HUMAN-ROBOT COOPERATIVE GRASPING

CHEN NUTAN

NATIONAL UNIVERSITY OF SINGAPORE
2012

HUMAN-ROBOT COOPERATIVE GRASPING

CHEN NUTAN
(B.Eng.(Hons.), DUT)

A THESIS SUBMITTED
FOR THE DEGREE OF MASTER ENGINEERING
DEPARTMENT OF MECHANICAL ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE
2012

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Chen Nutan

Chen Nutan
6 December 2012

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Chew Chee-Meng (Associate Professor, Department of Mechanical Engineering, National University of Singapore), for his constant support, invaluable suggestions, insightful comments and continuous encouragement during this research. My interests in the field of robotics started when I joined in Control and Mechatronics Laboratories under Prof. Chew. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Master study.

My sincere thanks also goes to my co-supervisor, Han Boon Siew (Senior Research Officer, Institute for Infocomm Research, A*Star), for providing the opportunity of developing the world famous robots, Meka and Hubo, to me, and leading me working on diverse exciting projects. His innovative ideas also stimulate me to try new equipments and methods.

Next, I would like to thank Tee Keng Peng (Doctor, Institute for Infocomm Research, A*Star), for his technical guidance. I gained the skills of research methodology and algorithm developments from him. His stimulating suggestions, encouragement and guidance helped me during the whole research.

Last but not least, I appreciate the staffs in Institute for Infocomm Research for all their support, help, and suggestions. Especially I am obliged to Wong Chern Yuen Anthony, Chua Yuan Wei, Yan Rui, and Chang Taiwen for the discussions, and for the days we were diligently working together before deadlines. In addition, I thank my seniors in Control and Mechatronics Laboratories of National University of Singapore: Huang Weiwei, Albertus Hendrawan Adiwahono, Shen Bingquan, Li Renjun, and Wu Ning for enlightening me at the first glance of research. With their invaluable advices, I could finish this project smoother.

Contents

Acknowledgements **ii**

Summary **vi**

1 Introduction **1**

1.1 Related Research Areas 2

1.1.1 Teleoperation 2

1.1.2 Autonomous Grasping 3

1.1.3 Human-robot Cooperation 5

1.2 Related Work 6

1.3 Motivation and Objective 8

1.4 Dissertation Outline 11

2 Telemanipulation System **12**

2.1 Human-teleoperated Pre-grasp Position 13

2.1.1 Human Arm Tracking 13

2.1.2 Human Hand Detection 16

2.1.3 Robot Control 19

2.1.4 Feedback System 25

2.2 Gesture Based Grasp Activation 26

2.2.1 Table And Objects Perception 26

2.2.2 Deciding How to Grasp 30

2.3 IR-based Grasp Assistance 34

2.3.1 Hardware–infrared Sensors with Robot Hand 35

2.3.2 Final Grasping Algorithm 36
2.3.3 Adjustment For Orientation 42

3 Experiment Results and Discussion 44
3.1 Robot Setting 44
3.2 Teleoperation Results 47
3.3 Experimental Results of Gesture-based Grasp Activation 50
3.3.1 Autonomous Grasping Results 50
3.3.2 Combined Method Results 51
3.4 Experimental Results of IR-based Grasp Assistance 56
3.4.1 Comparison of Full Teleoperation with Full Assistance 57
3.4.2 Ratio of Teleoperation to Assistance 58
3.4.3 Graspable Areas 66
3.4.4 Graspable Objects 67
3.4.5 Adjustment of Orientation 69
3.4.6 Tracking Mobile Object 71

4 Discussion 74

5 Conclusion and Future Work 76

Bibliography 78

A Appendix 87
A.1 Appendix 1 87

Summary

In the field of robotics, teleoperation and grasping make sense and play a valuable role, whether in terms of entertainment or practical application. It is meaningful for robots to imitate human and finish some tasks such as grasping. With these abilities, robots can really be developed to assist human in our daily life.

In order to enable robots to grasp a desired object through teleoperation, this thesis describes combined approaches of real time remote teleoperation and autonomous grasping for human-robot cooperation. For providing a user-friendly system with simple operation commands to grasp different objects successfully, vision-based teleoperation and autonomous grasping are combined. In the teleoperation process, motion tracking is carried out by Kinect in real time to detect the positions of human shoulder, elbow and hand joints such that the robot can imitate human. Hand gestures are detected by Kinect or AcceleGlove to control the robot hand gestures. Autonomous grasping is then employed, which is effective and generate more natural grasping gestures. The autonomous grasping subsystems are robust and can grasp both known and unknown objects. Two approaches of autonomous grasping are employed in this thesis. In the first approach, hand gestures are recognized as the switch of teleoperation and autonomy. At the end of teleoperation, the person closes his or her hand to send a signal to the robot, then the system convert to autonomous grasping. After that, the second subsystem begins for autonomous grasping for known objects, which contains table and object perception and grasp planning. In another approach, an algorithm allows us to perform online adjustments to reach a pregrasp pose without requiring premature object contact or regrasping strategies. We use three infra-red (IR) sensors that mounted on the robot hand, and design an algorithm that controls the robot hand to grasp objects using the information from the sensor readings and the teleoperator.

To satisfy the requirement, high performance Meka robot is used. The robot has 26 DOFs in total, 7 DOFs for each arm, 5 DOFs for each hand and 2 DOFs for torso.

Experiment results show that it is effective and user-friendly, and it has the capability to complete the missions of grasping of known and unknown objects. For known objects, it can convert from teleoperation to autonomy smoothly with simple commands from hand gestures of the user, and it can find the edge of objects and even track mobile objects for unknown objects using IR algorithm. Both of the two strategies enhance the success rates compared to pure teleoperation.

List of Tables

2.1 Modified Denavit-Hartenberg representation of the right arm of MEKA 20

3.1 Parameters of the arms of MEKA robot 45

3.2 Parameters of the hands of MEKA robot 46

3.3 Grasping success rates 55

3.4 Grasping success rates with different ratio of teleoperation and IR algorithm 60

3.5 Grasping success rates 68

List of Figures

1.1 Overview of the proposed system structure 9

2.1 Overview of the proposed teleoperation structure 13

2.2 Skeleton. The right figure shows the view from Kinect, and the left figure the human skeleton extracted from the sensor data. 14

2.3 Kinect 14

2.4 Flow chart of detecting joint data 15

2.5 Point cloud of hand 16

2.6 Flow chart of detecting hands 17

2.7 Hand detection. The right figure shows the hand detection result from the left figure. 17

2.8 acceleglove output signal convention (top view, right hand) [1] 18

2.9 MEKA robot [2] 19

2.10 MEKA robotics arm joints: (a)the posture shown 0 joint angles, arrows show positive rotation and torque directions; (b) Z axis is the axis of joint rotation [2] 19

2.11 Coordinate sets of a person 20

2.12 Shoulder-elbow on $\{X', Y', Z'\}$ frame 21

2.13 Shoulder-elbow on $Y - Z$ plane 22

2.14 Shoulder-elbow on $Y' - Z'$ plane 22

2.15 The whole arm on $\{X'', Y'', Z''\}$ frame 23

2.16 The whole arm on $\{X, Y, Z\}$ frame 23

2.17 MEKA robotics hand joint names and directions: the posture shown 0 joint angles, arrows show positive rotation and torque directions [2] 24

2.18 Feedback system 25

2.19 Table and object detection. The left image shows a table, a can and MEKA robot from Kinect RGB sensor. The right image shows the detection of the table and the can and robot model from RVIZ (A 3d visualization environment for robots using ROS) simulation. 30

2.20 Tool frame [2] 31

2.21 *Graspit*! simulation 32

2.22 Jerk trajectory 34

2.23 Hardware of IR sensors: The red points are the centers of sensor, and two edges of objects are supposed to be on the sensor 1 and sensor 3 centers as the blue rectangular. 36

2.24 Conrol block diagram. FK and IK denote forward & inverse kinematics maps respectively. 41

3.1 Teleoperation with virtual robot 47

3.2 Teleoperation with real robot. The first image shows the robot mimics the person to move the arms. The second image shows pre-grasp. The third image shows the robot mimics the person to grasp an object. . . 48

3.3 Degree of joints during a motion 49

3.4 Simulation on RVIZ. It shows the detection of the table, the can, and the robot model. The blue cube indicates the selected object. 51

3.5 Autonomous grasping. The first image shows pre-grasp. The second image shows the robot hand wraps an object. The third image shows the robot lifts the object. 52

3.6 Grasping object using the combined method. The first image shows the teleoperation. The second image shows the switching from the teleoperation to autonomous grasping. The third image shows autonomous grasping. 53

3.7 The position of the tool frame during grasping 54

3.8 Grasp postures. The left image shows grasping a can. The right image shows grasping a tape. 55

3.9 Teleoperation with and without the proposed IR Algorithm. The blue cylinder represents a soda can, and the points the pre-grasp positions. 57

3.10 2D vision feedback 58

3.11 Effort in grasp placement for different ratios of teleoperation to assistance. Mean values and standard deviations are shown. 62

3.12 Error in final position(e_f) for different ratios of teleop to assistance. Mean values and standard deviations are shown. 63

3.13 The position of IR sensors with respect to robot base frame during the person moves close to an object then moves away 64

3.14 The combined efforts of grasp placement and error recovery ability. Mean values and standard deviations are shown. 65

3.15 Graspable area maps verified experimentally. The shaded rectangle and circle are sample objects. End ”×” represents the position of the center of the surface of the object facing the IR sensors. 67

3.16 Failures of grasping 68

3.17 Object orientation test 69

3.18 Orientation result 70

3.19 Object trajectory detection 71

3.20 Tracking mobile object. The black rectangle is an object. A is a point on the corner of the object. The blue nearly rectangular line is the trajectory of the object. The green stars are the trajectory of sensor 3. The three red circles represent the three sensors. It is with respect to robot base frame. 71

3.21 Trajectories of sensor 3 and the object during the tracking task . . . 72

A.1 Grasp experiments: for every object, the top picture shows the initial position of the sensor detecting an object. The middle and the bottom pictures show the final grasp from front view and side view respectively. 88

A.2 Grasp experiments: for every object, the top picture shows the initial position of the sensor detecting an object. The middle and the bottom pictures show the final grasp from front view and side view respectively. 89

A.3 Grasp experiments: for every object, the top picture shows the initial position of the sensor detecting an object. The middle and the bottom pictures show the final grasp from front view and side view respectively. 90

Chapter 1

Introduction

The use of robots can be traced back to the end of the twentieth Century. With many years' trial and error and a large number of researchers' contribution, many world famous robots appear, such as ASIMO [3], PR2 [4] lightweight robot (LWR) [5] and Robonaut [6]. Nowadays, robots fulfill some tasks and perform meaningful services, from industrial manufacturing to healthcare, transportation, and exploration of the deep space and sea. It is an increasing research topic due to robots' useful applications and challenging potential. Interacting and working with humans, the robots will become a part of our lives.

1.1 Related Research Areas

This thesis describes a method in which a robot imitates a human in performing the task of grasping. The three fields related to the work done in this thesis are teleoperation, autonomous grasping and human-robot interaction. This section describes the related research areas.

1.1.1 Teleoperation

Teleoperation might be one of the first applications of robotics outside of industry and military.

Teleoperation means operating a robot at a distance, where a human operator is in control or human is in the loop [7]. A variety of interface components have been used for teleoperation. For example, the phantom [8], which is a device used for controlling end-effectors, provides force feedback to the user. Other traditional devices such as dataglove [9], exoskeletons [10], electromyographic muscular activity sensors [11] and inertial sensors [12] which require intricate instruments or complex operations. The most common devices used for teleoperation are the mouse and keyboard.

Vision-based teleoperation [13,14] have gained popularity in recent years as they are more portable and do not require that the user wear any special equipment. However, the limitation of vision-based method is that only partial information of human can be collected due to occlusion.

1.1.2 Autonomous Grasping

There are two main classes of object - known and unknown objects. A known object is one identified by priori information whereas an unknown object does not. Particularly, if both the shape and the appearance of the object are previously known, this object is assumed as a known object. These shape and appearance are employed for relevant grasping strategies through development [15] or supervised learning [16]. On the contrary, unknown objects require more general methods due to the uncertainties in the objects to generate suitable grasping points.

Although a variety of concepts and methods for grasping have been developed during the last decades, grasping an unknown object still remains a challenging problem. For known object where a full 3-D model can be obtained, various more robust approaches can be used for grasping. For example, methods based on friction cones, or form-closure and force-closure [17], or pre-stored primitives [18], etc. can be applied.

In reality, due to uncertainties of sensor limitations and unpredicted environment conditions, it is difficult to obtain a full 3-D model of an object through stereo camera or other sensors such as a laser scanner. Therefore, it is necessary to extract grasp points from partial information of an object.

Some approaches have been explored to grasp unknown objects. An approach, described in [19], of partial features predicts the grasp position of unknown objects using 2D images. However, one grasp point is defined per object, which is not general and may result in an unstable grasp. A system for grasping objects using unknown geometry [20] was developed. This system requires a 360 degrees scan of an object using a laser scanner on a rotating disc. This method is time-consuming and requires that the object be placed on the rotary disc.

A framework of automatic grasping of unknown objects by a laser scanner and a simulation environment is shown in [21]. Another method [22] combining online silhouette and structured-light generates a 3D object model with a robust force closure grasp. However, only several simple objects have been tested for both [21] and [22], which cannot demonstrate that they are suitable for complicated and general objects.

A vision based approach was presented in [23]. Object information was obtained using monocular and binocular visual cues and their integration. Curvature information [24] was obtained from the silhouette of the object. The pose of the robot is then updated and a suitable grasping configuration is achieved by maximizing the curvature value. A strategy for grasping unknown objects based on co-planarity and color information was developed in [25]. However, the environments in [25] are simple, which cannot be applied to the real world.

1.1.3 Human-robot Cooperation

Human-robot cooperation is useful for performing special tasks in dangerous, distant or inaccessible environments in military missions such as clearing nuclear waste [26] and defusing a bomb. They are also useful for applications such as serving elderly and disabled people [27]. Such systems take advantage of the ability of both the human and the robot. They reduce human workload, costs, fatigue-driven error and risk [28], and augment human's abilities. Hence, given the present state of robotics, it is one of the fundamental methods for controlling robots.

In the applications stated above, there is synergy between robots and human. They share a workspace and goals in terms of achieving the task. This close interaction needs new theoretical models–there is need to improve a robots utility while evaluating the risks and benefits of this robot for modern society.

There are many investigative studies on robot assistive technology for many applications. Specifically, robots are studied as tools to aid in daily tasks, act as guides and becoming assistants with high communication behavior [29, 30].

The concept of human and robots sharing a common intent without complex communication was mentioned in [31]. The system consists of perception, recognition and intention inference. The result of the study was positive although toys represented robots.

Teleoperation with haptic feedback was developed to achieve a more natural and effective method for human-robot cooperation. This method of interaction allowed for a more ecological interface [32,33]. Both the human operator and the robot share control depending on the situation. This system is more intuitive for human operators and has proven to be more effective.

Another method of collaboration is to treat the human as a robot assistant while the robot acts autonomously [34,35]. The robot works autonomously until it encounters a problem, where the robot will seek assistance from a person. Alternatively, the robot performance could be improved through human suggestions.

Recently, Robonaut, an assistant humanoid robot designed by NASA [36], was sent to outer space. Robonaut was teleoperated remotely with force feedback integrated.

1.2 Related Work

There are few approaches that combine teleoperation with autonomous grasping [37, 38]. Although there exist some combined approaches for other robot control, such as local autonomous formation control [39], and event-based planning [40], they are not for remote grasping.

Middle or long range sensors such as laser scanner [41] and stereo cameras [42] can detect and localize objects fairly accurately, but they are not suitable for teleoperation.

Firstly, occlusion by the robot arm may occur during manipulation of the object. Secondly, image acquisition and processing are generally not fast enough for online reactive response in unstructured environments (e.g. when the object is moving).

Tactile sensors are employed as well in the following methods [43, 44], but they are contact based sensing methods. For teleoperation, contacting objects is almost as difficult as grasping them. Besides, it lacks sensitivity and hence not suitable for teleoperation.

Short range stereo cameras [45] mounted on the end-effector were developed. However, they have a narrow field of view and cannot be positioned at short distances to the object. Otherwise, there may be no good grasp points. In addition, the camera is being used for grippers, and might fail for humanoid hand due to the larger width, which might cause occlusion of the object.

Optical infrared sensors have also been employed for final grasp adjustments. Although they have less information compared to cameras or lasers, but they are less sensitive to environmental changes and require less computation. The method in [46] detects the orientation of an object surface using the IR sensors that fit inside the fingers. Continuous Shared Control [47] combines brain signal and IR sensors to grasp object. However, [46] can only adjust the fingers, and [47] can only be used for one dimension of the end-effector. [48] equips IR sensors on a gripper to adjust the griper for a normal force to the object boundary. However, the objects are supposed to be in the gripper before applying the method and it uses logic approach which is discrete control.

1.3 Motivation and Objective

Current autonomous robots cannot meet real life expectations because of their limited abilities for manipulation and interaction with humans. These robots could fulfill some simple tasks, but the process may be time-consuming. Moreover, robots cannot handle changes well without user intervention. With teleoperation, robots can receive human's commands in real time under human's assistance to execute tasks. However, teleoperation also has limitations. For example, simple teleoperation systems may not be able to collect sufficient information from a person resulting in robots ignoring some important tasks. On the other hand, teleopoeration systems that acquire more data require intricate instruments or complex operations. In addition, even if the systems can obtain full user information, it is hard to use in the real world without a trained person to operate the robots.

In order to overcome these challenges, we develop two combined approaches, which provide user-friendly operation. The first approach contains two subsystems for grasping known objects (see Fig. 1.1).

