is grip. Grip is related to the force applied on the object or the force going to be applied on the object. Considering the case of humans, for objects beyond a certain size, dragging the object by pushing and pulling makes sense as enough force cannot be applied to lift the object. Fingers apply the least force, later the palm with fingers applies little more force. Then with elbows more force is applied and slowly the whole body starts being used .To carry extreme weights the centre of mass of the object and the body trying to carry it is aligned. Example: porters in railway station. The material of the object and the surface in which the object is present is required to determine the force needed to lift

the object. For an unknown object, the object is glanced through vision, a fair guess about object weight is made, apply the assumed force and lift the object. Later apply the required force to keep the object lifted. If the assumed force is not sufficient again a guess is taken and the process continues. Grasping is totally based on grip. Whether the fingers of the manipulator has got grip over the object.

As there is no torque sensor present in the kinova-jaco robot[7.1.1], the mass and the frictional force are not going to be considered.

1.4 Overview of work

In this thesis some of the Grasping technique's like SHAF,PCA were compared by implementing them and conducting various tests on them. It was planned to use only visual sensor and not any stereo vision or tactile feedback. For the first stage, a recent method proposed by Fischinger (et al) was modified. This is a shape based method called 'symmetry height accumulated features' which helps in grasping unknown objects from a cluttered environment and the method uses a point cloud from a single depth camera. The idea behind this method is to define small regions and compare average heights using discretized point cloud data. The height difference gives an abstraction of the objects shape that enables training of a classifier (supervised learning). It determines if grasping will succeed.

Some simple grasping methods – one, using a PCA based approach was used as a baseline for comparison. A second state-of-the-art method was modified in addition to Fishinger(et al) known as rectangle representation for doing the comparisons but the experiments were not performed using this method.

The hardware used is Kinova Jaco arm, which is a 6 DOF (Degree of Freedom) Robotic arm and it can reach up to 90 cm with a lifting capacity of 1.5kg. The gripper of the robotic arm contains 3 individually controlled under actuated fingers, which safely handle variety of objects

For implementation, the source code is reused instead of developing from scratch. Then the grasping techniques are applied on the object. Then a note of the successful grasps and the failed grasps was made with also the possible reason for failure. Also, a grasp is considered to be successful if the robot manipulator is able to keep the object in air for 15 seconds.

Chapter 2

Literature Review

2.1 Overview

The idea of making a robot to take hold of something in order to perform something is known as Grasping. The action of holding a door knob to perform pushing or pulling action is also part of grasping process but grasping of objects is only considered in the thesis. Again grasping can be performed on stationary or moving objects, in which stationary objects are considered for our reference. In stationary objects also, the object can be a small ball or a big sofa, so again the restriction is placed to the objects that can be lifted by a hand.

In general, the various environments considered for grasping of objects are cluttered and uncluttered environment. The objects can be both unknown and known.

2.2 Grasping Techniques

Most of the algorithms considered use some kind of learning to obtain a stable grasp. In 1994, a group of researchers [20] worked on the grasping problem and created a family of simultaneous interacting contour trackers to pick the top object off a file from a camera on the end of the robot arm. In this algorithm geometrical information is obtained from the analysis of image motion as the robot tries to move around the vantage point. Later, the next year Marjan and Ales [21] developed a paradigm of purposive vision which works on the principle that one shall extract as much information as it is needed. Planar patches are obtained by the recover and select paradigm which contains enough information to enable generating object hypothesis for grasping of objects. In 1999, Namiki[23] and his team created a high speed grasping technique using visual and force feedback. The most important feature of the

system was its ability to process sensory feedback at high speed. Later in 2003, an algorithm [22] called automatic grasp planning using shape primitives was proposed. Some of the researchers have focused their attention on using features of the object or the environment. An interesting work [3] was done using cognitive vision system in which a representation was built based on edge and texture information. ECV system also known as early cognitive vision system has the edge and surface structures extracted by means of an extension of the biologically motivated hierarchical vision system. It extends the ECV system with a hierarchy of features in the texture domain. The proposed feature gives a sparse and abstract but meaningful representation of the scene which on one side reduces the search space for grasping and on the other side creates additional context information which is relevant for grasping. In this work a grasping benchmark was also proposed.

Further, using only geometric cues, a cloth grasp point detection based on multiple views with application to robotic towel folding was proposed by Jeremy and the team [14]. This algorithm was specific to the application of automatic laundry folding.

Another grasping technique proposed was Contact-Reactive Grasping with partial shape information [17]. A ranked set of grasps were created using heuristics based on overall shape of object and its local features. In this technique, the reactive grasping component of the system used tactile sensors for a variety of reactive behavior. Some of the behaviors were to recover from small positional errors, to grasp objects in a way that minimizes unwanted object pushing, and to locally adjust grasps that are likely to be marginal. The disadvantage of this method is that it fails in grasps that look marginal in terms of their contacts. It fails for objects that are too light to be sensed by tactile sensors, for objects too large to fit within the gripper, for flat objects and finally for objects that are not easily seen by the sensor generating the objects point cloud.

Taylor and Kleeman [19] suggested an algorithm which used colour, texture and edge information, to track objects in cluttered environment.

In visibility based accessibility analysis of the grasp points for real time manipulation [24], the objects were grasped exploiting the plane features and pre-existing object models.

Several researchers have come up with the idea of segmenting the 3D scene and then locating the unknown object.

Alper[10] using the argument that there is a strong correlation between local 3D structure and object placements proposed to capture complex 3D contexts without implementing specialized routines. The limitation of this approach is that they perform poorly in non typical scenes where contextual expectation does not agree with the scene at hand.

In Detecting and segmenting objects for mobile manipulation [11], Radu proposed to interpret the 3D scene using a set of 3D point features and probabilistic graphical methods. A scene interpretation algorithm was used in this method, in which estimation of fast point feature histograms was used as local 3D point features to segment the object into geometric primitives. Then learning and categorizing of object classes was done which used the previously estimated primitives at each point. Taking into account the contact stability conditions and force closure, a set of algorithms was proposed by Antonio[15] to grasp real objects.

Based on shape and size many algorithms were proposed to grasp the objects.

In Grasping of unknown objects from a table top[8], the authors proposed to find grasp candidate based on shape of object and validated it globally by checking collisions between gripper and surrounding objects.

In Grasping novel objects with depth segmentation [9], the authors suggest segmenting the cluttered scene with depth information and proposed a shape completion and grasp planner method which takes partial 3D information.

Also, Claire [12] created a rough shape estimation of unknown objects using non linear optimization techniques. The goal is to determine the quadric that approximates at best the shape of an unknown object using multi-view measurements. Non-linear optimization techniques are considered to achieve this goal. Since multiple views are necessary, an active vision process is considered in order to minimize the uncertainty on the estimated parameters and determine the next best view.

In a shape matching algorithm for synthesizing humanlike enveloping grasps [16], a shape matching algorithm was proposed that can accommodate the sparse shape information associated with hand pose.

Miller et al [4] also employed an automatic grasp planning system “grasp it” which modeled objects as sphere, cylinder, cone or box. Then a set of rules was used in this algorithm to generate a set of grasp starting positions and pregrasp shapes based on

the modeled objects. The grasp was tested using the simulator “GraspIt” and the best grasps were presented to the user.

On the other hand, Ben [13] proposed a cloud based grasping for objects having uncertainty in shape. This paper explores how Cloud Computing can facilitate grasping with shape uncertainty. It considers the most common robot gripper: a pair of thin parallel jaws, and a class of objects that can be modeled as extruded polygons. It models a conservative class of push-grasps that can enhance object alignment. The grasp planning algorithm takes as input an approximate object outline and Gaussian uncertainty around each vertex and center of mass. It defines a grasp quality metric based on a lower bound on the probability of achieving force closure. It presents a highly-parallelizable algorithm to compute this metric using Monte Carlo sampling. The algorithm uses Coulomb frictional grasp mechanics and a fast geometric test for conservative conditions for force closure. It runs the algorithm on a set of sample shapes and compares the grasps with those from a planner that does not model shape uncertainty.

Some studies have used a feedback system to grasp unknown objects. Using the sensor feedback information, an optimum grasp was calculated in Robotic grasping of unmodelled objects using time of flight range data and finger torque information [18]. The positive of the approach is that it doesn't require exploiting object models.

Some grasping techniques have been applied for cluttered environments.

In both visually guided grasping in 3D [20] and grasping arbitrarily shaped 3-D objects from a pile [21], the object from the top of a pile was picked for grasping using machine vision and range sensors respectively. Similarly some grasping techniques were applied only for detecting an unknown object and grasping it. These techniques ignored about collision with other objects.

In Robotic grasping of unknown objects: a knowledge based approach [7] of a system was developed in which a 3D model of the object was generated by rotating the object under a laser range scanner. Then this model was used for grasping.

Some of the other grasping techniques were proposed by Wang et al, [5] where the unknown object was grasped using 3D model reconstruction.

In this algorithm, following issues were considered to make a robotic multi fingered hand grasp of an unknown object. (a) features of the grasped object (b) grasping configuration of the hand (c) contact locations and contact normal vectors between the hand and the object (d) appropriate finger forces exerted on the grasped object to balance external forces acting on the object , which are subject to friction cone constraints.

Bone et al [6] suggested an approach which used the online silhouette and structured light 3D object modeling with online grasp planning. In this approach, a single wrist mounted video camera is moved around the stationary object to obtain images from multiple viewpoints. Object silhouettes are extracted from these images and used to form a 3D solid model of the object. To refine the model, the object's top surface is modeled by scanning with a line laser while recording images. The laser line in each image is used to form a 3D surface model that is combined with the silhouette result. Then the robot generates a force closure grasp and outputs the required gripper position and orientation for grasping the object.