The teleoperation subsystem enables the end-effectors to be brought close to the desired object. In this system, the information from the Kinect sensor is continuously detected from the human joints and sent to the robot control system. As the bridge of

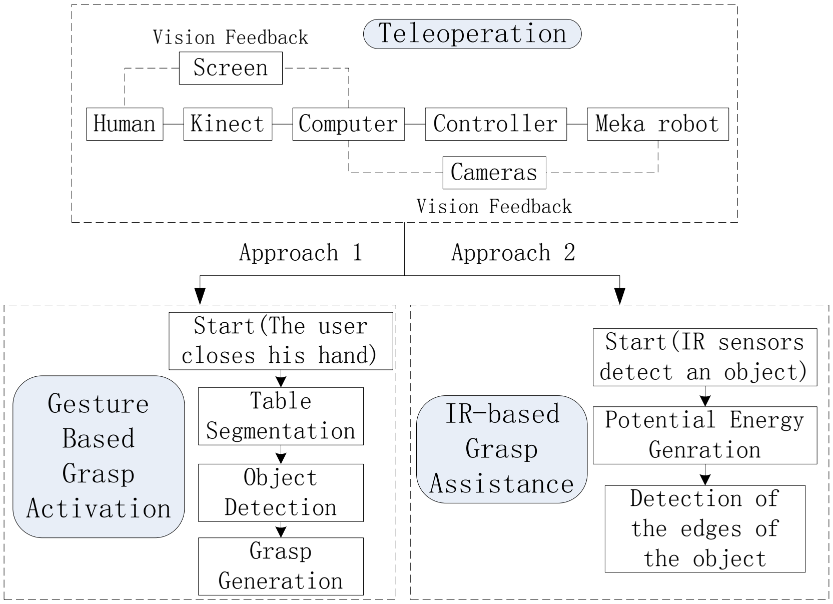

Teleoperation
Vision Feedback
Screen
Human
Kinect
Computer
Controller
Meka robot
Cameras
Vision Feedback
Approach 1
Approach 2
Gesture Based Grasp Activation
Start(The user closes his hand)
Table Segmentation
Object Detection
Grasp Generation
IR-based Grasp Assistance
Start(IR sensors detect an object)
Potential Energy Genration
Detection of the edges of the object

Figure 1.1: Overview of the proposed system structure

the two subsystems, hand gestures play a fundamental role in the switching. When the person closes his hand, a signal is sent to the robot to switch to autonomous grasping. The autonomous grasping subsystem contains table and object perception and grasp planning.

Another approach is final grasping correction for teleoperation using three IR sensors. It is effective, light, robust, small size and cheap which are desired qualities for assisting grasping in teleoperation. We chose the minimum number of sensors that yields the most useful information, thus reducing the size of the structure and simplifying the algorithm. Three sensors are mounted on the robot hand for localizing a nearby object and providing error signals that drive the hand in three dimensions based on a potential energy algorithm. The global minimum potential energy could be calculated in order to look for a grasp point, as well as teleoperation also affects the trajectory of the hand; therefore, the combined result enables the end-effector to follow teleoperation and track the object at the same time.

In this work, Meka robot [2] is chosen, whose manipulators are 7 DOF arms, end-effectors are 5 DOF hands, and body is a 2 DOF torso.

Main features of the proposed telemanipulation system:

1, cooperative grasping. It combines the advantages of having human initiative, and the accuracy and robustness of the robotic system.

2, need a little object knowledge. The proposed IR strategy is to grasp objects with limited perception data. The three IR used in the current approach only gives one dimension data.

3, online adjustment. It allows online grasp adjustments to estimate a suitable grasp point of unknown objects without requiring premature object contact or regrasping strategies.

4, robust grasping. The system is robust when grasping a wide range of objects and even tracks mobile objects.

5, portable interface component. It employs the low cost Microsoft Kinect as an interface instead of other higher end equipment for human motion capture.

1.4 Dissertation Outline

The structure of this thesis is as follows.

Firstly, Chapter 2 gives an overview of the teleoperation and autonomous grasping systems and provides a detailed description of the experimental methods, including different teleoperation techniques

In particular, it consists of three subsections. Subsection one describes teleoperation in this system using Kinect sensor. Both the second and third subsections looks into autonomous grasping, describing different approaches for grasping known and unknown objects respectively.

In Chapter 3, the results of the experiments are discussed. A comparison of the two different methods - teleoperation grasping and the combined method grasping, is done. The results demonstrate that the combined method is more effective and has a higher success rate.

Next, Chapter 4 discusses the experimental results and demonstrates the contribution of this thesis.

Finally, Chapter 5 concludes with a section discussing the future work for better human-robot cooperation.

Chapter 2

Telemanipulation System

To obtain the benefits of human dexterity and robot accuracy, the telemanipulation systems consist of human teleoperation and robot autonomous grasping. Direct position control is carried out for teleoperation. The data detected from human joint positions are transformed to the shoulder and elbow joint angles after inverse kinematics. For the first apporach, the gestures of hands are also detected continuously. Once one of the hands of the human is closed, the control is switched to autonomous grasping immediately. The robot then detects the nearest object to the tool frame on the table in front of the robot and generates a suitable grasp for the object. Alternatively, another approach of autonomous grasping that adjusting final grasp using short range Infra-Red porximity sensors is also presented. In this IR approach, sensors can search for the edge of an object. Teleoperation and IR signal have combined contribution to the robot grasping.

2.1 Human-teleoperated Pre-grasp Position

The proposed teleoperation system consists of a person, a MEKA robot and two Kinect sensors. The main purpose of this subsystem is to enable the end-effector to be brought to a good position close to an object for autonomous grasping. In this system, the person is at a local place while the robot is at a remote place interacting with the environment. The process includes three steps:

(1) human arm tracking;

(2) human hand detection;

(3) robot control.

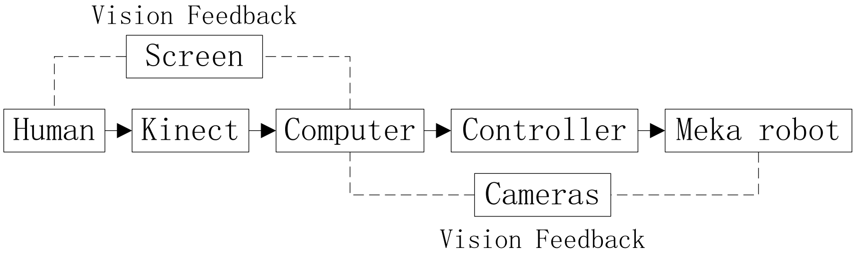


Figure 2.1: Overview of the proposed teleoperation structure

2.1.1 Human Arm Tracking

Motion tracking is a process for generating human skeleton information which contains position data of human joints (see Fig.2.2). It is carried out using Kinect which has a color image CMOS sensor and depth sensor including an infrared laser projector and a monochrome CMOS sensor as shown in Fig.2.3.

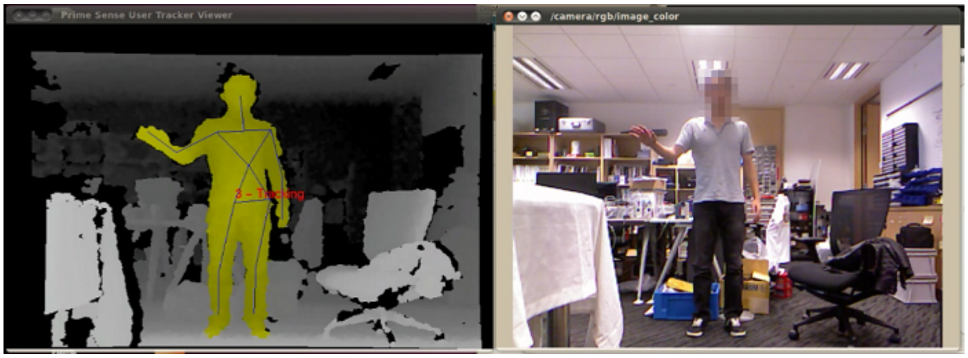


Figure 2.2: Skeleton. The right figure shows the view from Kinect, and the left figure the human skeleton extracted from the sensor data.

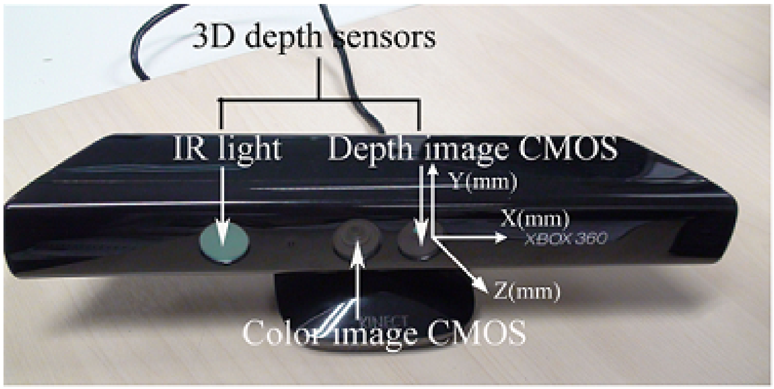


Figure 2.3: Kinect

As shown in Fig.2.4, Human joint detection mainly contains three processes, user generation, pose detection, and skeleton generation. In the process of user generation, human in the view of depth camera are detected using 3D raw data from sensors. In this process, the main functionalities includes: (1) Detecting the current number of users; (2) Calculating the center location of users' mass; (3) Tracking a new user. Pose detection enables the system to check if the user is in the specific position. Calibration will start after specific positions are detected. In this work, we choose T-shape pose as this specific pose. Skeleton generation calibrates the human skeleton, gets the joint

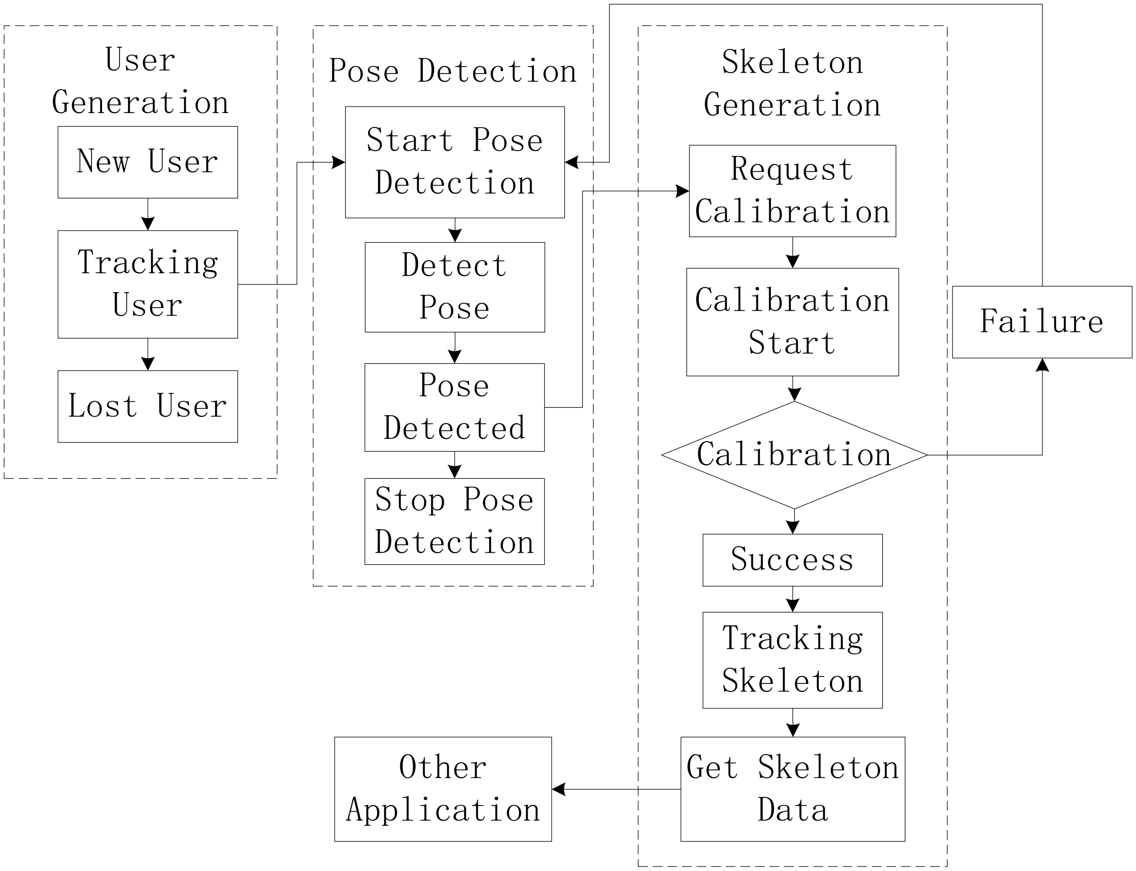


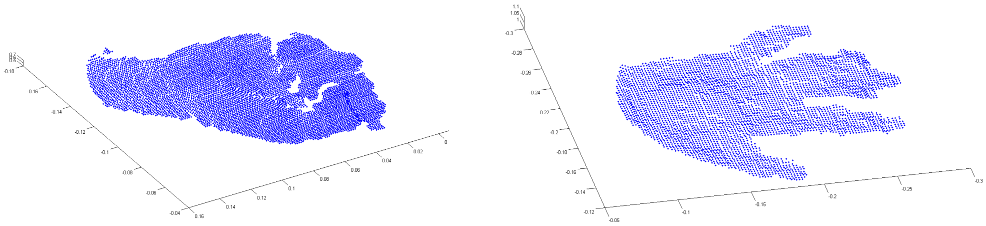
Figure 2.4: Flow chart of detecting joint data

data continuously using tracking algorithm, and enables the application to transfer the data to other applications [49]. In these processes, data of human joints is generated using Open Natural Interaction (OpenNI) which communicates vision/audio sensors and NITE, a motion tracking middleware.

The Kinect sensor could recognize 15 joints using the middleware of NITE, including head, neck, torso, left shoulder, left elbow, left hand, right shoulder, right elbow, right hand, left hip, left knee, left foot, right hip, right knee, right foot. For every joint, the data contains the position of the human joints, the orientations, and confidences.

In this project, the joints of the arms are employed for robot manipulators.

2.1.2 Human Hand Detection



(a) Point cloud of closed hand

(b) Point cloud of open hand

Figure 2.5: Point cloud of hand

By hand detection process, hand point clouds and hands' states can be provided [50]. We assume the palm faces to the camera. The first step is to detect the hand's position by the skeletal information getting from the above human tracking. The next step is to estimate the distance of a point of raw data and the hand position. If the distance is less than a threshold, the point will be deemed as one element of point cloud of the hand and split from the whole point cloud of the human. Lastly, two states of the gestures (closed and open) are identified. The centroid of the hand point cloud is calculated before computing normalized 3×3 covariance matrix ζ of the set of hand points in 3 dimensions. The eigenvalues of the covariance matrix reflects the distribution of the points in the main directions. In other words, the eigenvalues represent the three dimensions of the hand point cloud which contains the main information of the geometry of the point cloud. Therefore the 2 highest eigenvalues $eig_1(\zeta)$ and $eig_2(\zeta)$ are used for detecting the hand state.

$$eig_2(\zeta)/eig_1(\zeta) = c \tag{2.1}$$

where, if the ratio c is less than a threshold, the point cloud is considered as a long shaped object, and vice versa. Based on the experiment, c is chosen to be 0.4.

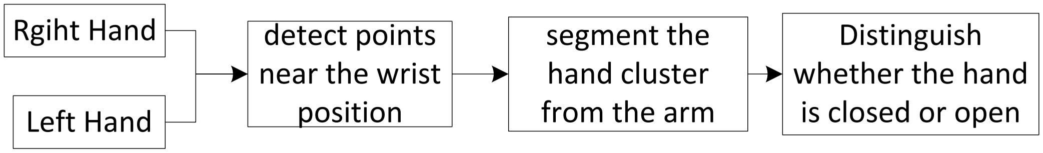

Rgiht Hand
Left Hand
detect points near the wrist position
segment the hand cluster from the arm
Distinguish whether the hand is closed or open

Figure 2.6: Flow chart of detecting hands

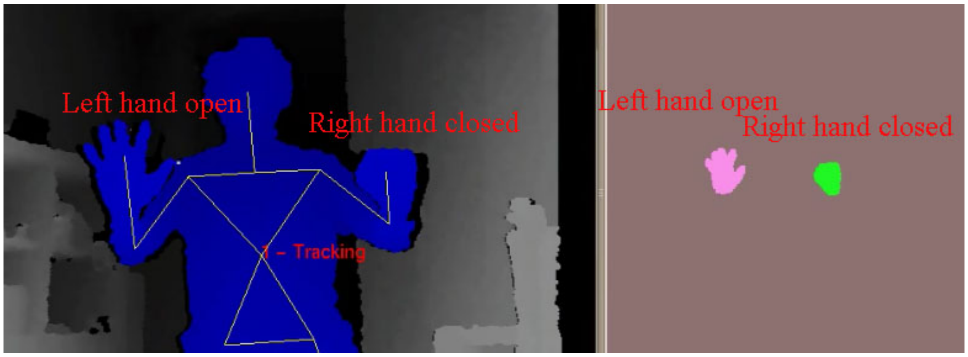


Figure 2.7: Hand detection. The right figure shows the hand detection result from the left figure.

Acceleglove (AnthroTronix, Inc.) is employed as another method of hand detection to compare with using Kinect. Acceleglove could detect the finger and hand information to be used for some applications such as controlling robots, video games, and simulators [51]. In this experiment, the Acceleglove enable human and robot to interact with each others.

First of all, the data of the fingers and palm are detected through 6 accelerometers. Every accelerometer outputs x, y and z signals. If the glove is horizontal, z axis is along gravity vector, as well as x and y axis parallel to a horizontal plane (see Fig. 2.8). When x axis is rotated or there is an acceleration in the y direction, the value of y axis would be changed. Analogously, when y axis is rotated or there is an acceleration in the x direction, the value of x axis would be changed.