Chapter 3

Factors involved in Grasping Technique

3.1 Introduction

The following chapter gives a bird's eye view of the usual factors involved in the grasping techniques.

3.2 Object Classification

The objects can be classified based on various ways. Based on deformability of the object, it can be classified. Using a balloon filled with different volumes of water levels, we can change the deformability. Later, based on flexible material also the objects can be defined. In flexible type materials, the objects having linear shape geometry are cable, wire, etc .Again in flexible type materials, the objects having sheet shape geometry are fabric/Garment, complex composite material, sheet metal, leather etc. Finally, flexible type materials having geometry of three dimensional shapes are dough, soft tissue etc .The objects can also be classified based on length, width and breadth. It can also be classified based on line of symmetry. Two other ways of classifying objects are based on depression - projection of objects and finally on complexity level of object's shape.

3.3 Factors considered while grasping

Is the object deformable and to what extent? Whether the object is heavy or light? If the object is heavy and the robot is unable to grasp, then we have to consider whether to pull or push the object to achieve the task on hand.

One of the main factor to be considered for grasping of objects is which state of matter the object is made of. Later the viscosity of the object also has to be

considered. It is also useful to consider which points of the object are near to the arm. Is there any manufacture specific grip on the object. Example: a dumbbell is shaped in a certain way for easier grasping, if instead a cylindrical shaped smooth heavyweight object is used then two hands are required to grasp it. Is there any user specified constraints while grasping objects. Example: a bowl containing sauce can never be tilted while grasping object.

3.4 Forces present on a object

The two forces which are present on the object are the contact force and the frictional force. The contact force is perpendicular to the surface of contact and the frictional force is parallel to the surface of contact. The contact force is important because more the Surface area of the robot, more is the application area of force. So Kids always try to use more surface area for lifting as they do not have the dexterity or strength to apply more force. We as humans sometimes create object which facilitates grip, like a difference between a coffee cup and drinking water cup. The frictional force is important because we apply different amount of force to grasp the same object depending on which surface it is in contact with. That is it depends on the friction between the two objects.

According to Newton's first law, if the sum of forces acting on a particle is zero then and only then the particle remains unaccelerated. i.e. the particle is at rest or at constant velocity . So the force to apply on the object to grasp it also depends on this. Therefore, an accelerated object requires more amount of force to be applied to grasp than the one at rest .Note that Newton's law is valid only for inertial reference frame and earth is not strictly an inertial reference frame.

Considering the force present on the object, if we determine grasping of object the focus will be on mainly 2 things – Firstly, to get more surface area of the manipulator's palm over the object(for more force application).Secondly, to determine the location where the least amount of force has to be applied on the object.

Chapter 4

Symmetry Height Accumulated Features

4.1 Overview of SHAF

Overview of SHAF is given in the following section, to get a general idea about the algorithm which is used in experimenting. The following flowchart provides an overview of the breakdown of SHAF.

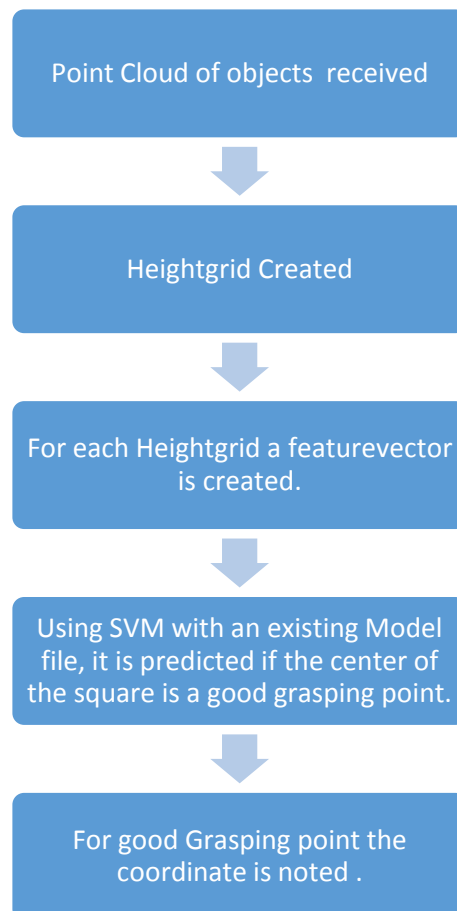


Fig.3.Breaking down SHAF:Flowchart

SHAF also known as symmetry height accumulated features is a shape based method which helps in grasping unknown objects from a cluttered environment. The method

uses a point cloud from a single depth camera. The idea behind this method is to define small regions and compare average heights using discretized point cloud data. The height difference gives an abstraction of the objects shape that enables training of a classifier (supervised learning). It determine if grasping will succeed.

In this method, the point cloud is discretized that is a height grid is created where each cell saves the highest z value with corresponding x and y value. One height accumulated feature is defined as the two, three or four region on the height grid together with a weighting factor for each region.

For a feature which has two overlapping regions, the feature value is calculated in a way that the feature value is zero if both regions have same average height, bigger than zero if the region inside is higher and smaller than zero if the smaller region has in average a smaller height.

Around 35000 features were created automatically using the above constraint by David Fischinger and the team at Vienna university of Technology (Austria). Also 500 features was created manually. It was noted that the same feature value can be achieved if the centre region height exceeded both regions height by x or if the centre region and one side region have equal heights and exceeds the second side region by $2x$. To remove this ambiguity, symmetry features were introduced. It calculated the minimum distance of accumulated heights between center regions and side regions, if the centre was in average the highest and (-1) otherwise.

Now to perform the actual grasp, initially a learning process is provided for grasping. Then secondly, a weighting method is used to achieve more robust grasp. Then thirdly, the SHAF method is used to explore the whole grasp space. Finally grasping point and path planning is done to achieve the required grasp.

For learning purpose, 700 point clouds of a variety of objects were gathered. In which 450 times the most promising grasp of the scene was labeled and 250 times scenes were labeled with positions where no grasp success was possible. Then using methods like scaling, mirroring and inverting, around 8300 positive and 12,800 negative examples were generated. For all these examples, SHAF (symmetry height accumulated features) were calculated and a SVM classifier with radial basis function kernel was trained.

As discussed above, a weighting method is used to obtain a more robust grasp. In a real life scene, usually the trained grasp classifier does not return a single grasp point

but instead returns a bunch of grasp points in a particular region. Generally, a point centered at such a grasp region is a good choice for a stable grasp. Each point classified as good grasping position is evaluated by

$$v(r, c) = \sum_{x,y \in N}^n I_{grasp}(x, y) \cdot w_{r,c}(x, y)$$

Where r, c indicate the actual row and column of the grasp location in the grid. I is the indicator function for a grasp point.

$$I_{grasp}(x, y) = \begin{cases} 1 & \text{if grasp at location } x, y \\ 0 & \text{if no grasp at location } x, y \end{cases}$$

Thirdly, to explore the whole grasp space, we rotate the point cloud and detect the best grasp on the newly calculated height grid and transform the coordinates back to the original coordinate frame.

Finally, using moveit, package path planning and the task of grasping is performed.

4.2 Implementation of SHAF

4.2.1 Roadmap to implementation of SHAF

Initially, worked on the old software of jaco robot which had no proper documentation. It was also time-consuming to run the software.

A software later had been developed by a group of aalto students [25] to access jaco arm using moveit and ROS hydro. Modified version of Calibration code written from ROS Fuerte to ROS Hydro was developed by an aalto phd student [Rajkumar Muthuswamy].

The shaf code [26] first provided was written in Ros fuerte and was written for PR2 robot. The packages were rosbuilt and used outdated versions of pcl, openrave. The new code of shaf was requested from the author (David Fischinger) and the author provided with SHAF code using Ros hydro for kuka arm and michelangelo hand.

4.2.2 Implementation of SHAF

For the implementation, it has to be noted that Explanation and source code of symmetry height accumulated features (SHAF) was provided by David Fischinger from Vienna university of technology, Austria. The code used kuka arm with ROS

Hydro and Openrave interface. As we were using kinova jaco arm in which moveit interface has been implemented, the code had to be modified.

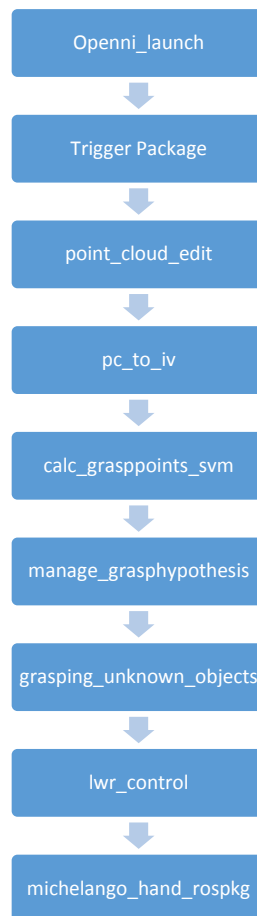


Fig.4. Flow chart of implementation of SHAF

Primary implementation of SHAF Packages

The above flowchart provides the implementation of SHAF.

Trigger:It publishes a point cloud for use in grasp detection.

Point_cloud_edit: It filters, transforms and cuts the point cloud to make it usable (in robot coordinate system).

Pc_to_iv: It transforms the ROS pointcloud(PointCloud2) into an inventor file(for use in openrave).

Calc_grasppoints_svm: It calculates the grasp using svm for learning.

Manage_grasphypothesis: It collects possible grasp points and selects the best one for execution.

Grasping_unknown_objects: It provides the simulation in openrave and performs the fine calculation of grasps.

Lwr_control: It is an interface to kuka arm(FRI library from Stanford is used to communicate with the kuka arm).