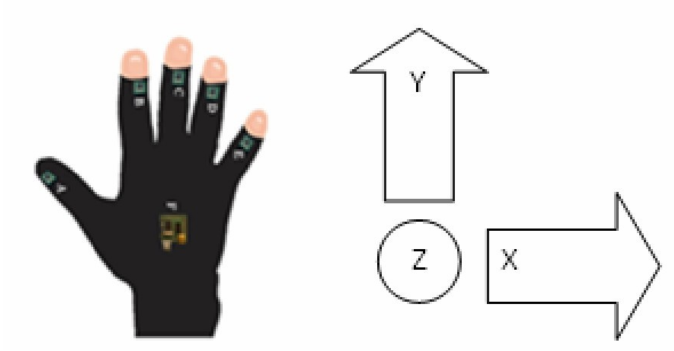


Figure 2.8: acceleglove output signal convention (top view, right hand) [1]

After detecting the data from sensors, the gestures could be trained recognized using the raw signals. In the process, four gestures, *start, stop, closed* and *open*, are recorded for each hand in a library. Besides, around 20 group data are recorded for every gesture, because more instances trained for one gesture could enhance the recognition confidence. A probability filter is employed to recognize the gestures; therefore, only if the probability is more than the threshold can the gesture be recognized.

At last, the gesture information are transferred to the robot using TCP/IP, because Acceleglove is used under Windows, while Robot uses ROS under Linux in this experiment.

2.1.3 Robot Control

In this experiment, MEKA robot is used for executing the tasks assigned by the person.

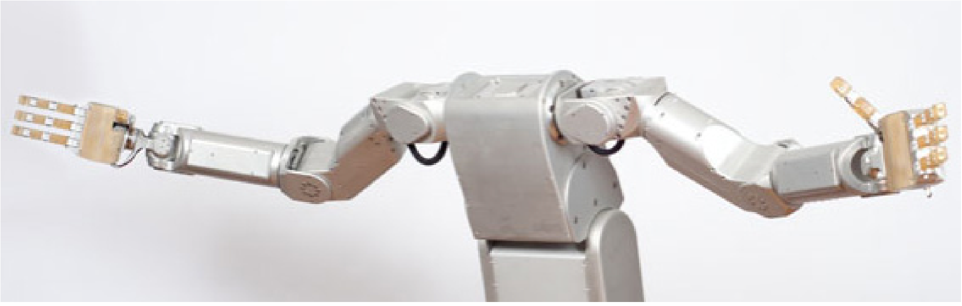
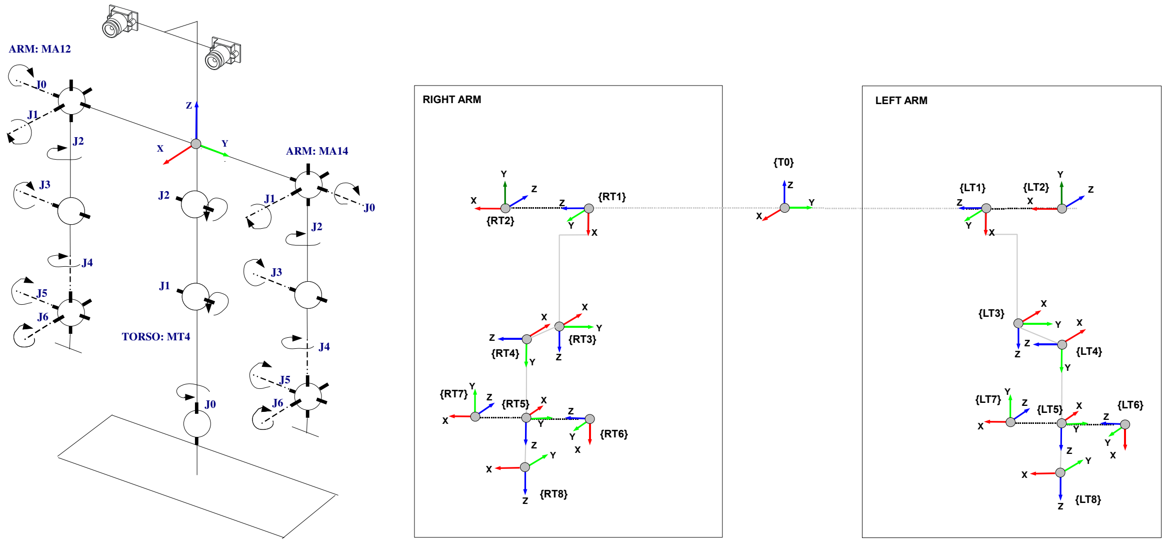


Figure 2.9: MEKA robot [2]



(a) Joint names and directions

(b) Kinematic frames

Figure 2.10: MEKA robotics arm joints: (a)the posture shown 0 joint angles, arrows show positive rotation and torque directions; (b) Z axis is the axis of joint rotation [2]

Direct position control is used for the arm motion tracking. The MEKA robot (see Fig.2.9) is a humanoid robot with 7 DOFs for each arm. Its coordinate sets can

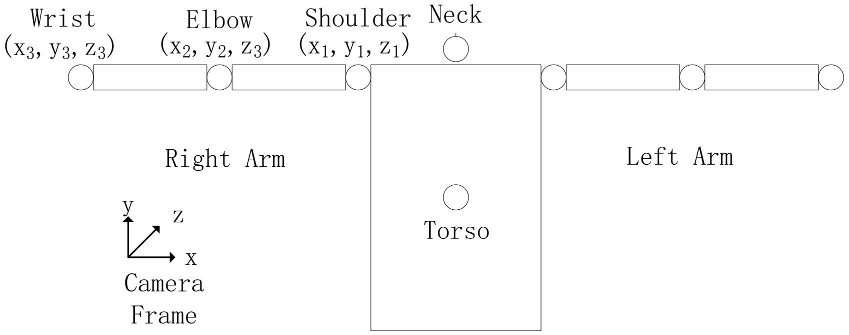


Figure 2.11: Coordinate sets of a person

Table 2.1: Modified Denavit-Hartenberg representation of the right arm of MEKA

Link	a(m)	α	d(m)	θ
1	0	90^o	0.18465	-90^o
2	0	90^o	0	90^o
3	0.03175	90^o	0.27857	90^o
4	-0.00502	90^o	0	0

be seen from Fig.2.10. Two methods are usually provided for transforming human joint data to a robot. One is transforming orientations of the main human joints to the relative robot joints. Another one is to calculate the robot's joints from the end-effectors of human hands. As different individuals have different arm sizes, the second method may cause end-effectors out of range. Hence, the first method is employed in this experiment. The data detected from human joint positions are transformed to the robot shoulder and elbow joint angles. γ_i is the the orientation of joint J_i of the robot arm. As seen from Fig.2.11, the right arm angles are determined as follows.

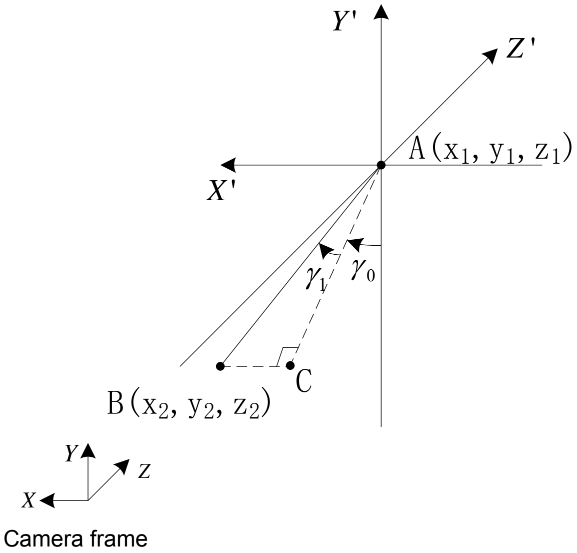


Figure 2.12: Shoulder-elbow on $\{X', Y', Z'\}$ frame

In 2.12, frame $\{X', Y', Z'\}$ is parallel to the camera frame with (x_1, y_1, z_1) as the origin. On the $Y - Z$ plane (see 2.13), we can get,

$$\gamma_0 = atan2(z_1 - z_2, y_1 - y_2) \tag{2.2}$$

As can be seen from 2.14, AC is the projection of AB on $X' - Y'$ plane. AC_1 and AC_2 are projections of AC on z axis and y axis respectively. AD_1 and AD_2 are projections of AC_1 and AC_2 on AC. Thus,

$$\vec{AC} = \vec{AD}_1 + \vec{AD}_2 \tag{2.3}$$

then,

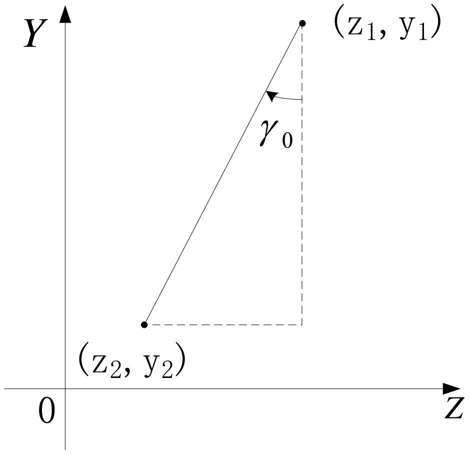


Figure 2.13: Shoulder-elbow on $Y - Z$ plane

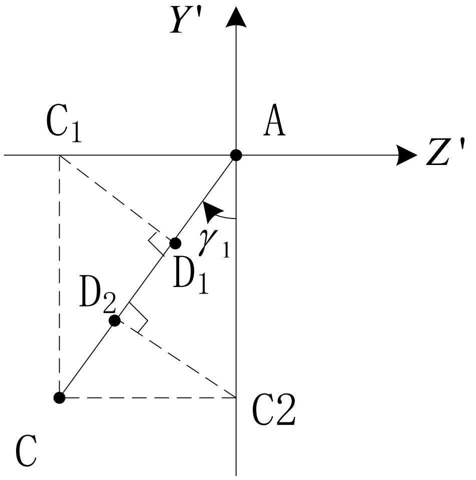


Figure 2.14: Shoulder-elbow on $Y' - Z'$ plane

$$AC = (z_2 - z_1)\cos(\pi/2 - \gamma_1) + (y_2 - y_1)\cos(\gamma_0) \tag{2.4}$$

$$\begin{aligned}\gamma_1 &= atan2((x_2 - x_1), -AC) \\ &= atan2((x_2 - x_1), ((z_1 - z_2)\cos(\pi/2 - \gamma_1) + (y_1 - y_2)\cos(\gamma_0)))\end{aligned} \tag{2.5}$$

From 2.15, frame $\{X'', Y'', Z''\}$ is parallel to the camera frame with (x_2, y_2, z_2) as the origin, and BF is projection of BE on $X'' - Z''$ plane. γ_2 can be computed as follow,

$$F = Rotz(\gamma_0)^0_1RRotz(\gamma_1), \tag{2.6}$$

$$P = F^{-1}V_{32}, \tag{2.7}$$

$$\gamma_2 = atan2(-P_x, -P_z), \tag{2.8}$$

where V_{ij} is the vector points from frame i to frame j, $P = [P_x, P_y, P_z]$, and

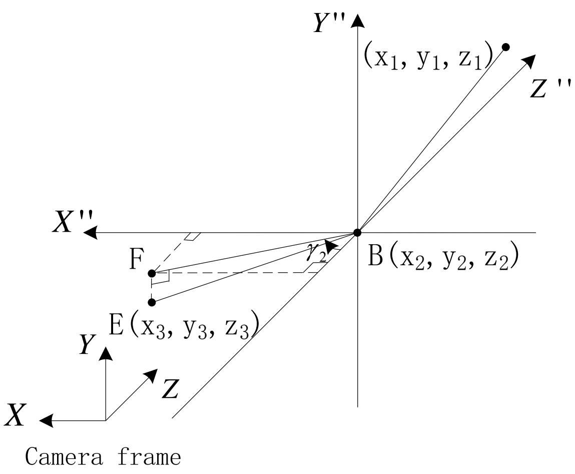


Figure 2.15: The whole arm on $\{X'', Y'', Z''\}$ frame

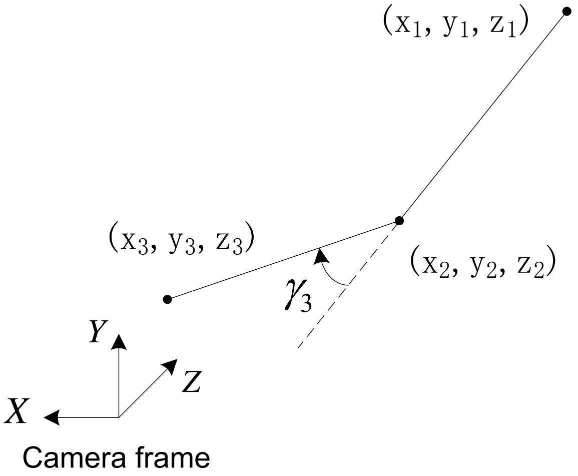


Figure 2.16: The whole arm on $\{X, Y, Z\}$ frame

$$Rotz(*) = \begin{pmatrix} cos(*) & -sin(*) & 0 \\ sin(*) & cos(*) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.9}$$

We only consider the angles of the joints, so that the frame transformation from joint 0 to joint 1 can be represented by,

$${}^{0}_{1}R = \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{pmatrix} \tag{2.10}$$

As shown in From 2.16,

$$\gamma_2 = acos(V_{12}V_{23}) \tag{2.11}$$

In order to achieve smooth motion for robot, mean filter is employed to reduce noise of the robot joint data. The positions calculated after the above algorithm at time t_1, t_2, t_3, ..., t_i, t_{i+1}, t_{i+2}, ..., t_{i+k}, ..., are denoted by $q(t_1)$, $q(t_2)$, $q(t_3)$, ..., $q(t_i)$, $q(t_{i+1})$, $q(t_{i+2})$, ..., $q(t_{i+k})$, ..., where $q(t_i)$ is a vector of the 4 joints' angles.

$$q(t_i) = [\gamma_1, \gamma_2, \gamma_3, \gamma_4]_i^T \tag{2.12}$$

Then, the positions transfer to the robot at t_{i+k} are:

$$qd(t_{t+k}) = \frac{\sum_{j=1}^{k} q(t_{i+j})}{k} \tag{2.13}$$

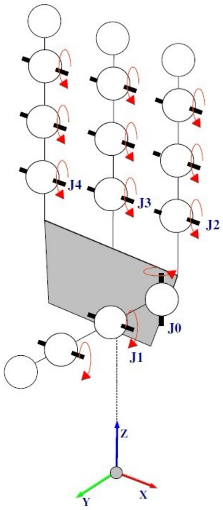
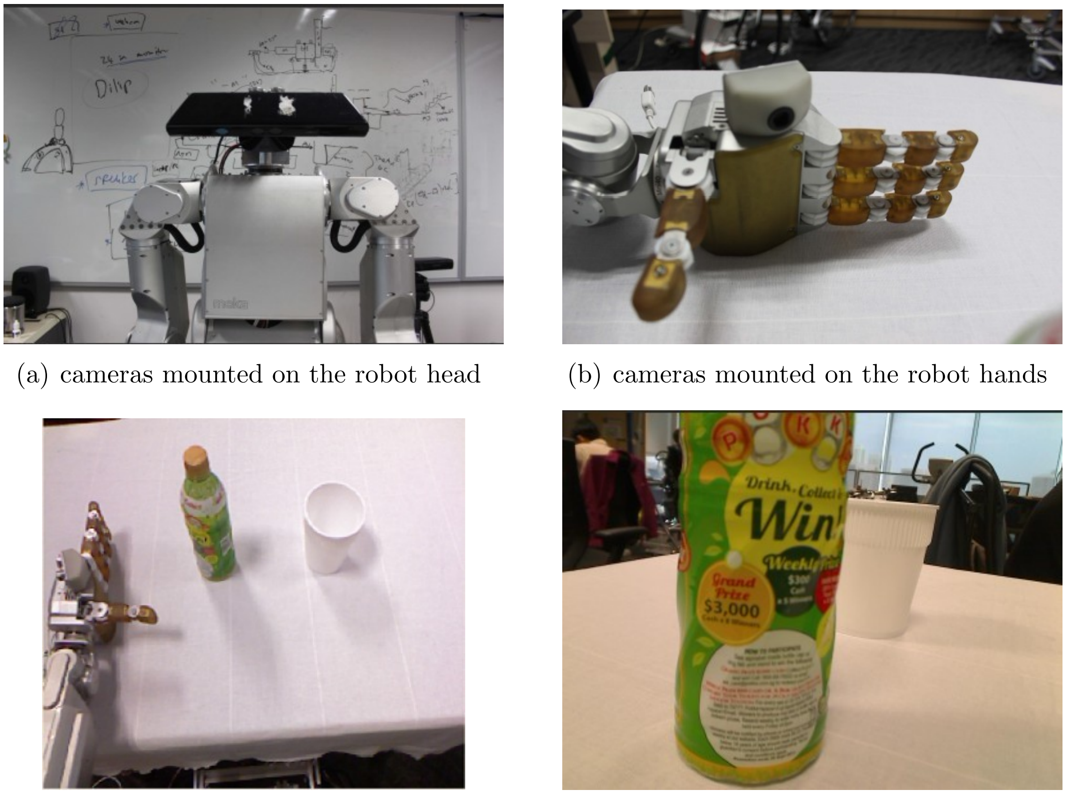


Figure 2.17: MEKA robotics hand joint names and directions: the posture shown 0 joint angles, arrows show positive rotation and torque directions [2]

Robot hands are mainly for grasping objects after the arm has reached the desired position. As seen from Fig.2.17, each hand has 5 DOFs, including 3 fingers driven by 3 tendon(J2, J3, J4) and a 2-DOF thumb driven by a tendon(J1) and directly driven by a motor(J0) [2]. The encoders measure the tendon position instead of every joint position. Under the circumstances, the true finger pose could neither be measured nor uniquely determined. Thus, there are only two gestures of the hands (closed and open) in this thesis.