Michelangelo_hand_ropkg: The given Package controls the manipulator.

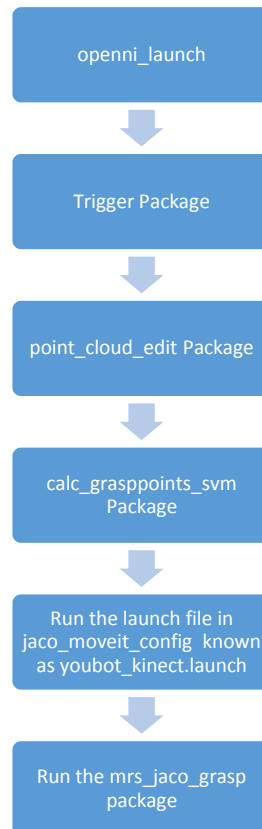


Fig.5.Modified flow chart of implementation of SHAF

Modified implementation of SHAF Packages

The above flowchart provides the modified implementation of SHAF.

Trigger: It publishes a point cloud for use in grasp detection.

Point_cloud_edit: It filters, transforms and cuts the point cloud to make it usable (in robot coordinate system).

Calc_grasppoints_svm Package:The given Package is calculating grasp points from a point cloud. In the first step, the point cloud is read from a ROS topic and a heightsgrid created. For each 14x14 square of the heightsgrid a featurevector is created. Using SVM with an existing model file, it is predicted that the centre of the

square is a good grasping point. For good grasping points the coordinates are published. The approach vector for all the algorithms is by default taken as top down.

youbot_kinect launch file :It publishes a static transform publisher from camera_depth_optical_frame to root of the jaco robot.

mrs_jaco_grasp: In the given package manipulator moves 25cm above object, turns the wrist and moves to the object and closes the gripper and moves 25cm away from the object and goes to a random position in an assigned area and opens the gripper.

Chapter 5

Rectangle Representation

5.1 Overview of Rectangle Representation

Overview of Rectangle Representation is given in the following section, to get a general idea about the algorithm. Given an image and a depth map, the method determines the oriented rectangle where grasp is possible. It also provides the grasping opening width.

This algorithm takes an image with 4 values at each pixel namely RGBD and produces an optimal grasping rectangle with the 3D location, the 3D orientation and the distance between the two fingers. The final result that is the grasping rectangle is obtained through a two step process. In the preliminary step, few features are used to compute a score function which assigns real number to a rectangle based on its features. Then to obtain a rectangle with the highest score in the image, we use some more sophisticated features to refine the search.

We use supervised learning approach, to learn the score function. As there can be more than one grasp in an object some grasps tend to be desirable. In the supervised learning of a screwdriver, the handle is preferred due to its size than the shaft. During training, the good grasps are ranked rather than classified as good ones.

To efficiently learn the representation, first, we describe a certain class of features that makes the inference in the learning algorithm fast. Second, we describe certain advanced features that are significantly more accurate but take more time to compute. Each step is learned using svm ranking algorithm. With the top results from the first step, we run a second classifier that is more accurate in order to find the top choice.

In the present setting, the robot takes an image along with a point cloud. The representation used for learning the grasping point has a sufficient seven dimensional statistics (3D location, 3D orientation and distance between fingers).

5.2 Implementation of Rectangle Representation

For the implementation, the authors provided the code [26] in which the PR2 robot was used. The code was written in ROS Fuerte and also used openrave interface. The code was modified in ROS Hydro for kinova jaco arm and also using moveit interface. The implementation worked only for a certain distance when the kinect was kept in the top down view. After certain distance it stopped detecting the grasping rectangles in the object as the view of the kinect included much larger area. To perform the grasping the kinect has to be at a certain distance above the robot arm so that it won't hit the arm. Hence the experiments were not able to be performed. But keeping the kinect at a lower distance, the grasping rectangles of the object were obtained. As shown in Fig.6 and Fig.7 the grasping rectangle gives an idea of how the grasps would look like.



Fig.6.Top Ten Grasping Rectangles for toy train

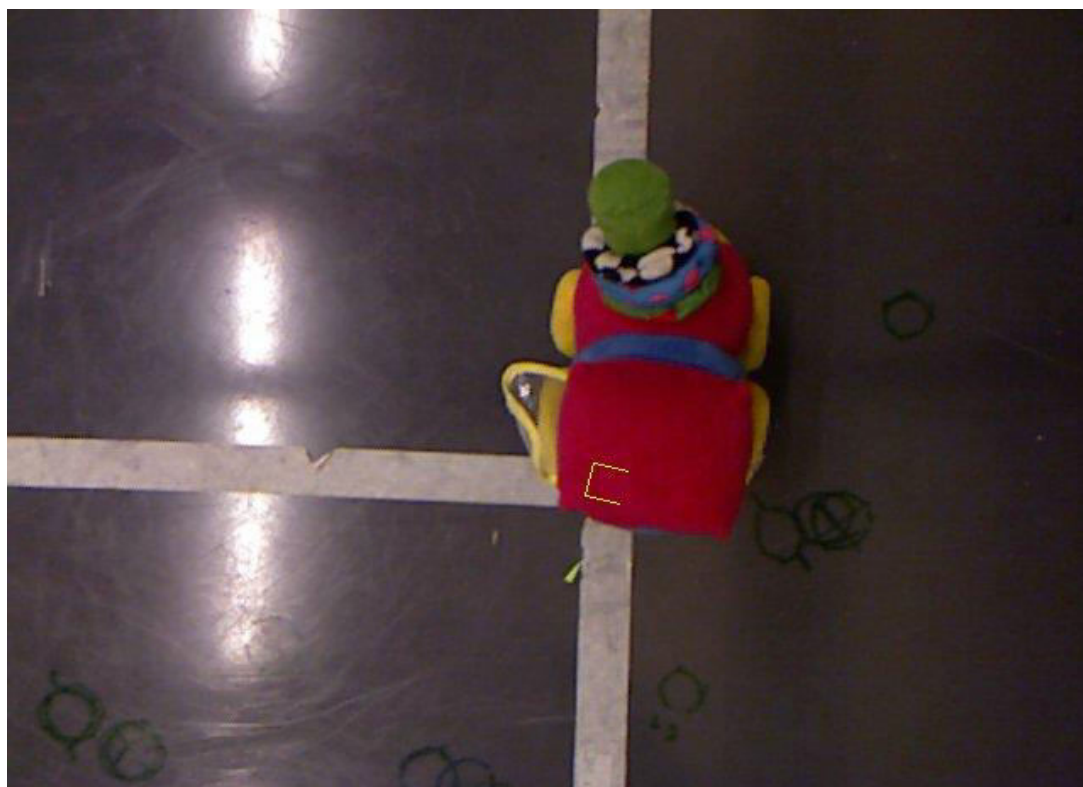


Fig.7.Top Grasping Rectangle for toy train

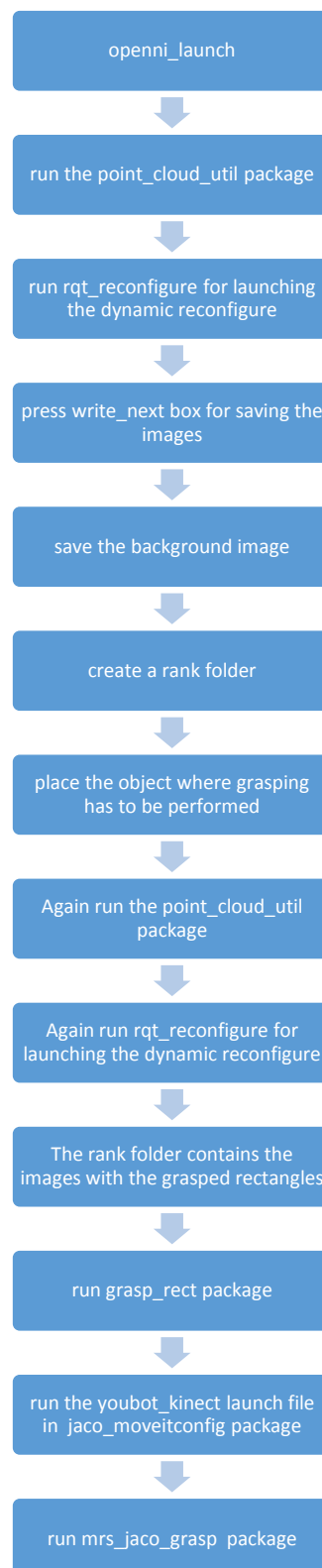


Fig.8.Flow chart of the implementation of Rectangle representation

Implementation of Rectangle Representation Packages

The above flowchart provides the implementation of Rectangle representation.

Note: The general modifications of using kinova jaco instead of PR2 robot has not been elaborated as the algorithm has not been used in the experiments.

Rqt_reconfigure Package: The Package is run for launching the dynamic reconfigure.

Point_cloud_utils Package:The package opens a dialog box with a write_next box. When the user presses the write_next box, the image is saved in /tmp folder of the system. To make the algorithm work, the image of the scene without the object is taken. Then the required object is placed on the scene and a /tmp/rank folder is created manually. The Package runs again and saves the image of the object with the calculated grasped rectangles.

Grasp_rect Package: It transforms a point from \"camera_depth_optical_frame\" to \"root\".That is it receives the kinect coordinates of the top grasped rectangle and converts it to jaco root coordinates.

youbot_kinect launch file :The file publishes a static transform publisher from camera_depth_optical_frame to root of the jaco robot.

mrs_jaco_grasp: In the given package the manipulator moves 25cm above the object, turns the wrist and moves to the object and closes the gripper and moves 25cm away from the object and goes to a random position in an assigned area and opens the gripper.

Chapter 6

Principal Component Analysis

6.1 Overview of Principal Component Analysis

The grasping approach is based on using PCA to find a vector from point cloud data for grasping. PCA also known as principal component analysis is a simple approach which selects the finger distance and rotation of the robot hand by computing a vector at a narrow part of the unknown object with principal component analysis.

In the PCA based method following steps are performed

- 1) projects the point cloud PC1 of the target object hypothesis onto a plane, which is parallel to the wrist of the down pointing robot hand.
- 2) makes the point density of the projected point cloud uniform to get the point cloud PC2
- 3) projects the centroid of PC2 towards PC1 along the wrist-plane normal, by the distance between the centroids of PC1 and PC2 to get the grasp centroid.

Then the approach computes the PCA decomposition of PC2.

In PCA, the first eigenvector aligns to the largest variance in the point cloud data and the second eigenvector is orthogonal to the first. Example, for a long object, the first eigenvector can be aligned along the length of the object and the second eigenvector along the width of the object. Then it uses the second eigenvector to get a narrow grasp.

In more detail, the approach projects two points in opposite directions from the grasp centroid, along the computed second eigenvector. Finally the approach selects the two points from PC1 which are closest to the projected points as the two grasp contact points. In addition, the approach checks whether some part of another object hypothesis blocks the direct path up from the two grasp contact points, and if so, sets the probability of a successful grasp to zero.

6.2 Implementation of Principal Component Analysis

The implementation for PCA based approach was already done for the old driver of jaco arm and ROS Fuerte was used. It was modified to the new driver of the jaco arm which used moveit package and also modified to ROS Hydro.

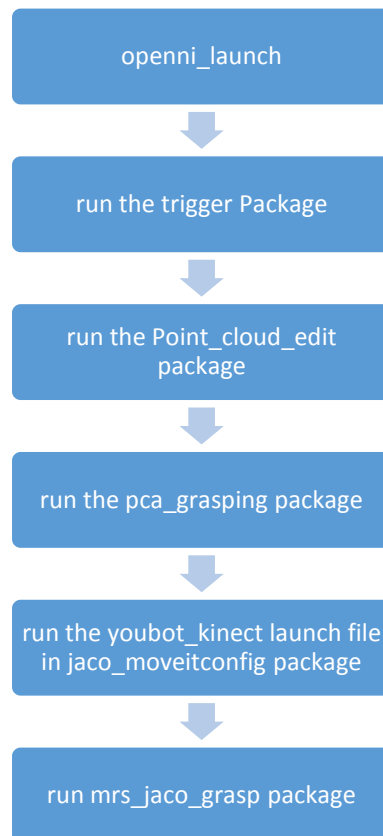


Fig.9.Flow chart of the implementation of PCA

The above flowchart provides the implementation of PCA.

Trigger: It publishes a point cloud for use in grasp detection.

Point_cloud_edit: It filters, transforms and cuts the point cloud to make it usable (in robot coordinate system).

Pca_grasping Package :In pca_grasping Package, It creates a planar coefficient model and a projection onto the wrist plane. It also Creates KD-tree for minimum distance computation. In this package, it

Project's points onto a plane that goes through robot hand wrist. (Note that the plane is in world coordinates).Then it compute the first two dominating vectors using PCA. Later it uses the narrower vector for choosing the grasp points

youbot_kinect launch file :The file publishes a static transform publisher from camera_depth_optical_frame to root of the jaco robot.

mrs_jaco_grasp: In the given package the manipulator moves 25cm above the object , turns the wrist and moves to the object and closes the gripper and moves 25cm away from the object and goes to a random position in an assigned area and opens the gripper.

Chapter 7

Experiments & Results

7.1 Experimental setup- Hardware

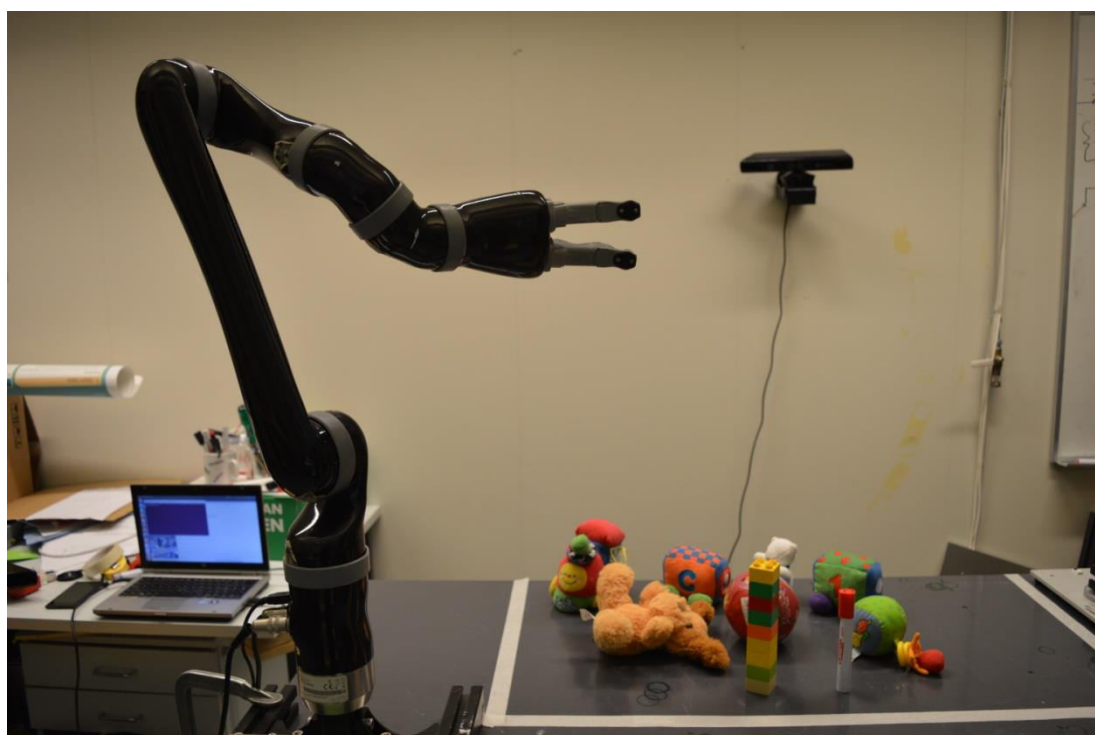


Fig.10.Experimental Setup

The Hardware used during the experiments was the kinova jaco robotic arm[7.1] for manipulating objects and the kinect [7.2] for visually sensing the objects.

7.1.1 Kinova Jaco Arm

Kinova started primarily for creating the best assistive robot for upper body disabled people. They created a Jaco arm which is a device that enables the user to interact efficiently with the environment. It is a light weight robotic manipulator with 6 degrees of freedom. The Jaco arm weighs 5.6 kg and can reach approximately 90cm in all directions and lift objects of up to 1.5kg. Its main structure is entirely made of

carbon fibre with an aluminum extruded support that can be affixed to almost any surface. The maximum linear arm speed of the arm is 15cm/sec. The gripper of the robotic arm has three under-actuated fingers that can be individually controlled. Jaco can be controlled with a 3 axis, 7 button joystick or with a computer. The control of the arm is both Cartesian and angular. In the case of Cartesian user only controls the movements of and around the hand. In the case of angular it consists of moving the robotic arm joint by joint by specifying angles to each of them. It allows three different modes: translation, rotation and grip.

Jaco has two default factory setting positions called the home and retract positions. Home refers to the position of the arm when it is ready to use and retract refers to the position when it is not in use.

Three fingered kinova jaco arm can apply a maximum force of 40N in an object and it can hold maximum a cylindrical object of diameter 10cm.

7.1.2 Kinect

Kinect is a peripheral containing suite of sensors created primarily for Microsoft xbox 360 to provide a controller free gaming and entertainment experience. It uses a infrared projector, camera and a special microchip to track the movement of objects in three dimensions. The depth sensor of the kinect consists of an infrared laser projector combined with a monochrome CMOS sensor, which captures video data in 3D under any ambient light conditions. The sensing range of the depth sensor is adjustable, and Kinect software is capable of automatically calibrating the sensor based on physical environment.

The information gathered about kinect is only based on reverse engineering as kinect is a closed product. Reverse engineering has determined that the Kinect's various sensors output video at a frame rate of ~9 Hz to 30 Hz depending on resolution. The default RGB video stream uses 8-bit VGA resolution (640 × 480 pixels) with a Bayer color filter, but the hardware is capable of resolutions up to 1280x1024 (at a lower frame rate) and other color formats such as UYVY. The monochrome depth sensing video stream is in VGA resolution (640 × 480 pixels) with 11-bit depth, which provides 2,048 levels of sensitivity. The Kinect can also stream the view from its IR camera directly (i.e.: before it has been converted into a depth map) as 640x480 video, or 1280x1024 at a lower frame rate. The Kinect sensor has a practical ranging

limit of 1.2–3.5 m (3.9–11.5 ft) distance when used with the Xbox software. The area required to play Kinect is roughly 6 m^2 , although the sensor can maintain tracking through an extended range of approximately 0.7–6 m (2.3–19.7 ft). The sensor has an angular field of view of 57° horizontally and 43° vertically, while the motorized pivot is capable of tilting the sensor up to 27° either up or down. The horizontal field of the Kinect sensor at the minimum viewing distance of $\sim 0.8 \text{ m}$ (2.6 ft) is therefore $\sim 87 \text{ cm}$ (34 in), and the vertical field is $\sim 63 \text{ cm}$ (25 in), resulting in a resolution of just over 1.3 mm (0.051 in) per pixel. The microphone array features four microphone capsules and operates with each channel processing 16-bit audio at a sampling rate of 16 kHz.