2.1.4 Feedback System



(a) cameras mounted on the robot head

(b) cameras mounted on the robot hands

(c) vision feedback from the camera of the robot head

(d) vision feedback from the camera of the robot hand

Figure 2.18: Feedback system

Visual feedback of the robot environment is employed during teleoperation, which displays the robot environment to the user. As shown in Fig.2.18, in the feedback system, three cameras at the robot environment obtain the views from the robot head and hands, and a monitor at local displays the three views. One of the three cameras is mounted on the head of Meka robot as robot's eyes, and the rest(Point Grey) are equipped on the end-effectors besides the thumbs. The feedback images are shown in Fig.2.18 (c) and (d).

2.2 Gesture Based Grasp Activation

The teleoperation behaviors attempt to move the manipulators such that the hand is near an object and send a signal to start autonomous grasping. The grasping behavior will search for the nearest object besides the robot hand using another Kinect, and consequently generate a suitable position and orientation to grasp it.

2.2.1 Table And Objects Perception

Before generating grasping strategies, the robot has to get the information of the objects. In this experiment, all graspable objects are assumed on a table. Therefore, the table should be detected, then the objects can be separated and be identified.

All data are point cloud generated from a Kinect sensor mounted on a table. Environment perception contains two main components:

- table detection: it could segment the table, and extract the clusters above the table and remove the useless information.

• object detection: recognize what are the clusters above the table.

Three assumptions in environment perception process:

• all the graspable objects are on a table which is the dominant plane in the robot environment.

• the distance between two objects is more than 3 cm.

• alternatively, the objects for grasping are upright on the table.

A table is detected by finding the dominant plane in the point cloud using RANSAC (Random Sample Consensus) [52], which is an algorithm for robust fitting of models in the presence of many data outliers [53]. A table can be detected no matter what color it is. We assume:

• a plane can be estimated by n points which are not in a line, where $n = 3$.

• the number of the pixels detected from sensor in total is M.

• p is the probability of a randomly selected points which are in a refined model.

• p_{fail} is the probability of a randomly selected points which are in an unrefined model.

• k is the number of iteration.

Then, the algorithm:

• selects n points at random and estimates the plane

• finds how many points(l) of M fit the model (plane) with a given tolerance.

• if l is big enough, the model is considered as a fit model.

• repeat above processes k times if not success.

Consequently,

$$p_{fail} = (1 - p^n)^k \tag{2.14}$$

then take the logarithm of both sides,

$$k = \frac{log(p_{fail})}{log(1 - p^n)} \tag{2.15}$$

After segmenting the table, the objects which are placed on the table can be recognized. The objects are separated as clusters, which are the points corresponding to an individual object. If a cluster matches a mesh (see Fig. 2.19) from a database which contains a large number of common objects, this point cloud will be given a unique ID and it is considered as a known object. All unknown objects which cannot match any model in the database are filtered for reducing noises and avoiding grasping an uncommon object. In addition, every object's position with reference to camera frame is stored according to their ID.

ICP (Iterated Closest Point) is employed for aligning two free-form shapes of a cluster and a mesh [54]. 4 dimensions of an object are fixed, since the positions of the object can be detected from the sensor and it is upright on the table as the assumptions; therefore 2D ICP are used to identify the object. There are two sets of points, and the elements of model mesh and a cluster are $\{m_i\}_{i=1}^{N_m}$ and $\{c_i\}_{i=1}^{N_c}$ respectively. The aim is to solve the transformation matrix T, which can best match the mesh and the cluster. In this experiment, the registration is 2D under Euclidean transformation.

The 3 parameters are rotation θ and translation (p_x, p_y). There are several steps in the algorithm,

(1) start from initial transformation

(2) for every point on $\{c_i\}_{i=1}^{N_c}$, find the closest point on $\{m_i\}_{i=1}^{N_m}$.

(3) find best transformation for this correspondance.

(4) update the transformation.

(5) iterate step (2)-(4) using the parameters of θ, p_x and p_y.

Then, the transformation function is

$$T_{2D}(a) = \begin{pmatrix} cos\theta & sin\theta & p_x \\ -sin\theta & cos\theta & p_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2.16}$$

where a is a vector of $[\theta, p_x, p_y]$.

The error function is

$$e^2 = \frac{1}{N_c} \sum_{i=1}^{N_c} ||match_m(x_i) - T_{2D}(a) * x_i|| \tag{2.17}$$

where, x_i is the position of a point in $\{c_i\}_{i=1}^{N_c}$ which is in $\mathbb{R}^2$, $match_m(x_i)$ is the closest point from $\{m_i\}_{i=1}^{N_m}$ to x_i with Euclidean distance.

To search for the best transformation matrix, we can find vector a that minimize the error.

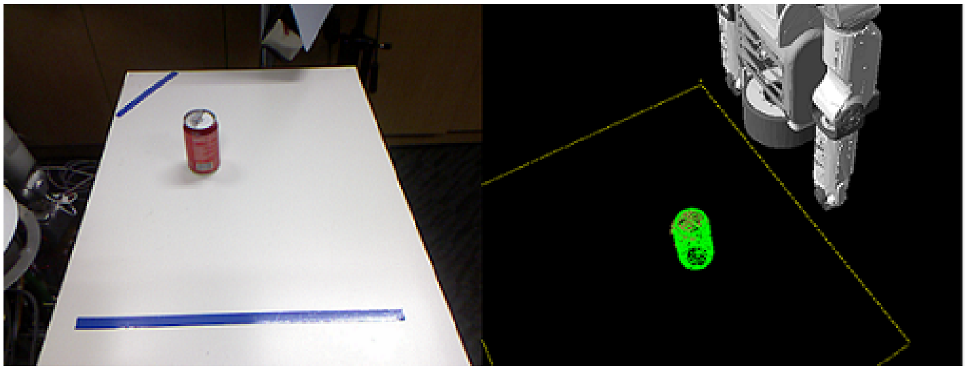


Figure 2.19: Table and object detection. The left image shows a table, a can and MEKA robot from Kinect RGB sensor. The right image shows the detection of the table and the can and robot model from RVIZ (A 3d visualization environment for robots using ROS) simulation.

2.2.2 Deciding How to Grasp

After getting the object information, the object positions are converted from camera frame to robot base frame and the nearest object from the end-effector can be detected.

The calibration of the camera frame and the robot base frame is designed before the experiment. Four non-coplanar points are chosen, then their positions can be collected in both the camera frame and the robot base frame using a known object. In the camera frame, the centroid position of the object point cloud is considered as the position of the object. In addition, when the end-effector grasps the object manually, the centroid point position of the object with respect to robot tool frame could be calculated using a grasping software, *Graspit*!, which will be explained in the next paragraph. Hence, the object position with respect to the robot base frame can be computed though the frame relationship of the robot. Lastly, the transform matrix of the camera frame and the robot base frame ${}^{C}_{R}T$ can be calculated as follows:

$$ {}^{C}_{R}T = [{}^{R}P_1 \ {}^{R}P_2 \ {}^{R}P_3 \ {}^{R}P_4][{}^{C}P_1 \ {}^{C}P_2 \ {}^{C}P_3 \ {}^{C}P_4]^{-1} \tag{2.18} $$

where R and C denote the robot base frame and the camera frame respectively, and $P_i = [x_i, y_i, z_i, 1]^T (i \in \{1, 2, 3, 4\})$.

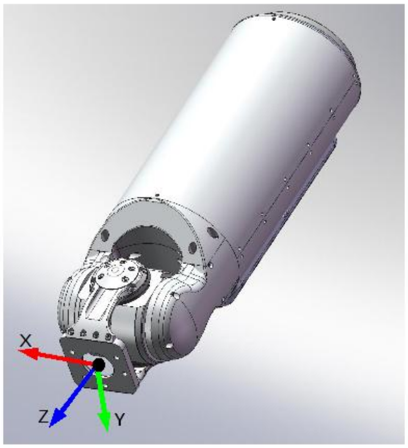
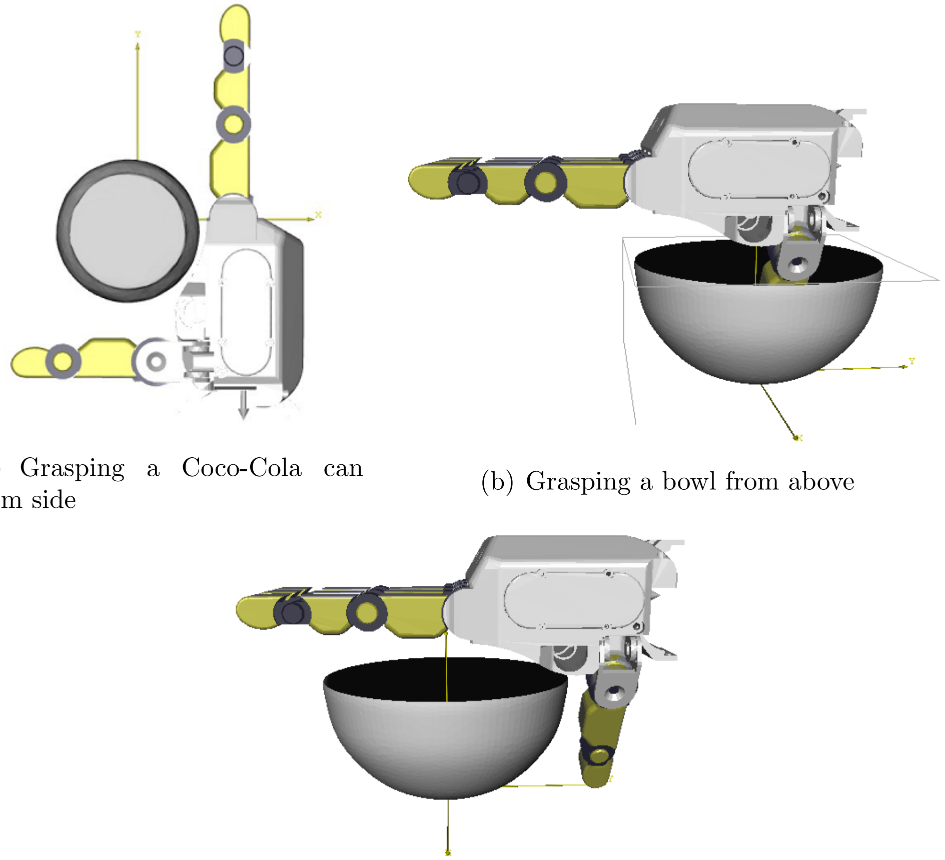


Figure 2.20: Tool frame [2]

Then it is necessary to generate a grasp pose for the hand. To do this, the called software *Graspit*! [55] is employed to compute the position and orientation relationship between a hand and an object. However, in order to grasp with natural manipulation motion, only the position of the tool frame is used from *Graspit*!, while the orientation is generated through the object position with respect to the robot shoulder. Specifically, the x axis of the tool frame (see Fig. 2.20) is vertical to the table and the z axis of the tool frame is from the shoulder to the object that is projected onto the table plane. For different classes of object, there are different grasp poses which are stored in the database in advance. For example, a can grasped from the side is preferred, while a short object such as a bowl should be grasped from above (see, Fig. 2.21).



(a) Grasping a Coco-Cola can from side

(b) Grasping a bowl from above

(c) Grasping a bowl from above

Figure 2.21: *Graspit*! simulation

After the grasp pose is generated, inverse kinematic is used for the end-effector reaching the pre-grasp position. For the purpose of maximizing the grasping area, the torso would bend and rotate around to reach the desired position if the object is not reachable at the current state. First of all, inverse kinematic solver can be generated by OpenRAVE using Meka robot's kinematics. OpenRAVE is a software for testing, developing, and deploying motion planning algorithms for real-world robots and their applications [56]. Because Meka arm has 7DOF with a redundant joint, a free angle is desirable, and J2 is selected as the free angle for Meka robot. The free angle is a variable and must be changed for the solver to compute a inverse kinematics solution. After that, it is necessary to check the joint limits, iterate the free angle to search the solutions, and choose the closest one to the current joint state from the available solutions. Lastly, the highest level procedure enable the inverse kinematics solutions be calculated with respect to the base frame of the robot by providing necessary robot data such as torso angles [2].

In addition, minimum jerk trajectory [12] is used between the pre-grasp and the grasp poses such that the end-effector moves smoothly and does not hit the object. For every joint, the minimum jerk trajectory from angle θ_i to angle θ_f over T seconds is:

$$\theta(t) = \theta_i + (\theta_f - \theta_i)(10(\frac{t}{T})^3 - 15(\frac{t}{T})^4 + 6(\frac{t}{T})^5) \tag{2.19}$$

The gesture based grasp activation has some specific contributions related to the existing approaches. Motion planning is not required during autonomous grasping. The human operator is supposed to bring the end-effector close enough to the object, so that the remaining trajectory is very short, hence a collision between the manipulator and environment is quite unlikely.

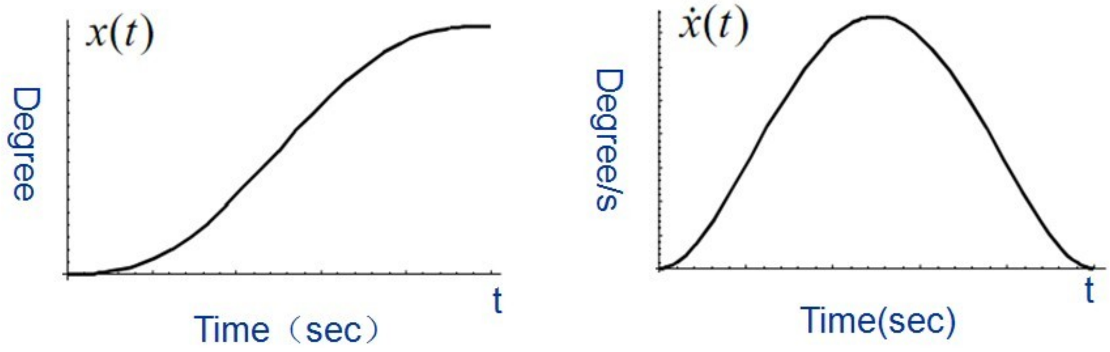


Figure 2.22: Jerk trajectory

2.3 IR-based Grasp Assistance

This section presents that teleoperation is assisted by three infrared sensors which are mounted on the hand. The first capability is that the hand can autonomously move to a suitable pre-grasp position if at least one of the three sensors detects the object. The second functionality of the system is robust. Even though it is a mobile object, the end-effector also could track it. Thirdly, it could avoid collision, for the reason that it will keep a distance to any object. With the more sensors detect the object, the distance would be larger. At last, additional algorithm for orientation can adjust the hand according to the object surface, so that it can avoid the failure caused by the collision of the robot fingers and polygon objects.

This algorithm is based on our previous work of teleoperation. KINECT is employed as the sensor to collect human joint position data, then the robot could mimic human after kinematics. At the same time, the human hand gesture could be recognized as open or closed, which could provide the signal for the robot to open or close hand.

2.3.1 Hardware–infrared Sensors with Robot Hand

The algorithm utilizes data from the infrared sensors to construct a potential field, which is used to control the end-effector of the robot. The design is driven by a series of constraints, including size, sensor response and field of view.

Infrared Proximity Sensors (Short Range - Sharp GP2D120XJ00F) are employed in this experiment. After calibration, the range of every IR sensor is $[4cm, 12cm]$. Optical proximity sensors are efficient, robust, and less sensitive to light, which suit our application of online grasp adjustments from a pre-grasp point. In addition, IR sensors have much lower computational expense due to its simplicity compared with 3 dimensional sensors. Low computational complexity is crucial for real time feedback in our system.

In order to improve the stability of grasping, the object should be wrapped in the curve of the palm and have enough contact area with the palm in the z direction, so that the sensors are mounted above the palm with a triangle. To achieve the effect with the least sensors, three sensors are arranged as Fig. 2.23 for adjusting the hand on x, y and z directions. Sensor 1 and 3 are arranged in this configuration to find, respectively, the y and z direction edges of the object with respect to the sensor

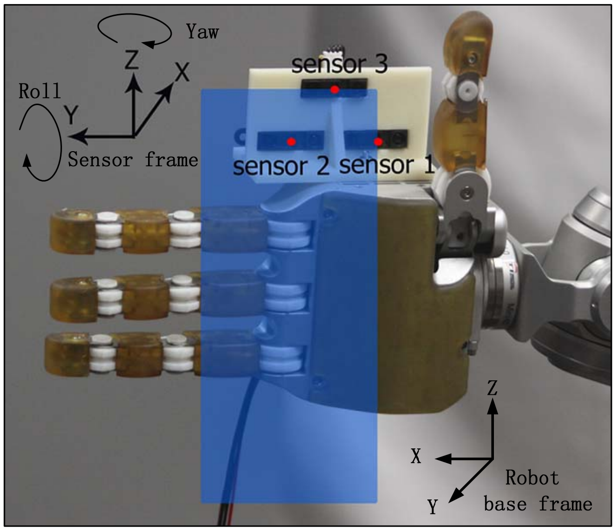


Figure 2.23: Hardware of IR sensors: The red points are the centers of sensor, and two edges of objects are supposed to be on the sensor 1 and sensor 3 centers as the blue rectangular.

frame. If the sensors are too close to each other, sensor 2 may be shift out of the edges which may lead to lose signal. In contrast, if the distances among sensors are too large, the whole size of the structure would be large and it cannot track small objects. Thus, the ideal distance is $4 - 5cm$. Based on the sensor configuration, we assume the objects to be grasped are not too small.

2.3.2 Final Grasping Algorithm

Given a series of observations from the sensors, the goal is to search for the edges of the object.

When the hand is close to an object, the signal from the IR sensors will be used to construct a potential field, U_{IR}. The hand would move to a desired pre-grasp location corresponding to the minimum of potential functions for the x, y, and z directions, which are defined respectively as follows:

$$U_{IRx} = \frac{1}{2}[(r_1 - a_1)^2 + (r_2 - a_2)^2 + (r_3 - a_3)^2] \tag{2.20}$$

$$U_{IRy} = \frac{1}{2}(r_1 - a_1)^2 \tag{2.21}$$

$$U_{IRz} = \frac{1}{2}(r_3 - a_3)^2 \tag{2.22}$$

where,
l_1 and l_2 are the minimum and maximum ranges for the sensor respectively,
$r_i \in [l_1, l_2]$, r_i are the distances detected from sensor i,
a_i are the desired r_i values of pre-grasp position.