7.2 Overview of the Experiments

The calibration is performed between the jaco robotic arm and the kinect, so that vision and arm can coordinate with each other. The placement of arm with respect to vision also plays an important role. The arm's reachability and kinect's vision affects the grasping process. So, the result of these algorithms may vary for different placement of the kinect and jaco arm. The Calibration is tested to maintain the accuracy for grasping process. If there is an offset in the calibration, then the offset is added to the algorithms to maintain the accuracy.

Each of the algorithm is implemented as briefed in the respective chapter[4][5][6]. Later, Experiments are performed using each algorithm on each object separately. About 30 objects were marked for the experiments which included soft toys and other general toys of various shapes and sizes which can be held by a hand. Most of the Objects used during the experiments are displayed in the figure below.



Fig.11. Objects used in Experiments

Each of the object was marked as success or failure for each of the attempts of the three algorithms. Two attempts were performed on each of the algorithms. For each attempt of each algorithm, the percentage of success rate was calculated. Later taking the percentage of success rate as the observed proportion and keeping the confidence level as 95%, the confidence interval was calculated for each of the algorithm. Later the results were compared and a conclusion was reached.

7.2.1 Constraints while Experimenting with SHAF algorithm

Some of the constraints observed while performing experiments with the SHAF algorithm are:

1. While closing the gripper arm loses the object
2. Doesn't pick up the object while grasping
3. pushes the object while moving to the required position
4. Throws the object during acceleration to an out of workspace area
5. The fingers of the arm is not able to grasp the Object
6. pushes the object while closing the gripper

The below figures show the constraints observed in action.

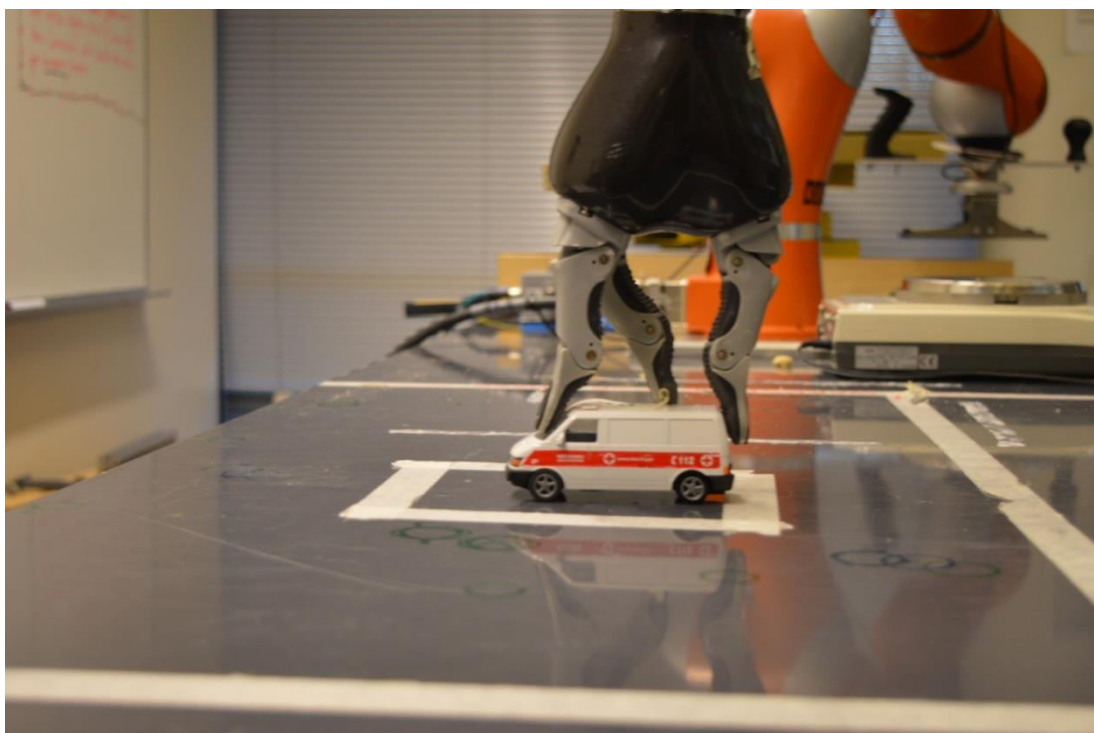


Fig.12.Jaco grasping truck

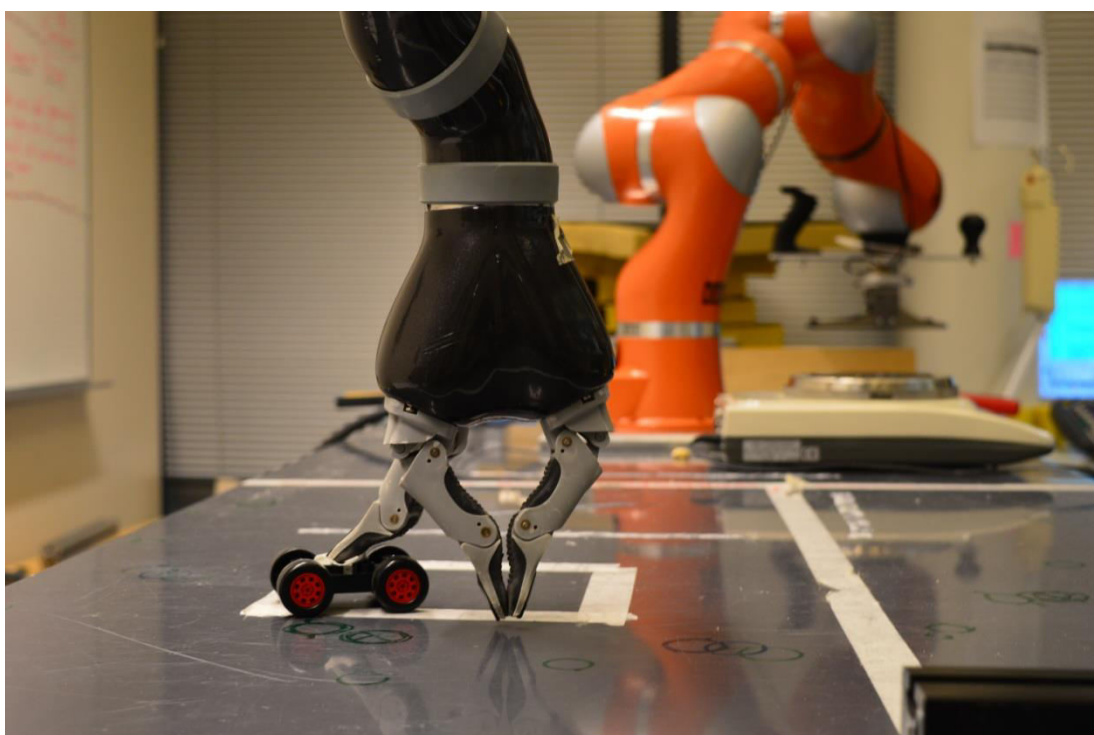


Fig.13.Jaco grasping a four wheeler

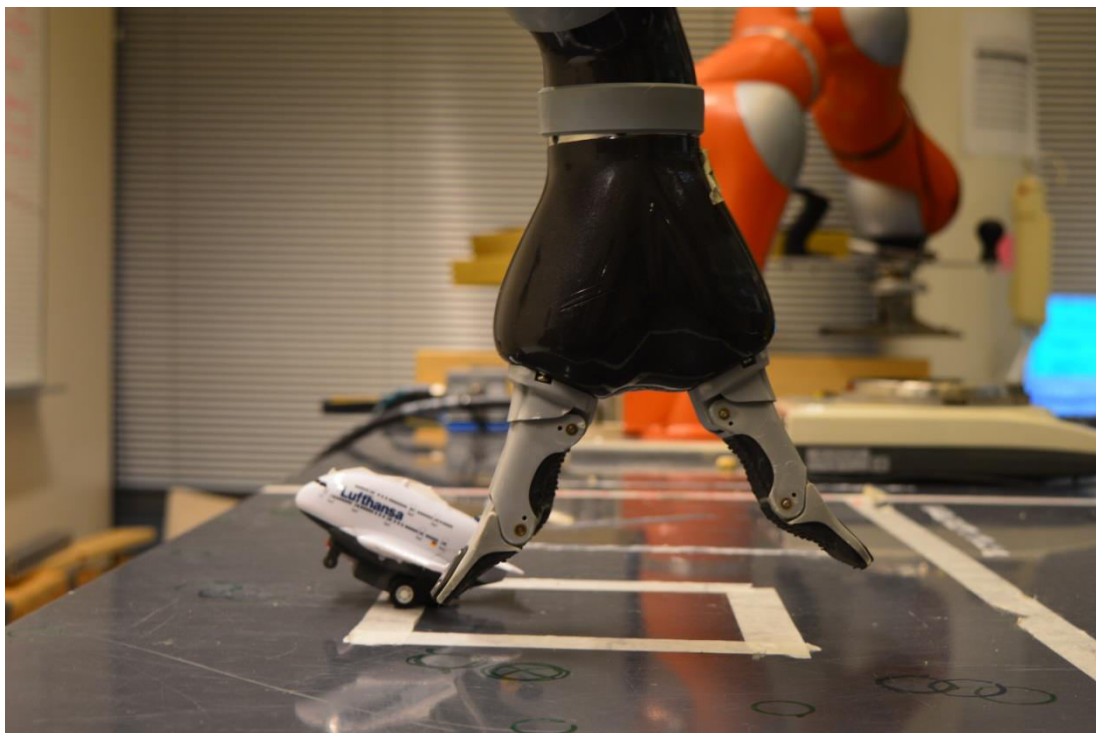


Fig.14.Jaco grasping a toy plane



Fig.15.Jaco grasping a toy car

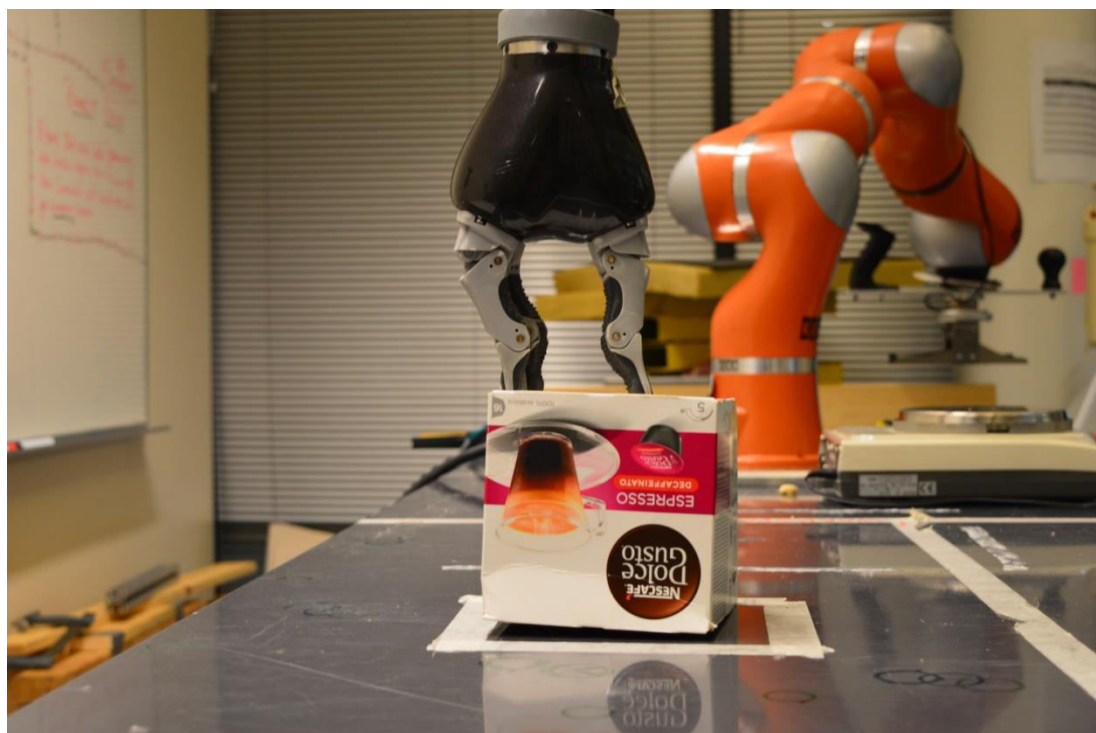


Fig.16.Jaco grasping an empty cardboard box

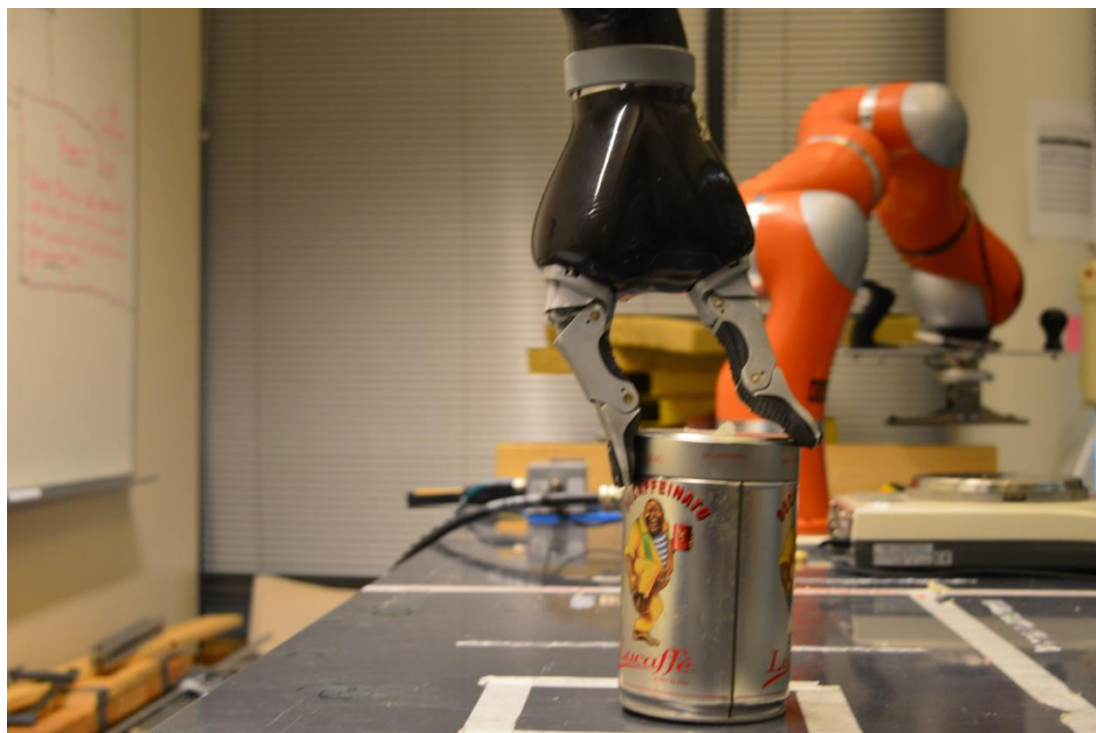


Fig.17.Jaco grasping a box

7.2.2 Constraints while Experimenting with Principal Component analysis algorithm

1. The jaco arm doesn't reach the object's height.
2. Object is crashed during the Grasping process.
3. Pushes the object while reaching to the grasp location.
4. Reaches the correct grasping position but moves the object while closing the fingers.

The below figures show the constraints observed in action.