If all of the three sensors cannot detect any object, U_{IR} is zero; therefore, the execution of the assisted pre-grasp motion depends on the initial placement. On the contrary, if any one of the sensors detects an object, U_{IR} has values.

y-direction

In the y direction (see (2.23)), if sensor 1 detects the object, the hand moves towards the negative y direction, and vise versa. According to the experiment experience, a_1 should be close but less than l_2. This enables the hand to move in the positive y direction slowly to avoid vibrations caused by changing direction too frequently when sensor 1 detects the edge of the object. Thus, we have

$$\begin{cases} r_1 < a_1, \ \textit{move to negative y direction} \\ r_1 = a_1, \ \textit{stable in y direction} \\ r_1 > a_1. \ \textit{move to positive y direction} \end{cases} \tag{2.23}$$

z-direction

The motion in the z direction is similar with that of the y direction.

$$\begin{cases} r_3 < a_3, \ \textit{move to positive z direction} \\ r_3 = a_3, \ \textit{stable in z direction} \\ r_3 > a_3. \ \textit{move to negative z direction} \end{cases} \tag{2.24}$$

x-direction

In the x direction (see (2.26)), it should meet the following requirements:

If more than one sensor detects the object, it may cause $r_1 + r_2 + r_3$ become small. What is more, it means the hand may move on the y or z direction, which is elaborated through analyzing U_{IRy}. In this case, the hand keeps a distance from the object to avoid hitting the object.

If sensor 1 and sensor 3 cannot detect the object, as well as sensor 2 can, the hand moves to be close to the object and keep a shorter distance to the object compared to the previous case.

The values of a_i should satisfy the condition:

$$a_1 + a_2 + a_3 = d + 2l_2 \tag{2.25}$$

where d is the ideal distance from the sensor to the object for grasping. Therefore, when sensor 1 and sensor 3 are out of the edge of the object($r_1 = r_3 = l_2$), r_2 will become d to keep stable, which means the robot can close hand directly to grasp the object.

$$\begin{cases} r_1 + r_2 + r_3 < a_1 + a_2 + a_3, \ \textit{move to positive } x \textit{ direction} \\ r_1 + r_2 + r_3 = a_1 + a_2 + a_3, \ \textit{stable in } x \textit{ direction} \\ r_1 + r_2 + r_3 > a_1 + a_2 + a_3. \ \textit{move to negative } x \textit{ direction} \end{cases} \tag{2.26}$$

From (2.20), it is clear that all sensors contribute to the x direction and have no difference between each other in the form. Thus, in some particular cases, such as grasping slim objects, the hand will move close to the object no matter which sensor detects the object while the rest of the two sensor readings are l_2. This enables the robot to robustly grasp a wide range of objects.

Next, we design virtual forces that drive the desired hand position. The virtual forces are derived from potential energy as follows:

$$F_x = -\nabla U_{IRx} = -[(r_1 - a_1) + (r_2 - a_2) + (r_3 - a_3)] \tag{2.27}$$

$$F_y = -\nabla U_{IRy} = -(r_1 - a_1) \tag{2.28}$$

$$F_z = -\nabla U_{IRz} = -(r_3 - a_3) \tag{2.29}$$

This virtual force vector is fed to a virtual mass-damper system to derive the desired hand position.

$$F = m\ddot{x} + k\dot{x} \tag{2.30}$$

where $m\ddot{x}$ is the component of inertial force and $k\dot{x}$ is the component of spring force, the following equations can be derived,

$$\ddot{x} = -c_x\dot{x} + F_x \tag{2.31}$$

$$\ddot{y} = -c_y\dot{y} - F_y \tag{2.32}$$

$$\ddot{z} = -c_z\dot{z} + F_z \tag{2.33}$$

where, $c_x > 0$, $c_y > 0$ and $c_z > 0$ are the parameters to tune the damping. ξ_{IR_x}, ξ_{IR_y}, and ξ_{IR_z} are the desired hand position to reduce the potential energy. Because positive force drives the hand to negative direction in y axis, the component of F_y is negative in (2.32), and it is inverse in x and z directions (see 2.31 and 2.33). The

IR algorithm outputs are the positions and velocities of tool frame in x, y and z directions.

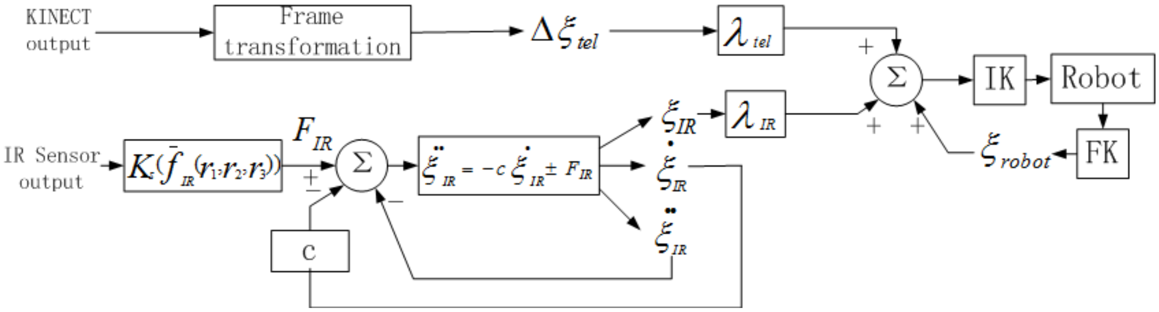


Figure 2.24: Conrol block diagram. FK and IK denote forward & inverse kinematics maps respectively.

If one or more sensors detect an object, the system will be converted from entirely teleoperation to the proposed combined system. The combined system is illustrated in Fig. 2.24.

In the system, in order to match the data from IR algorithm, teleoperation output is the end-effector position with fixed orientation. The positions of the end-effector are obtained by calculating the forward kinematics. The positions at time points $t_1, t_2, t_3, ...$, are denoted by $\Delta\xi_{tel}(t_1), \Delta\xi_{tel}(t_2), \Delta\xi_{tel}(t_3), ...$, where $\Delta\xi_{tel}(t_i)$ is the position changed during teleoperation. The final position is given by:

$$\xi(t_i) = \lambda_{IR}\xi_{IR} + \lambda_{tel}\Delta\xi_{tel}(t_i) + \xi(t_{i-1}) \tag{2.34}$$

where,

$$\lambda_{IR} + \lambda_{tel} = 1, \tag{2.35}$$

and ξ_{IR} the position from IR algorithm, ξ the position combining teleoperation and IR algorithm, $\lambda_{tel} > 0$ and $\lambda_{IR} > 0$ are the parameters for tuning the proportion of teleoperation and IR algorithm in the system.

2.3.3 Adjustment For Orientation

To grasp a polygon robustly, we develop an algorithm based on potential energy for adjusting robot hand orientation. Thus, the robot can tune the orientation of the hand autonomously using this algorithm before performing the adjustment of positions. It is assumed that all of the three sensors detect the same flat surface. To simplify the algorithm, only roll and yaw of the hand are considered. We define:

$$U_{IR\alpha} = \frac{1}{2}[(r_3 - r_2)^2] \tag{2.36}$$

$$U_{IR\gamma} = \frac{1}{2}[(r_1 - r_2)^2] \tag{2.37}$$

where, the subscripts α and γ denote the roll and yaw of the hand respectively. The adjustment strategy for the roll orientation is that if the value of sensor 3 is less than that of sensor 2, the hand should rotate to negative roll orientation to decrease the difference of the values of sensor 3 and sensor 2, and vice versa. This is summarized as follows:

$$\begin{cases} r_3 < r_2, \ \textit{rotate to negative roll orientation} \\ r_3 = r_2, \ \textit{stable in roll orientation} \\ r_3 > r_2. \ \textit{rotate to positive roll orientation} \end{cases} \tag{2.38}$$

The adjustment strategy for the yaw orientation is similar to that roll orientation, but involves sensors 1 and 2 instead. It is summarized as follows:

$$\begin{cases} r_1 < r_2, \ \textit{rotate to positive yaw orientation} \\ r_1 = r_2, \ \textit{stable in yaw orientation} \\ r_1 > r_2. \ \textit{rotate to negative yaw orientation} \end{cases} \tag{2.39}$$

To realize the above-mentioned strategy, the virtual forces are generated:

$$F_\alpha = -\nabla U_{IR\alpha} = -(r_3 - r_2) \tag{2.40}$$

$$F_\gamma = -\nabla U_{IR\gamma} = -(r_1 - r_2) \tag{2.41}$$

The angles of roll and yaw can be computed using:

$$\ddot{\alpha} = -c_\alpha \dot{\alpha} + F_\alpha \tag{2.42}$$

$$\ddot{\gamma} = -c_\gamma \dot{\gamma} - F_\gamma \tag{2.43}$$

where, $c_\alpha > 0$ and $c_\gamma > 0$ and $c_z > 0$ are the parameters to tune the damping.

Chapter 3

Experiment Results and Discussion

In the experiments described here, the performance of gesture based grasp activation and IR-based grasp assistance is evaluated. The experiments are implemented using Robot Operating System (ROS).

3.1 Robot Setting

Before starting experiments, the robot is set up. The parameters of Meka robot arms and hands are set as Table 3.1 and Table 3.2.

The manipulator uses joint angle control with gravity compensation, while the hand employs both joint angle control with gravity compensation and torque control with gravity compensation.

For joint angle control with gravity compensation, the dynamic equation is:

Table 3.1: Parameters of the arms of MEKA robot

Parameters	Arm joints			
	Shoulder		Elbow	
	J0	J1	J2	J3
Mobility range (Robot right arm) [deg]	-47 to 197	-77 to 77	0 to 144	-80 to 125
Force resolution	3 mNm			
Slew Rate [deg/s]	0-30			
Joint controller stiffness	0.75			
Joint controller mode	Position Control			
Angular resolution	0.022 degree			
Payload (each arm)	0 to 2.4 Kg			

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{3.1}$$

where, q is a vector of joint variables, τ the control torque vector, $M(q)$ the inertia matrix, $C(q,\dot{q})\dot{q}$ the Coriolis and centrifugal force vector, and G the gravitational force vector.

The controller is given by:

$$\tau = K(q - q_d) + D(\dot{q} - \dot{q}_d) + I\int(q - q_d)dt + G \tag{3.2}$$

Table 3.2: Parameters of the hands of MEKA robot

Parameters	Hands			
	Thumb	Forefinger	Middle Finger	Ring Finger
	J0,J1	J2	J3	J4
DOF	2	1	1	1
Mobility range [deg]	0 to 300, 0 to 300	0 to 300	1	0 to 300

where K, D and I are diagonal constant matrices, and q_d is the desired position.

For joint torque control with gravity compensation, the controller is:

$$\tau = \tau_d + G \tag{3.3}$$

where τ_d is the desired torque for grasping object.

In order to avoid crushing the hands or objects during grasping but to pick up objects robustly after grasping, the final grasp should employ torque control with gravity for the hands. The strategy for grasping by hands is:

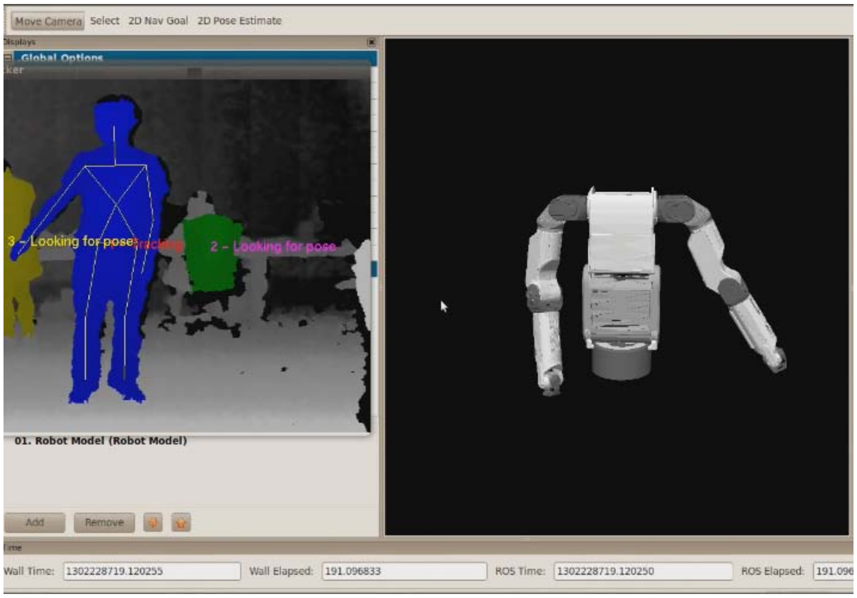
• In the joint angle control with gravity compensation mode, enable the hand to be in a preshaped pose.

- After the hand preshape, change J1-J4 (fingers) to joint torque control with gravity compensation mode and a torque=120mNm to close the fingers, while J0 (thumb) posture is kept constant as in the previous mode.

- To open the hand, switch all joints back to position control, and set J0-J5 to 0 degrees.

3.2 Teleoperation Results

In this experiment, the robot can grasp objects by imitating human using entirely teleoperation. The teleoperation effects are demonstrated through lag analyzation.



Wall Time: 1302228719.120255 Wall Elapsed: 191.096833 ROS Time: 1302228719.120250 ROS Elapsed: 191.096

Figure 3.1: Teleoperation with virtual robot

For the purpose of testing safety, the simulation (see Fig. 3.1) is performed first. The experiment provides a RVIZ (ROS visualization)3D geometrical model of MEKA robot and thus generates arbitrary view point. After verifying that the robot model could imitate the human, we implement the application on the real robot.

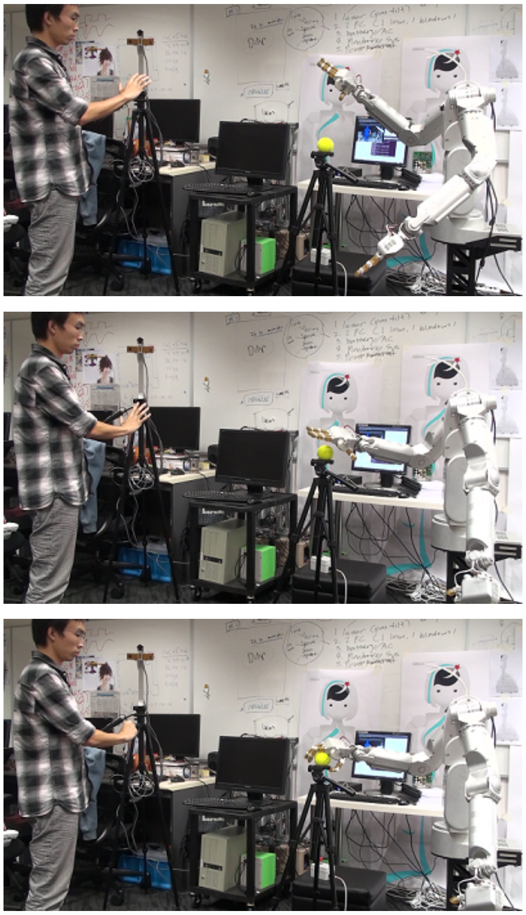
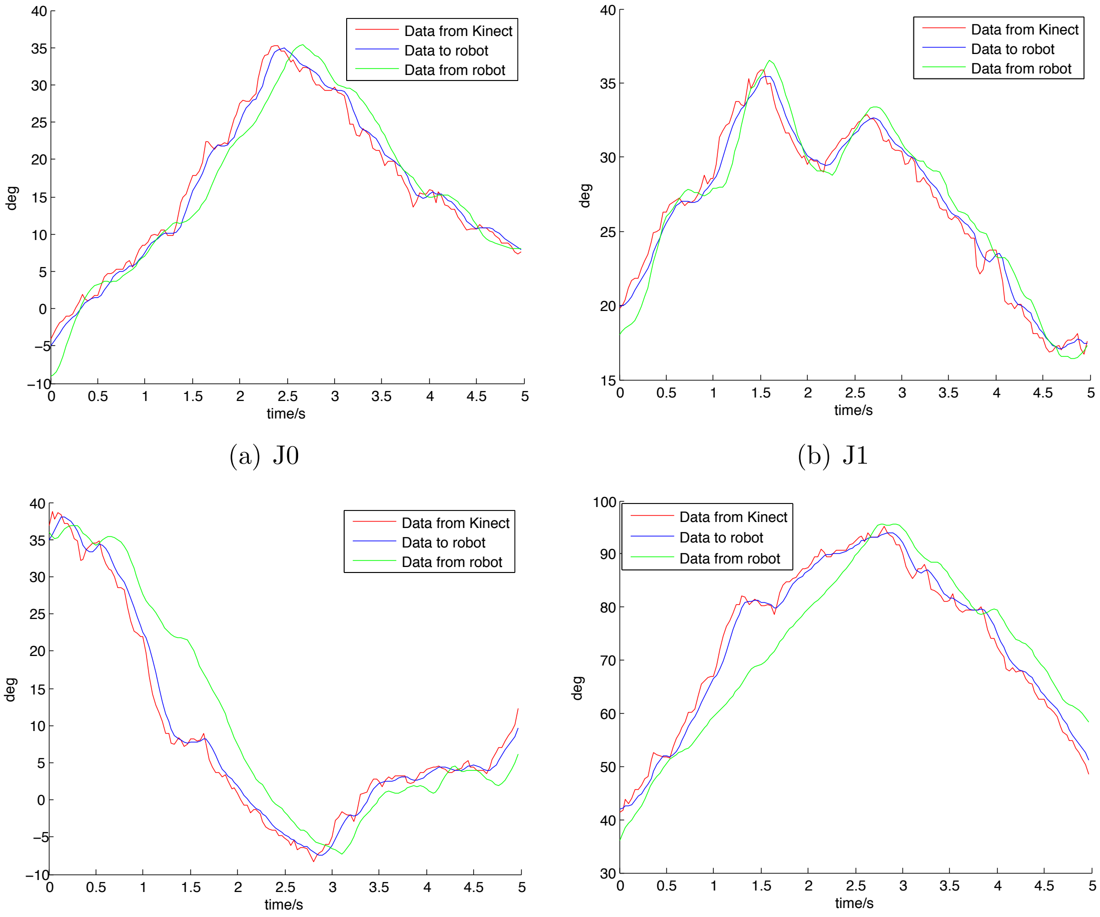


Figure 3.2: Teleoperation with real robot. The first image shows the robot mimics the person to move the arms. The second image shows pre-grasp. The third image shows the robot mimics the person to grasp an object.