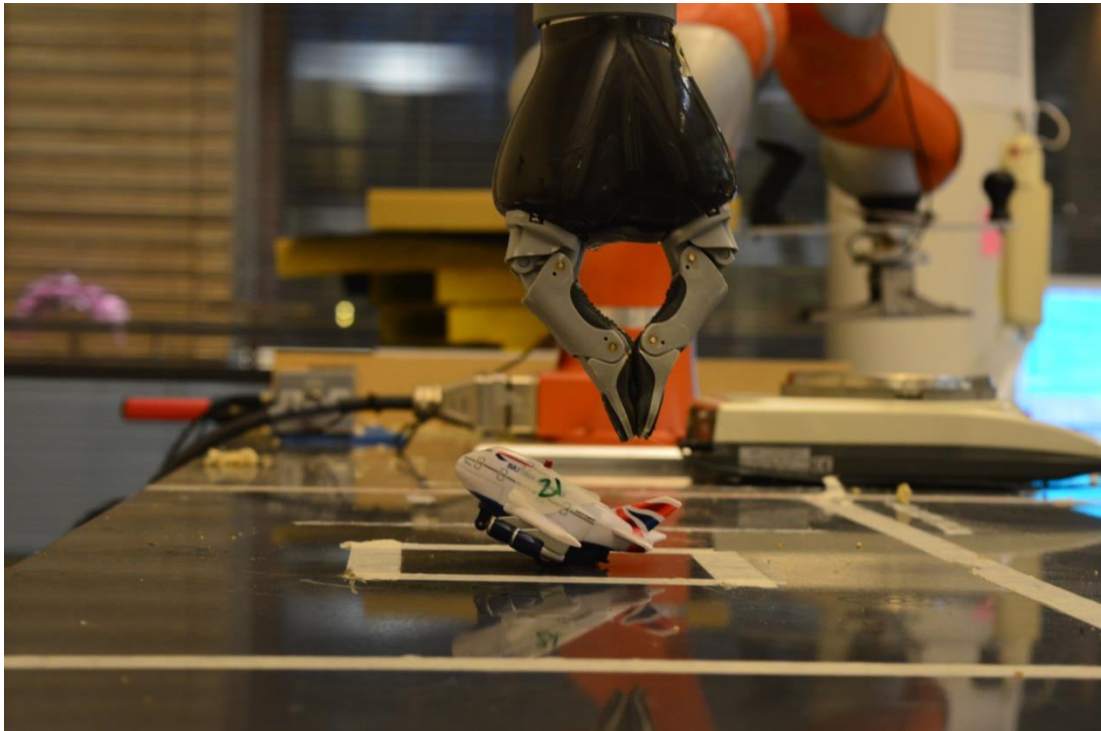


Fig.18.Jaco grasping a toy plane

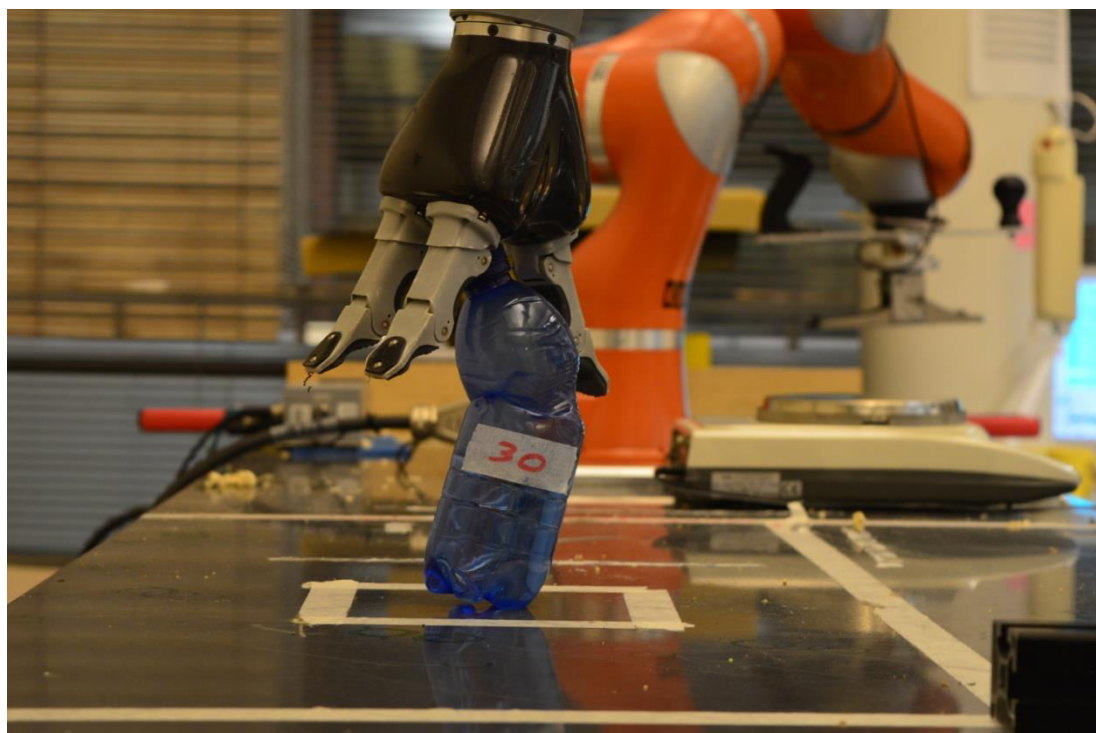


Fig.19.Jaco grasping a bottle



Fig.20.Jaco grasping a block of lego

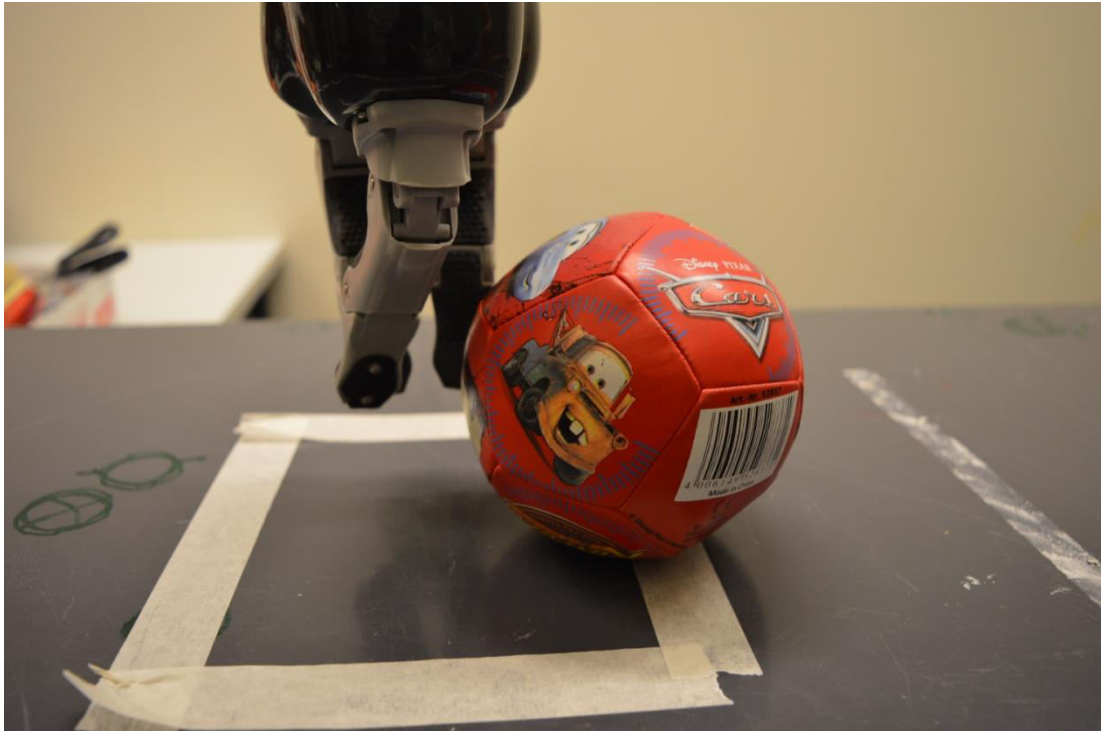


Fig.21.Jaco grasping a ball

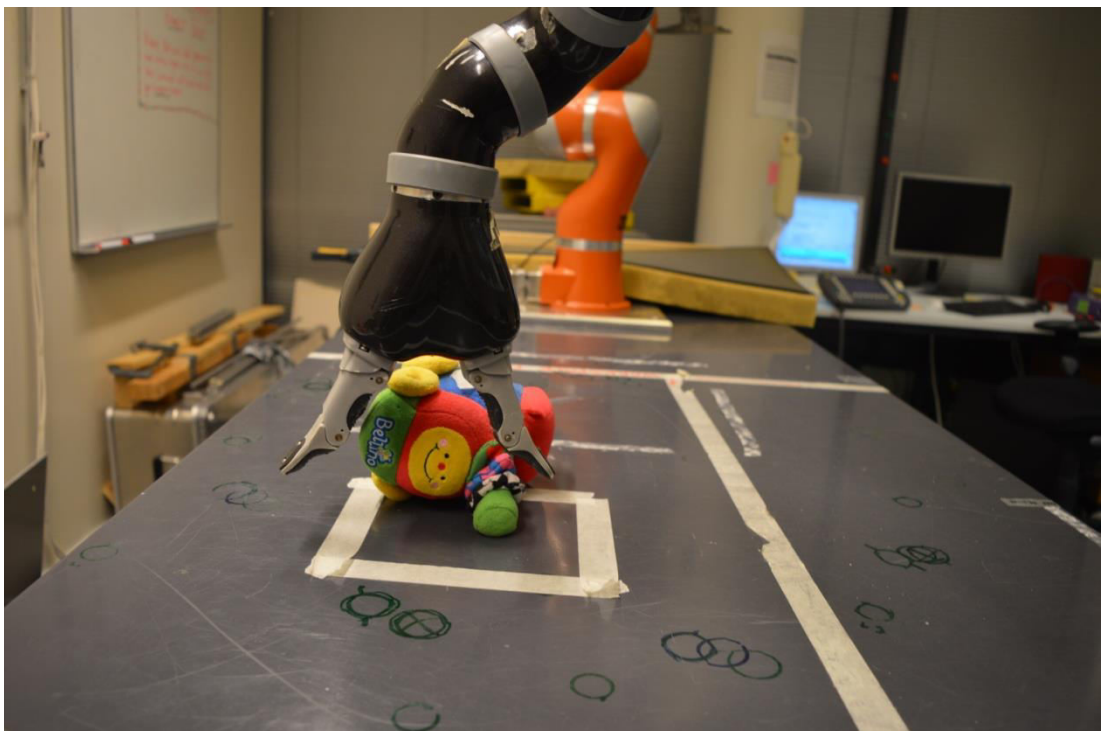
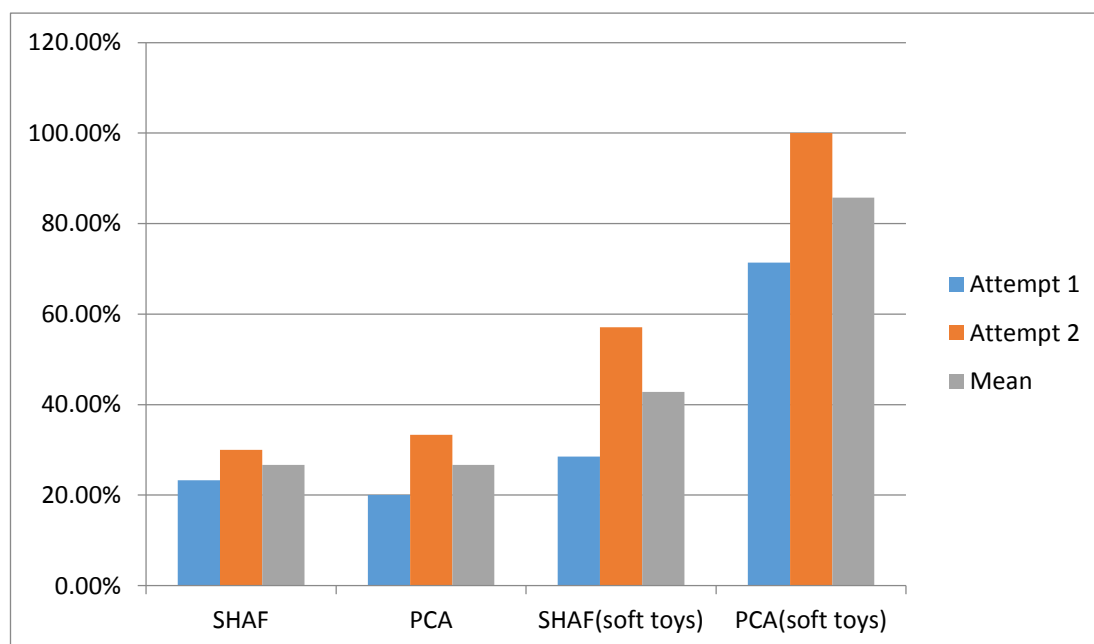


Fig.22.Jaco grasping a soft toy train

The above constraints observed in each of the algorithms also prove the importance of sense of touch or a need for a tactile sensor.