The robot can follow the user's motion to achieve some applications such as grasping. The teleoperation results based on the real robot to pick up a ball are illustrated in Fig. 3.2. The robot arm can follow the user's arm motion to reach a pre-grasp position. Then, if the user close the hand, the robot can close the hand to grasp the object directly.



(a) J0

(b) J1

(c) J2

(d) J3

Figure 3.3: Degree of joints during a motion

As can be seen from Fig. 3.3, in this experiment, Kinect collects the data of 3 joints for the right arm during a simple motion. The figure shows the delay of the system. The red line, blue line, and green line represent the data collected from Kinect, joint angles sent to the robot and the current robot joint angles respectively. For convenience, the joint positions collected from Kinect are converted to robot joint angles in this figure. The frequency of the system to collect data and set data to robot is 30 times per second. The delay between data collected from Kinect to data sent to the robot is mainly caused by the mean filter. The mean filter collected 5 sets data to calculate one average set of data, which is approximately $0.133s$ delay. With the mean filter, the data is smoother than the raw data from Kinect. Alternatively, the delay between data sent to the robot to data read from the robot is caused by two factors. One factor is the system frequency which is about $0.033s$. Another factor is caused by the robot system. The delay is influenced by the velocities of the user and the robot. The slower the arm move, the smaller the delay is, while the smaller velocity of the manipulator is, the larger the delay is.

3.3 Experimental Results of Gesture-based Grasp Activation

3.3.1 Autonomous Grasping Results

Similar to teleoperation experiments, autonomous grasping is also simulated on RVIZ first then be performed on the real robot. As seen from Fig. 3.4, the blue box wraps the desired object that will be grasped. The red and green arrows besides the blue box are the x and y axis of the right tool frame (see Fig. 2.20) for grasping a can, which roughly illustrates that the grasp points are correct.

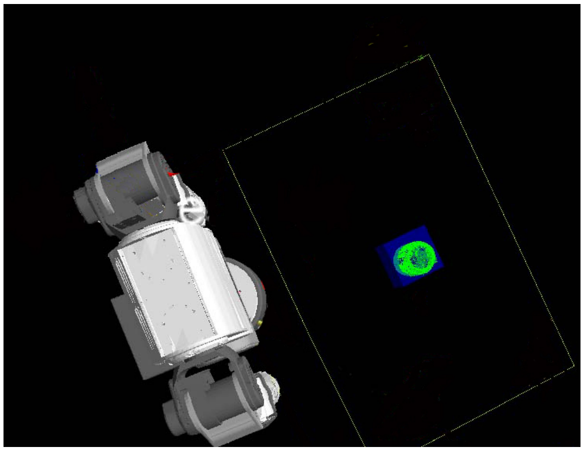


Figure 3.4: Simulation on RVIZ. It shows the detection of the table, the can, and the robot model. The blue cube indicates the selected object.

In addition, Fig. 3.5 shows that the robot that grasps a can. The robot can grasp a known object that matches the database on a flat surface. The object can be randomly put on the plat surface, as long as it is in the range for robot to grasp. If the object is a little far to grasp, the robot would bend the torso to fetch the object, which can extend the graspable area. However, if there are a lot of objects on the table, the person has to choose the object manually and set the command for choosing in advance. In the combined system, the intended object is the one that is closest to the robot wrist frame when the user sends a close command, which will be elaborated in the section of the combined method results.

3.3.2 Combined Method Results

Developed methods of combining teleoperation and autonomy through the signal of the human hand were confirmed in the experiment. Fig. 3.6 shows the experimental

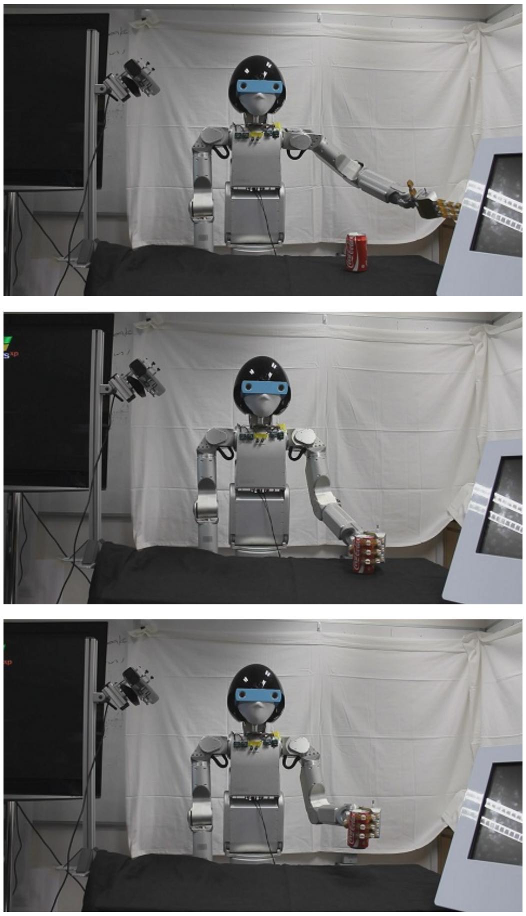


Figure 3.5: Autonomous grasping. The first image shows pre-grasp. The second image shows the robot hand wraps an object. The third image shows the robot lifts the object.

results in grasping a can. The robot can imitate the person to move the arm and then autonomously pick up the can.

As illustrated in Fig. 3.7, the results using full teleoperation and using the gesture based grasp activation are compared. The figures show the position of the left tool frame, from rasing up the arm from the initial position, grasping an object, to lifting the object up. In the fully teleoperated mode, the top picture of Fig. 3.7 illustrates

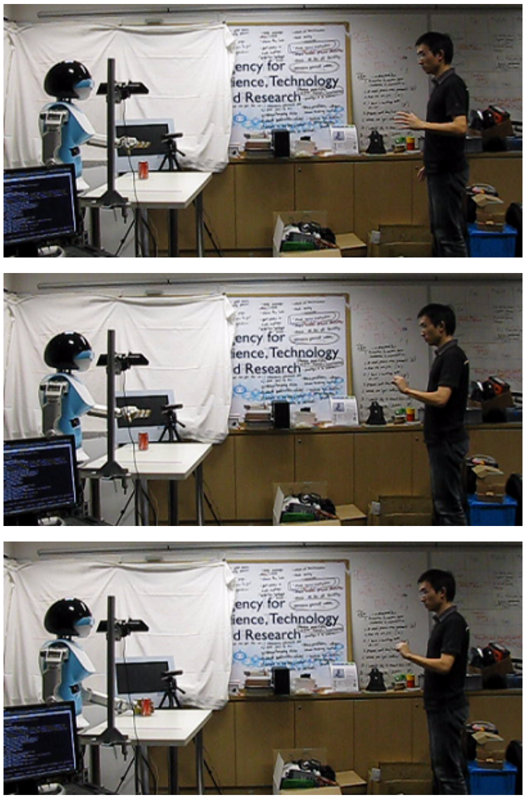
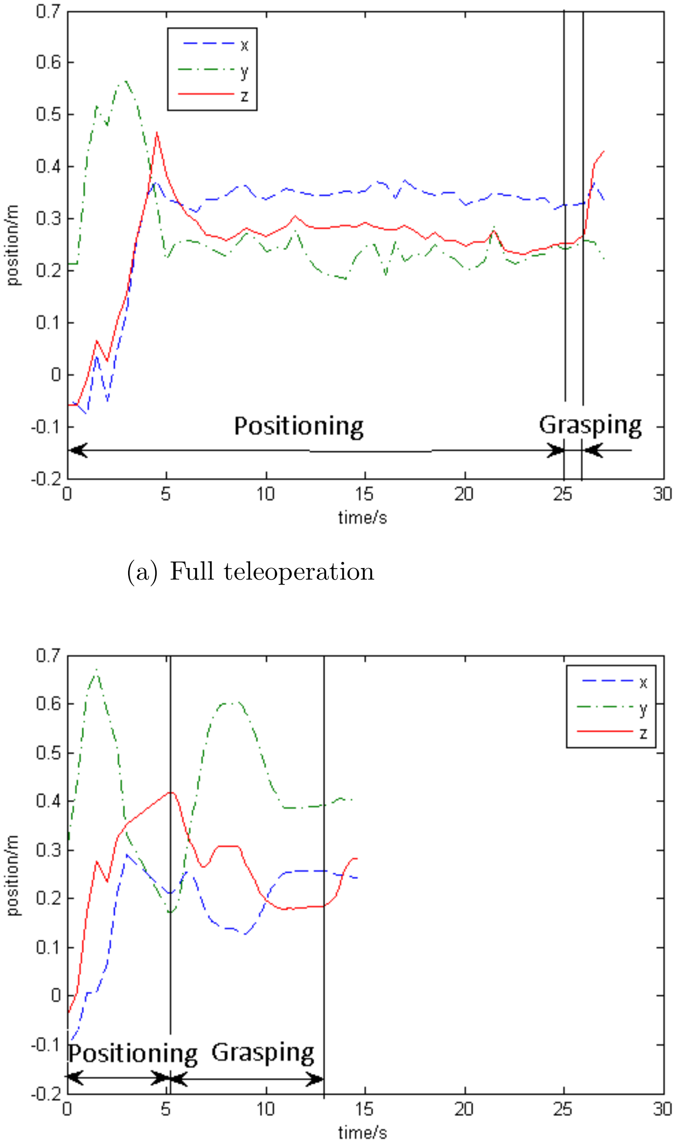


Figure 3.6: Grasping object using the combined method. The first image shows the teleoperation. The second image shows the switching from the teleoperation to autonomous grasping. The third image shows autonomous grasping.

that the end-effector is not stable during grasping, because the person has to adjust the end-effector to reach an appropriate position. It is not easy to reach the position precisely and it depends on the skills of the person. On the contrary, by using the combined method, the trajectories of the motion are much smoother during the whole procedure, including the switching point from teleoperation to autonomous grasping. It turns to autonomous grasping at 5.04s, then picks up the object. The total time is 14.56s, which is almost half of the time compared with using full teleoperation. As illustrated in figure, y has large variation during grasping phase, because the end-effector is first moved to a safe pre-grasp position to avoid possible collision with



(a) Full teleoperation

(b) Gesture-based

Figure 3.7: The position of the tool frame during grasping

tabletop. Obviously, with the aid of autonomous grasping, human need not train the skills to grasp the object. It makes the whole process be much easier operated by the user. Thus, the system is much more effective.

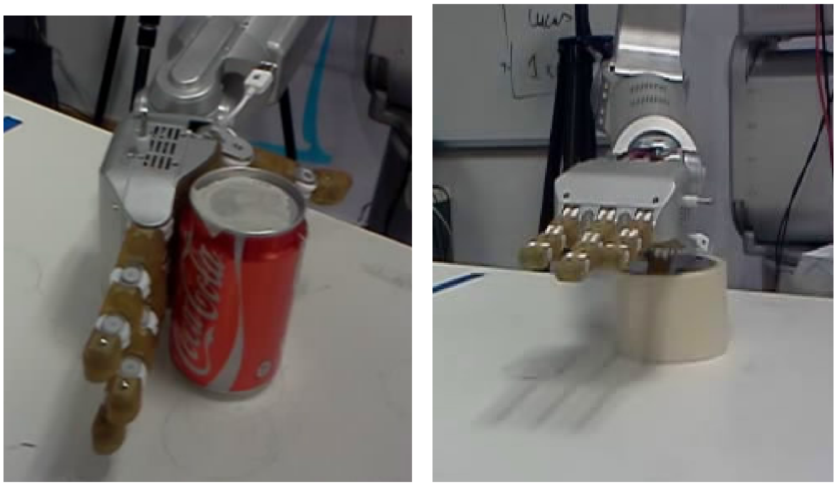


Figure 3.8: Grasp postures. The left image shows grasping a can. The right image shows grasping a tape.

Table 3.3: Grasping success rates

Object Method	can	plastic bottle	tape
Full Teleoperation	4/10	3/10	9/10
Human-aided	10/10	10/10	10/10

We do some experiments to compare the two methods. In the experiments, three objects which have different shapes (can, plastic bottle and tape) are studied. Table 3.3 demonstrates that the success rates of the combined method are higher than that of full teleoperation. The failures of full teleoperation mainly include two types across the various tests. The first type is the grasp poses, which are determined by the limitation of teleoperation. Since it is not easy to collect full data of the person without using additional teleoperation components, only four robot joints can be controlled. Therefore, the robot can only grasp the object from above instead of from the side. In particular, a natural pose for grasping a tape is from above, while a natural pose for grasping a long cylinder is from the side. That is also the reason that the success rates of grasping the can are much lower than the rates of grasping tapes. Another source of failure for full teleoperation is human skills. Even for grasping the tape, the failure cannot be avoided because of the human skills, and the skill level is different from one person to another. Especially, in this experiment, higher skills are required because of the lack of force feedback. In contrast, using the combined method even without training for the person, the robot can use the redundant DOFs of the manipulators and grasp the object accurately by adapting the grasp to different shapes of the object using the structure based grasp activation method.

3.4 Experimental Results of IR-based Grasp Assistance

In this proces, the ability of the IR algorithm is evaluated using graspable area detection, graspable object detection, and tracking object ability test. In addition, some experiments are performed combining teleoperation.

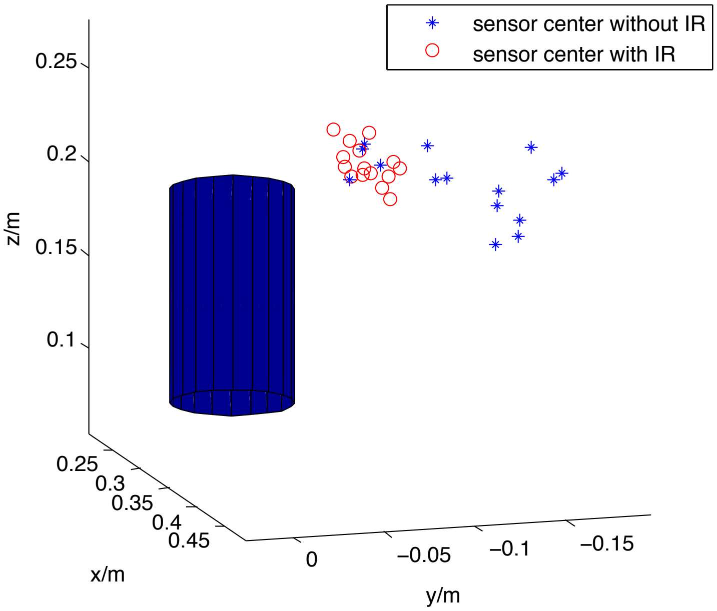


Figure 3.9: Teleoperation with and without the proposed IR Algorithm. The blue cylinder represents a soda can, and the points the pre-grasp positions.

3.4.1 Comparison of Full Teleoperation with Full Assistance

Fig. 3.9 shows 15 samples of the grasp positions of teleoperation, with and without the proposed IR algorithm in the robot base frame. For convenience, the center of sensor 3 represents the grasp positions. To illustrate the advantages of the algorithm with IR sensors, teleoperation data from KINECT, which is used for both with and without IR algorithm, is recorded and played back. The recorded data are randomly selected in the conditions that the IR sensors could detect the object. In this experiment, 5 out of 15 points are successful for fully teleoperated grasping, while 15 points are successful for fully assisted grasping. The variances of position are $(3.0898 \times 10^{-4} m^2, 1.7 \times 10^{-3} m^2, 3.4870 \times 10^{-4} m^2)$ and $(1.8371 \times 10^{-4} m^2, 1.2504 \times$

$10^{-4}m^2, 1.1074 \times 10^{-4}m^2$) for without and with IR algorithm respectively in the form of (variance(x),variance(y),variance(z)). The results show that the novel algorithm could attract the end-effector to be close to an object for a robust grasp after it detects the object.

3.4.2 Ratio of Teleoperation to Assistance

The ratio of teleoperation to autonomy

$$r = \frac{\lambda_{tel}}{\lambda_{IR}} \tag{3.4}$$

is an important factor affecting the ease of use of the system. We conduct an experiment that investigates the effect of this ratio and estimates an optimal value.

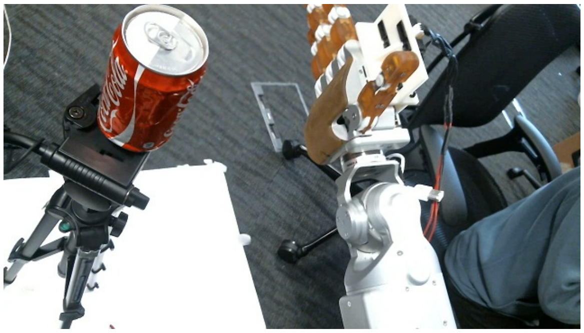


Figure 3.10: 2D vision feedback

Five healthy right-handed subjects, comprising 3 males and 2 females (subject 3 and subject 4), participated in this experiment. All subjects are naive to robot teleoperation, and two subjects are naive to robotics in general. The experimental task is to teleoperate the robot arm to grasp an object, namely a beverage can. The object position is fixed, and the initial positions of the robot and the subject are similar for every trial. Before data collection, each subject is given a session to familiarize himself/herself with the system, by performing 10 full-teleoperation and 10 full-autonomy trials. Also, they are instructed on how to use the IR-based control, in that the robot hand, on detection of the object, will move to a suitable grasp autonomously, but they can continue to move his/her hand to position the robot hand. When the sensors detect the object, audio feedback, in the form of a 'beep', is sounded to the subject. As long as the output of sensor 2 is less than $5cm$, and those of sensors 1 and 3 larger than $10cm$, the robot hand closes automatically. The subject can also raise his/her left hand as a 'manual override' command for closing the robot hand. Visual feedback is obtained from a monitor showing a video streamed from a monocular camera mounted on the robot head. Both the object and the robot hand can be seen clearly from this view (see Fig. 3.10). In the experiment, the value of the ratio r is varied from the set $\Omega_r = \{0, 0.5, 1, 2, \infty\}$, where $r = \infty$ refers to the full-teleoperation mode with $\lambda_{IR} = 0$. Each subject performs a total of 50 trials over 10 sessions, i.e., each session contains 5 trials. The order of r from chosen from Ω_r is randomized, and subjects are not aware of the r values.