7.3 Results

Observing that there was higher probability of grasping for soft toys, the result of the algorithms was calculated separately for soft toys. The final results are displayed in the figure shown below:



Graph 1. Calculated probability of each of the algorithms

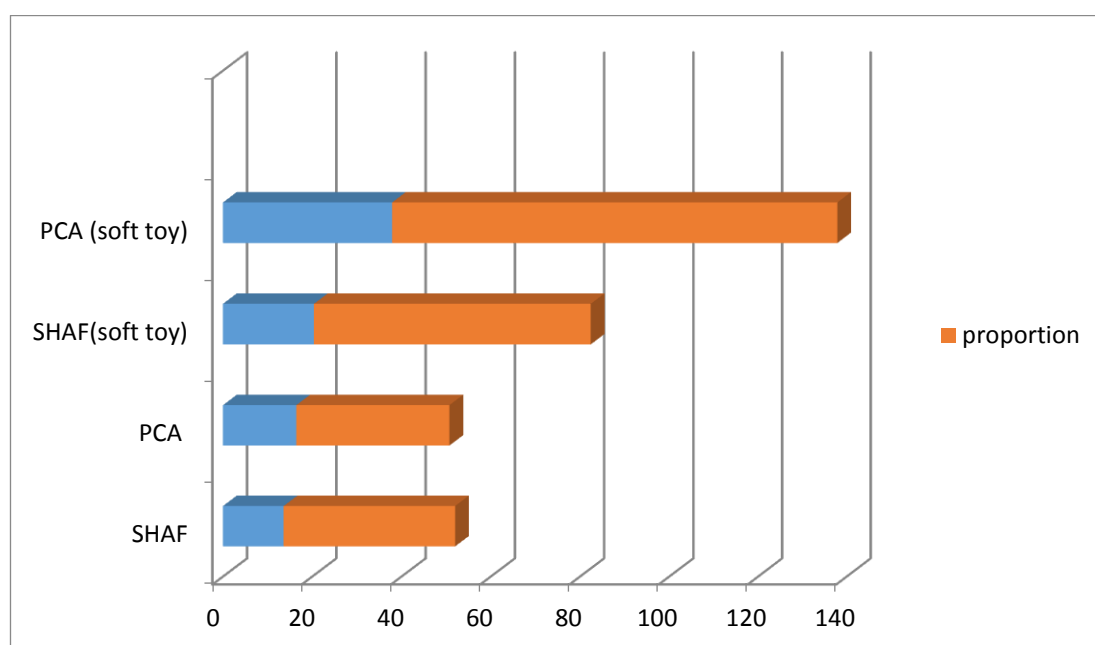
	ATTEMPT 1	ATTEMPT 2	MEAN
SHAF ALGORITHM	23.30%	30%	26.65%
PCA ALGORITHM	20%	33.30%	26.65%
SHAF (SOFT TOYS)	28.50%	57.10%	42.80%
PCA (SOFT TOYS)	71.40%	100%	85.70%

Table 1. Calculated probability of each of the algorithms

The Experiments were performed twice on each of the algorithms and the probability was calculated. Later the mean probabilities of these two attempts were calculated on each of the algorithm. Interestingly, the mean probabilities of SHAF and PCA were same. When the probabilities of soft toys for these algorithms were taken separately, then it was noted that their probabilities were higher comparatively. The mean

probability of PCA algorithm for soft toy was comparatively much higher than the mean probability of SHAF algorithm for soft toys.

Later, taking confidence level as 95%, the population proportion was calculated for each of the algorithms. For qualitative variables, the population proportion is a parameter of interest. Also it was calculated for soft toys separately for both the attempts. Then mean of this population proportion was calculated and displayed in the below figure:



Graph 2. Calculated population proportion of each of the algorithms for confidence level 95%

POPULATION PROPORTION	
SHAF ALGORITHM	13.60 – 38.43
PCA ALGORITHM	16.44 – 34.31
SHAF(soft toys)	20.40 - 62.03
PCA(soft toys)	37.90 – 100.0

Table 2. Calculated population proportion of each of the algorithms for confidence level 95%

The results show that there is not much difference between the population proportion intervals of PCA and SHAF algorithm. It seems that the result differs when only soft toys are taken and the population proportion interval of PCA algorithm is much higher compared to SHAF algorithm.

7.4 Analyzing the results

The results show that the probabilities of both the algorithms doesn't differ much confirming on the notion discussed in the beginning that without tactile sensors the grasping problem is like a claw arcade game played at fairs. The result however seemed to differ for soft toys with PCA having an advantage over SHAF. One of the reasons may be the fact that most soft toys have even height rather than uneven heights and PCA algorithm seemed to have advantage over this as SHAF algorithm is calculated based on heights.

Chapter 8

Conclusion

This work has studied Visual Grasping over a range of Unknown Objects. The experimental conclusion is that the probability of both the algorithm SHAF and PCA doesn't differ much. It confirms the notion that without tactile sensor the grasping problem is like a claw arcade game played at fairs. The results have shown that force play a huge role in the grasping problem. It will be a good idea to try the same algorithms considering tactile sensors and force control robotic arms like the new kinova's robotic arm jaco-2. Initially the force applied for lifting is more than the force applied on the fingers for grasping. After the lifting process is over the force applied shifts to the fingers and the fingers apply a constant force for grasping objects. As discussed previously, through vision we can consider the material of the object (by knowing the color and texture of the object) for a fair guess of the mass of the object. Through vision we can also consider the placement of the object for a fair guess of the frictional force present between the object and its contact surface. Considering the above factors will give a fair idea about the force to be applied for lifting the object. Later using tactile sensors, the real mass can be calculated and a constant force be applied. The way fingers are placed while grasping is also extremely important. More surface area of hand over the object more the force applied on the object. It will be good to consider the above factor for planning grasping.

Relevance of the proposed design can be tried on future grasping related techniques.

References

- [1] Ian Lenz, Honglak Lee and Ashutosh Saxena, "Deep Learning for Detecting Robotic Grasps", 2013
- [2] Ashutosh Saxena, Justin Driemeyer, Justin Keams, Chioma Osundu, Andrew Y. Ng, "Learning to grasp novel objects using vision"
- [3] Mila Popovich, Gert Kootstra, Jimmy Alison Jorgensen, Danica Kragic, Norbert Kruger, "Grasping unknown objects using an early cognitive vision system for general scene understanding", IEEE, 2011
- [4] Miller A. T., Knoop, "Automatic grasp planning using shape primitives", 2003
- [5] Wang, Jiang, "Grasping unknown objects based on 3D model reconstruction", 2005
- [6] Bone, G. M., Lambert, A., Edwards, M., "Automated modelling and robotic grasping of unknown three dimensional objects", 2008
- [7] Stansfields, S. A., "Robotic grasping of unknown objects: a knowledge based approach", 1991
- [8] Mario Richtsfeld and Markus Vincze, "Grasping of unknown objects from a table top", 2008
- [9] Deepak Rao, Quoc V. Le, Thanathorn Phoka, Morgan Quigley, Attawith Sudsang and Andrew Y. Ng, "Grasping novel objects with depth segmentation", IEEE, 2010
- [10] Alper Aydemir, Patric Jensfelt, "Exploiting and modelling local 3D structure for predicting object locations", IEEE, 2012
- [11] Radu Bogdan Rusu, Andreas Holzbach, Michael Beetz, "Detecting and segmenting objects for mobile manipulation"
- [12] Claire Dune, Eric Marchand, Christophe Collinet, Christophe Leroux, "Active rough shape estimation of unknown objects", IEEE, 2008
- [13] Ben Keheo, Dmitry Berenson, Ken Goldberg, "Toward cloud-based grasping with uncertainty in shape: estimating lower bounds on achieving force closure with zero-slip push grasps"
- [14] Jeremy Maitin-Shepard, Marco Cusumano-Towner, Jinna Lei and Pieter Abbeel, "Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding"
- [15] Antonio Morales, Pedro J. Sanz, Angel P. del Pobil, Andrew H. Fagg, "Vision-based three-finger grasp synthesis constrained by hand geometry"
- [16] Ying Li, Nancy S. Pollard, "A shape matching algorithm for synthesizing humanlike enveloping grasps", 2005
- [17] Kaijen Hsiao, Sachin Chitta, Matei Ciocarlie and E. Gil Jones, "Contact-reactive grasping of objects with partial shape information", IEEE, 2010
- [18] Alexis Maldonado, Ulrich Klank and Michael Beetz, "Robotic grasping of unmodeled objects using time of flight range data and finger torque information", IEEE, 2010
- [19] G. Taylor and L. Kleeman, "Integration of robust visual perception and control for a domestic humanoid robot", IEEE, 2004
- [20] M. Taylor, A. Blake and A. Cox, "Visually guided grasping in 3D", IEEE, 1994
- [21] M. Trobina and A. Leonardis, "Grasping arbitrarily shaped 3-D objects from a pile", IEEE, 1995
- [22] 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, 2005

- [23]A. Namiki, Y. Nakabo, I. Ishii and M. Ishikawa , “high speed grasping using visual and force feedback”, IEEE,1999
- [24]H.Y. Jang, H. Moradi, S. Lee and J. Han, “A visibility based accessibility analysis of the grasp points for real time manipulation”, IEEE, 2005
- [25] Miikka Eloranta, Matti Laukkanen, Joonas Elovaara and Kristian Laakkonen,”Robot operating system (ROS) HYDRO driver development for a Kinova Jaco Arm”, 2014
- [26] Rectangle Representation Code and SHAF Framework available as ROS Packages, 2012. URL [http://pr.cs.cornell.edu/grasping/rect data/data.php/](http://pr.cs.cornell.edu/grasping/rect_data/data.php/).