IR sensors can improve the grasp ability during teleoperation. Table 3.4 shows the success rates, which we denote by s_r, with different values of r. Without IR sensors, the uncertainty of teleoperation, which is the main cause of low success rate with $r = +\infty$, cannot be ignored. It is difficult for the subject to localize the robot

Table 3.4: Grasping success rates with different ratio of teleoperation and IR algorithm

r	0	0.5	1	2	$+\infty$
subject 1	9/10	10/10	10/10	9/10	7/10
subject 2	8/10	10/10	10/10	7/10	3/10
subject 3	8/10	9/10	9/10	9/10	2/10
subject 4	7/10	9/10	9/10	9/10	3/10
subject 5	8/10	8/10	8/10	9/10	6/10

hand accurately due to the lack of depth information in 2D visual feedback. In addition, different dimensions between human and robot arms pose difficulty for accurate grasping. On the contrary, with IR sensors, the user only needs to roughly drive the robot hand close to the object, then the robot can search for the object autonomously. Although human error cannot be eliminated, the success rate with IR-based grasp assistance can be significantly improved.

We quantify the effectiveness of teleoperation by two measures: effort in grasp placements, and error recovery ability. We define effort in grasping placement as

$$e = s \cdot t \tag{3.5}$$

where,

t is the total duration starting from the time an object is detected to the time the hand arrives at the pre-grasp points,

s is the distance of travel of the end-effector during t,

e is the effort in grasping placement, which reflects the time and end-effector movement during positioning.

The samples for Fig. 3.11 are these of successful grasping of Table 3.4. From the general trend of the 5 subjects, with larger r, the effort is larger. If the influence of IR algorithm is too small, like $r = +\infty$, the person has to manually teleoperate the end-effector to the desired position with great effort, since it is very hard to drive the end-effector to the intended position with teleoperation delay and only 2D visual feedback. In that case, the end-effector will likely travel over a longer distance and duration, i.e. greater effort.

On the other hand, the error recovery ability refers to the ease of which the robot hand trajectory can be altered to the intended one after the sensors detect a wrong object. In our experiment, we quantify it as the distance of the real position of end-effector with the proposed IR algorithm from the ideal position with full teleoperation. High error recovery ability allows ease of manual override when the robot tracks a wrong object. In our experiment, the user teleoperates the robot hand to move close to an object before moving away. The error of end-effector from the final position to the desired position of teleoperation, denoted by e_f, can be seen from Fig. 3.12. It is obvious that with larger influence of teleoperation, the recovery ability is larger.

Fig. 3.13 shows an example of the user driving the robot hand to approach and then move away from an object. For ease of comparison, teleoperation is simulated

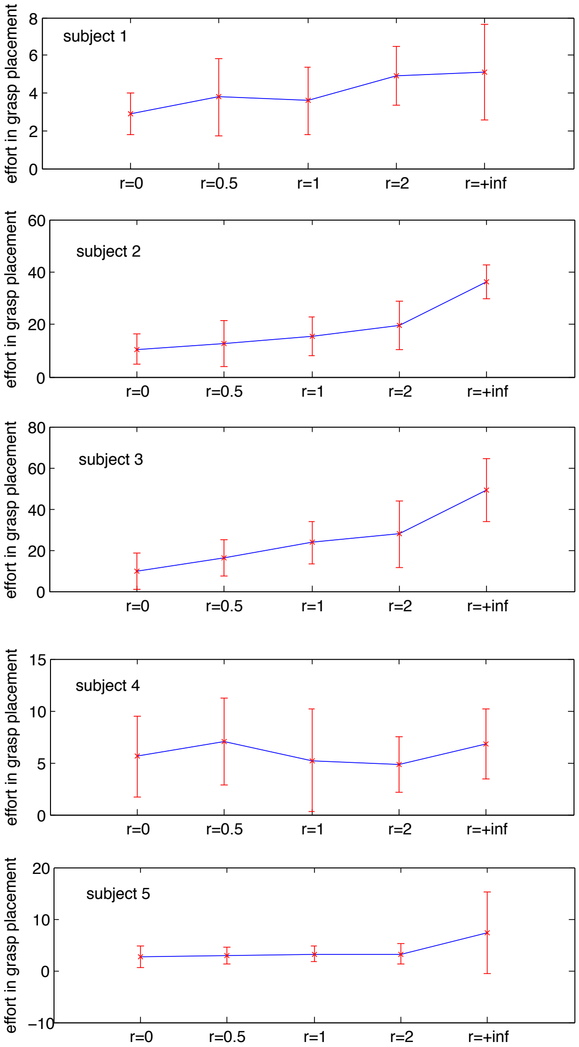


Figure 3.11: Effort in grasp placement for different ratios of teleoperation to assistance. Mean values and standard deviations are shown.

by recorded data as seen from Fig. A.1(b), and the rest(Fig. 3.13(a-e)) uses this teleoperation data. Fig. A.1(c) to Fig. A.3(c) illustrate the different effects using different ratios from $r = 0$ to $r = +\infty$. Generally, the system is entirely teleoperative until the sensors detect an object. With less teleoperation, it is able to track the object more accurately as shown in 3.13(a-c). However, more teleoperation enables

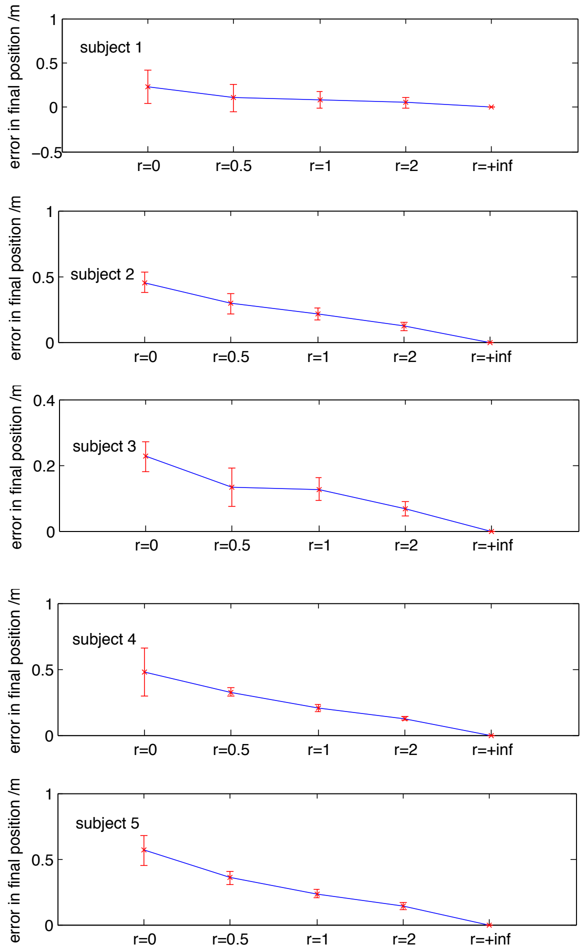
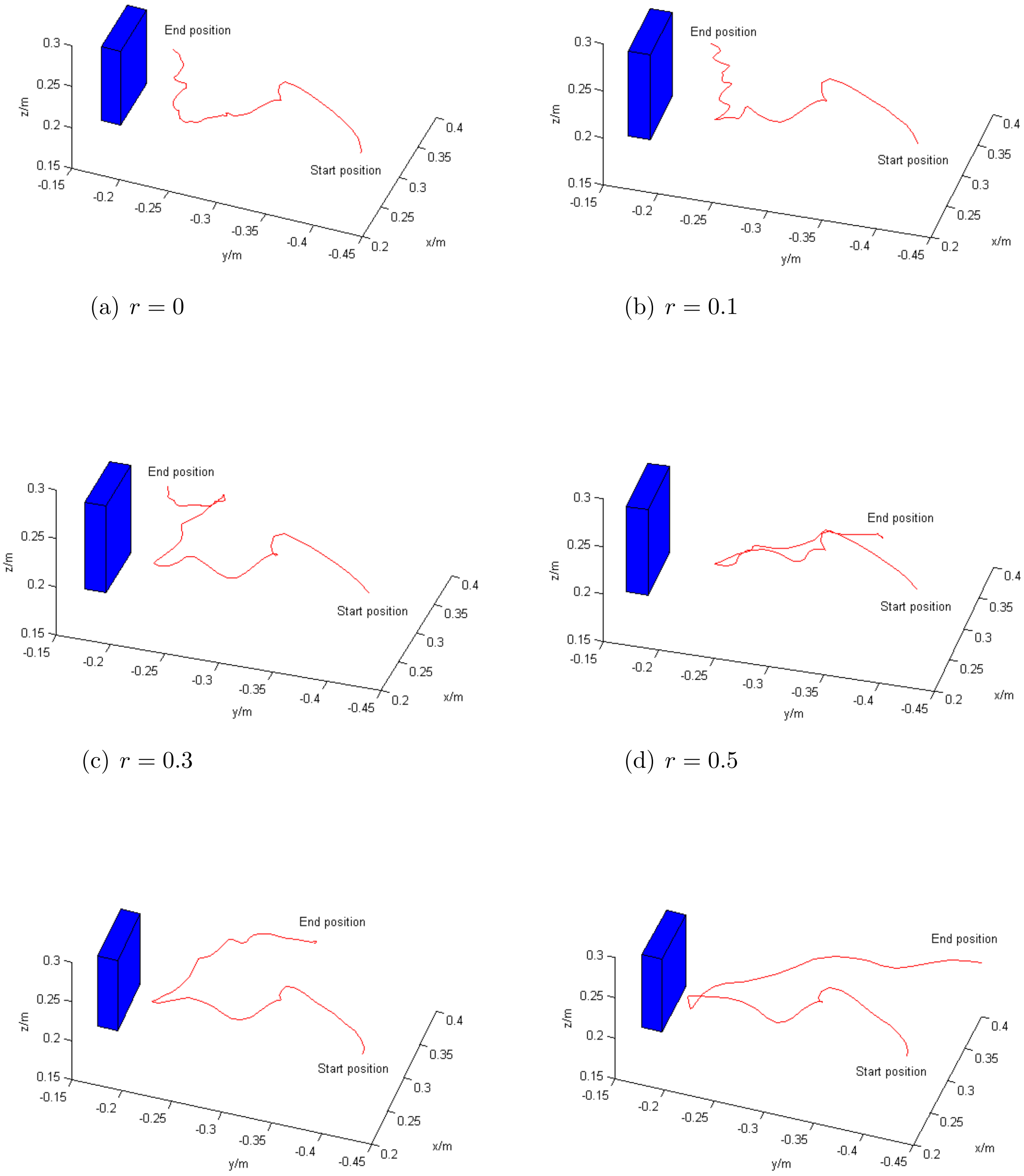


Figure 3.12: Error in final position(e_f) for different ratios of teleop to assistance. Mean values and standard deviations are shown.

users to change the trajectories more easily according to their will, as shown in Fig. 3.13(d-e).

In order to get an optimal parameter of r, all of the above conditions including success rates, effort in grasping placement and error in final position should be considered. The criterion function for r is defined as



(a) $r = 0$

(b) $r = 0.1$

(c) $r = 0.3$

(d) $r = 0.5$

(e) $r = 1$

(f) $r = +\infty$

Figure 3.13: The position of IR sensors with respect to robot base frame during the person moves close to an object then moves away

$$\phi = \begin{cases} f(\gamma), & \gamma = e_f \\ f(\gamma)/s_r. & \gamma = e \end{cases} \tag{3.6}$$

where,

$f(*)$ is a normalization algorithm for an subject,

s_r is success rate in Table. 3.4.

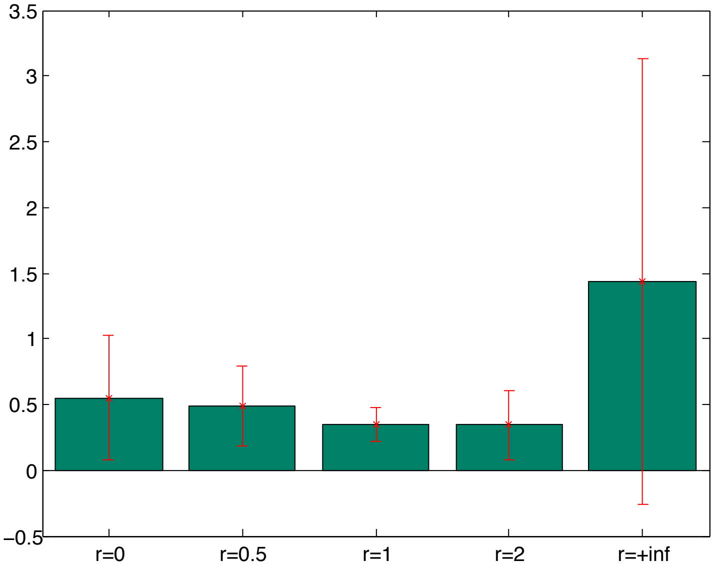
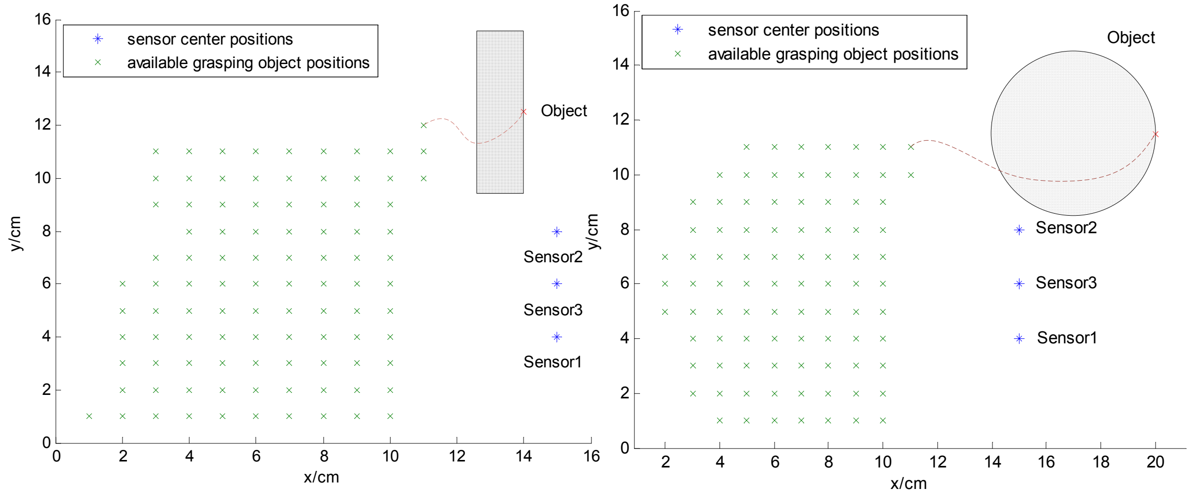


Figure 3.14: The combined efforts of grasp placement and error recovery ability. Mean values and standard deviations are shown.

Because e in (3.5) is based on the condition of successful grasps, the case $\gamma = e$ in (3.6) should take into account the effect of the success rate s_r. Note that ϕ is the effect of the 5 users. Small ϕ suggests that effort in grasp placement or error in final position is less, and the success rate is higher. Hence, from Fig. 3.14, the optimal choice for r is around $r = 1$, which gives the least combined effort and final error with a small standard deviation. It allows the robot hand to move away from the wrong object as commanded by the user, but generate a stable grasp automatically in the absence of user intervention.

3.4.3 Graspable Areas

The graspable area for the IR algorithm is experimentally verified for sample objects, including objects with circular and rectangular cross sections. In the experiment, the sensor initial positions are fixed, and the objects are put on every sample point of a map in different trails. If the sensors could detect the object and the hand could grasp the object successfully, the position of the object would be marked. This experiment is in the $x-y$ plane without z axis, and the sample distances of x and y directions are both $1cm$ in the map. In Fig. 3.15, end cross ”$\times$” represents the center of the surface facing the IR sensors. If the distance from object to sensors is more than around $12cm$, the hand could not grasp the object. At the same time, the object cannot be too close to the hand so as to avoid collision. The range of sensors are $[4cm, 12cm]$, and the distance of the sensor to the hollow of the palm is $3-4.5cm$; therefore, the available range in x direction should be around $8.5-9cm$ for rectangular object. The width of the object (cube) surface facing to the IR sensors is $6cm$, and the distance between sensor 1 and sensor 2 is $4cm$; therefore, the available range in y direction should be around $10cm$ for rectangular object. Hence, with some tolerance of sensor noises and errors, the graspable area is very close to expectation. It is similar for other shape objects such as circular object (see Fig. 3.15(b)).



(a) Graspable area map of a sample object with rectangular cross-section.

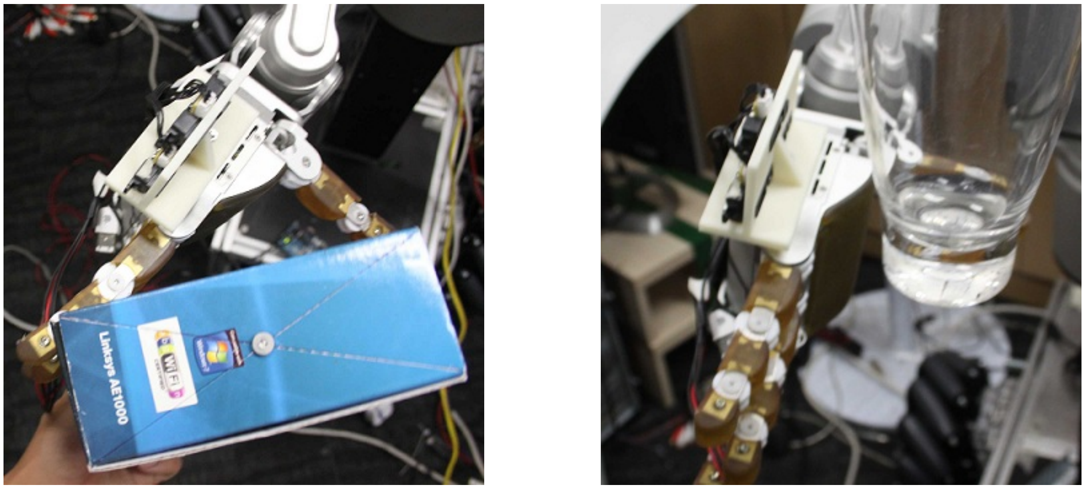
(b) Graspable area map of a sample object with circular cross-section.

Figure 3.15: Graspable area maps verified experimentally. The shaded rectangle and circle are sample objects. End ”×” represents the position of the center of the surface of the object facing the IR sensors.

3.4.4 Graspable Objects

The proposed method enables a robot to grasp a wide range of regular and irregular object as long as the objects are not too small. For ease of analysis, the experiments in this section are based on an entirely autonomous mode of operation, while without any teleoperation. Besides, it employs the algorithm of positioning but not adjustment for orientation.

As can be seen from Fig. A.3, a person hands over an object to the robot in the range of IR sensors but out of range of a direct grasp. Then the robot hand moves to a pre-grasp position and closes the hand to grasp the objects. Irregular objects such as a cutter, tape and toy hand, as well as slim objects such as a marker pen can be stably grasped. If only one sensor detects the object, the hand also will move close to the object, thus ensuring that slim objects can also be grasped. In addition, the sensors are close enough to each other so that even slim objects are detected by at least one sensor.



(a) collision

(b) transparent object

Figure 3.16: Failures of grasping

However, it fails to grasp transparent objects, which is a limitation of IR sensors (see Fig. 3.16(b)).

Table 3.5: Grasping success rates

degree	0	15	30	45	60	75	90	105	120	135	150	165	180
Success rate	5/5	5/5	5/5	5/5	5/5	5/5	5/5	3/5	2/5	0/5	5/5	5/5	5/5

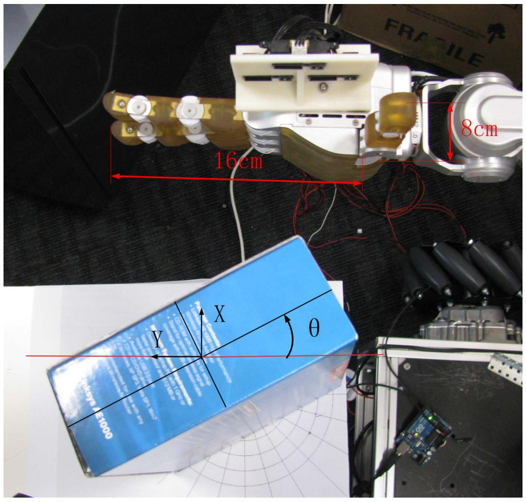


Figure 3.17: Object orientation test

Unsuitable orientation is another reason of failure. To test the influence of orientation for positioning algorithm without orientation adjustment, a box is used as a goal object on the table as shown in Fig. 3.17. The surface of the box towards the sensors has dimensions of $137mm \times 57mm$. The angle θ ranges from 0 to 180 degrees in steps of 15 degrees. The hand orientation is fixed, but the position varies according to the grasp algorithm. With $\theta \in [105, 135]$ degrees (see Table 3.5), the fingers may collide with the object as Fig. 3.16(a).

3.4.5 Adjustment of Orientation

Unsuitable orientation may cause failure for grasping. While the orientation issue can be circumvented by teleoperation, it requires additional effort from the user. In this thesis, we look into the orientation issue especially to achieve more autonomous grasping. The adjustment of orientation is developed to improve the system to become more intuitive.

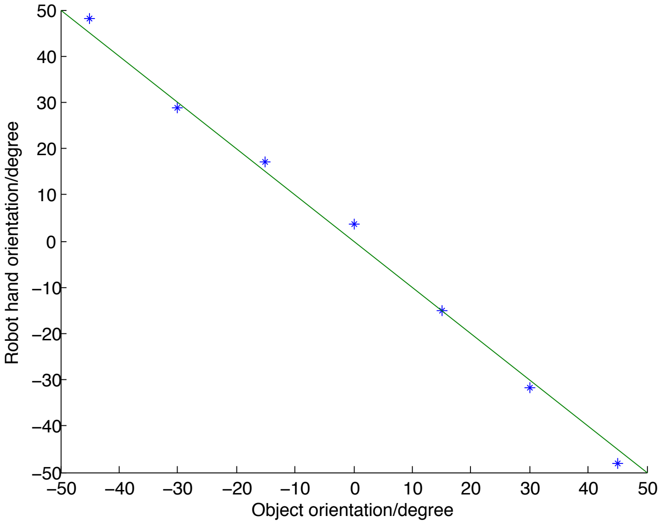
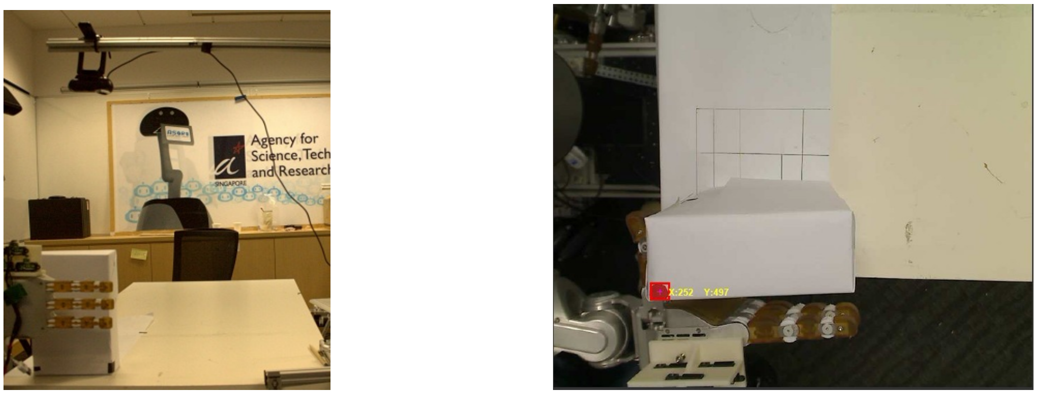


Figure 3.18: Orientation result

In order to deal with the orientation issue, we explore the orientation adjustment algorithm. Fig. 3.18 shows that the robot hand is able to fit the orientation of a cube. The experiment setup can be seen from Fig. 3.17, and the object is rotated about the z-axis by an angle θ that ranges from -45 to 45 degrees. Only the results for the yaw orientation are shown; the results for the roll orientation are similar. The initial yaw orientation of the hand is 0 degree, and the steady state yaw orientation after the algorithm converges is compared with the object orientation. In the 7 samples, the maximum error between the hand orientation and the object orientation is 3.6081 degrees, while the minimum error is 0.1611 degrees. The result demonstrates that the robot hand is able to match the object orientation with the proposed algorithm.

3.4.6 Tracking Mobile Object



(a) Camera setting

(b) View from the camera mounted above the table

Figure 3.19: Object trajectory detection

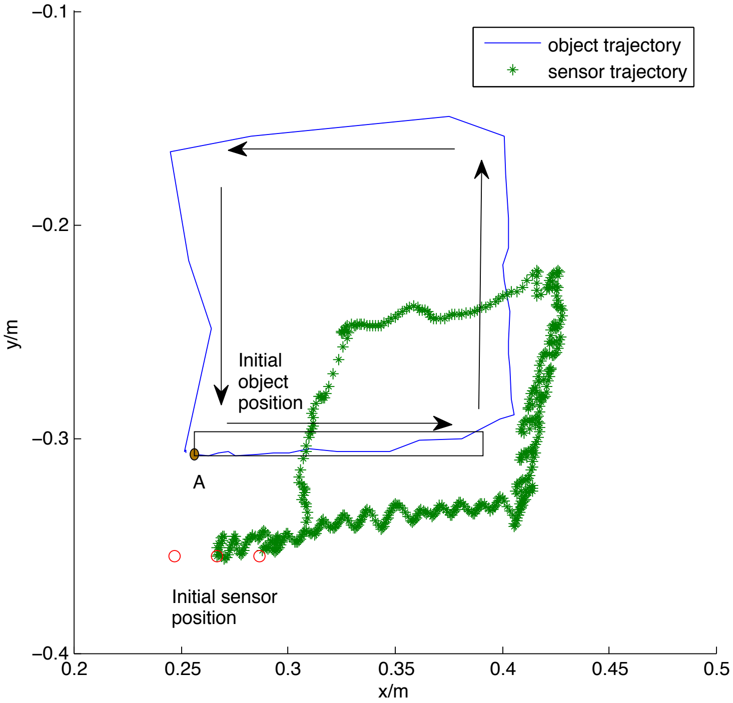


Figure 3.20: Tracking mobile object. The black rectangle is an object. A is a point on the corner of the object. The blue nearly rectangular line is the trajectory of the object. The green stars are the trajectory of sensor 3. The three red circles represent the three sensors. It is with respect to robot base frame.

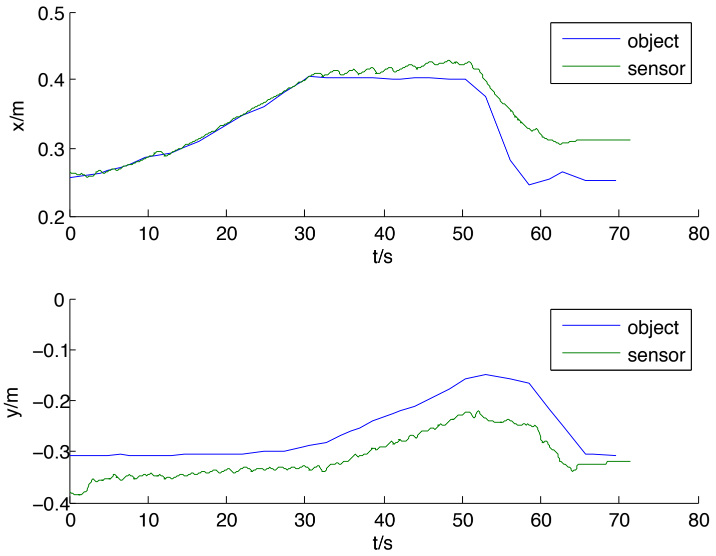


Figure 3.21: Trajectories of sensor 3 and the object during the tracking task

This proposed IR algorithm enables the robot to not only grasp an object robustly but also track it in 3 dimensions.

To test the tracking ability, a box represented by a point labeled 'A' in Fig. 3.20 is translated by hand to trace the edge of $15cm \times 15cm$ rectangle. In this experiment, the box trajectory can be detected using vision method. A camera is mounted above a table, and a red circle is marked on the top of the box. We can get the pixel position of the red circle center, then the object position with respect to the camera frame can be calculated with a scaling(see Fig. 3.19).

This experiment is only performed on the $x-y$ plane, since the result in the z direction is similar as that in the y direction. Fig. 3.20 shows the paths taken by sensor 3 and the object on the $x - y$ plane after the tracking task, and Fig. 3.21 the trajectories in the x and y directions.

In the top graph of Fig. 3.21, when the object is moving in the positive x direction, only sensor 2 can detect the object; therefore, sensor 3 has almost the same trajectory with 'A' point. When the object is moving forwards to the positive y direction, sensor 1 is pointing just off the left edge of the object face parallel to the x direction, resulting in an offset between the object and sensor 3. After that, sensor 1 also detects the object, and the offset becomes larger. In the bottom graph of Fig. 3.21, when the object is moving in the positive y direction, only sensor 2 can detect the object as stated above. However, both sensors 1 and 2 can detect the object most of the time in the opposite direction, and the distance from the sensor to the object is larger. It is the reason why the trajectory length is less than 15cm in the y direction.

Chapter 4

Discussion

The key motivation for this thesis is in establishing an effective system for a human-robot cooperative grasping. Our approach enables the robot to combine teleoperation and autonomous grasping. The experiments show that the user can remotely operate the robot to a suitable pre-grasp position using the Kinect sensor. Next, the robot can either grasp an object after being switched to full autonomous mode using hand gesture commands or perform on-line adjustment of the final grasp to assist the user.

Gesture-based approach combines the advantages of having human initiative, and the accuracy and robustness of the robotic system. Compared to previous approaches, it needs no full feedback from the robot and tedious path planning.

IR-based algorithm neither tries regrasping strategies, premature object contact, nor full object information. Instead it requires a little object knowledge from IR readings.

It allows online grasp adjustments to search for the edge of an object. In the experiments, the algorithm performed very well, and the robot successfully grasped a large number of unknown objects. Although unsuitable orientations of polygon objects may cause failure using only position algorithm, an additional algorithm enables the robot to match the object orientation.

There exists an optimal ratio of teleoperation to assistance in our IR-based algorithm. The optimal choice for r is around $r = 1$, which gives the least combined effort and final error with a small standard deviation. It allows the robot hand to move away from the wrong object as commanded by the user, but generate a stable grasp automatically in the absence of user intervention.

Another highlight of this proposed IR algorithm enables the robot to not only grasp an object robustly but also track it in 3 dimensions.

Chapter 5

Conclusion and Future Work

In this thesis, we have presented effective systems of human-robot cooperative grasping. The systems are developed to operate the robot through combining teleoperation and autonomous grasping. Initially, teleoperation of the robot was implemented with a Microsoft Kinect.

Subsequently, two autonomous grasp approaches were developed for known and unknown objects respectively. For known objects, gesture based grasp activation performed grasping an object without human intervention. IR-based grasp assistive was also developed that enables a robot to perform on-line adjustment of the final grasp during teleoperation. It is a novel method to deal with uncertainty in the environment during teleoperated grasping. For unknown objects, three IR sensors are used for proximal sensing the object distance. A potential function based algorithm provided a corrective signal for the hand to close in on the object using the distance information provided by the sensors. The experiment results demonstrate that a wide range of regular and irregular objects can be grasped, and moving objects can by tracked robustly.

The results proves that the combined approach provides a more precise and effective grasp as compared to a fully teleoperated grasp. A combined teleoperation system provides more flexibility in control and allows quick modification of the process during operation, which is better than full autonomous grasping in certain tasks.

The system can be improved if we embed the sensors in the robot hand instead of mount on the hand. This will be dealt with in the future research. Another limitation of our work is lack of force feedback, which makes the teleoperation less precise. This issue could be handled after improving haptic technology.

Bibliography

[1] I. AnthroTronix. (2011, January) Acceleglove user's guide. [Online]. Available: http://www.acceleglove.com/AcceleGloveUserGuide.pdf

[2] (2011) Meka robotics. [Online]. Available: http://mekabot.com/wiki/doku.php

[3] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, "The intelligent asimo: system overview and integration," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 3, 2002, pp. 2478 – 2483 vol.3.

[4] S. Cousins, "Ros on the pr2 [ros topics]," *Robotics Automation Magazine, IEEE*, vol. 17, no. 3, pp. 23 –25, sept. 2010.

[5] B. Gombert, G. Hirzinger, G. Plank, and M. Schedl, "Modular concepts for a new generation of light weight robots," in *Industrial Electronics, Control and Instrumentation, 1994. IECON '94., 20th International Conference on*, vol. 3, sep 1994, pp. 1507 –1514 vol.3.

[6] R. Ambrose, R. Savely, S. Goza, P. Strawser, M. Diftler, I. Spain, and N. Radford, "Mobile manipulation using nasa's robonaut," in *Robotics and Automation,*

2004. Proceedings. ICRA '04. 2004 IEEE International Conference on, vol. 2, 26-may 1, 2004, pp. 2104 – 2109 Vol.2.

[7] *Springer Handbook of Robotics*, 1st ed. Springer, Jun. 2008. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/354023957X

[8] A. Bejczy, W. Kim, and S. Venema, "The phantom robot: predictive displays for teleoperation with time delay," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, may 1990, pp. 546 –551 vol.1.

[9] T. Tezuka, A. Goto, K. Kashiwa, H. Yoshikawa, and R. Kawano, "A study on space interface for teleoperation system," in *Robot and Human Communication, 1994. RO-MAN '94 Nagoya, Proceedings., 3rd IEEE International Workshop on*, jul 1994, pp. 62 –67.

[10] S. Jacobsen, M. Olivier, F. Smith, D. Knutti, R. Johnson, G. Colvin, and W. Scroggin, "Research robots for applications in artificial intelligence, teleoperation and entertainment," *The International Journal of Robotics Research*, vol. 23, no. 4-5, pp. 319–330, 2004. [Online]. Available: http://ijr.sagepub.com/content/23/4-5/319.abstract

[11] O. Fukuda, T. Tsuji, M. Kaneko, and A. Otsuka, "A human-assisting manipulator teleoperated by emg signals and arm motions," *Robotics and Automation, IEEE Transactions on*, vol. 19, no. 2, pp. 210 – 222, apr 2003.

[12] E. R. Bachmann, R. B. McGhee, X. Yun, and M. J. Zyda, "Inertial and magnetic posture tracking for inserting humans into networked virtual environments," in

In ACM Symposium on Virtual Reality Software and Technology (VRST. ACM, 2001, pp. 9–16.

[13] J. Kofman, X. Wu, T. Luu, and S. Verma, "Teleoperation of a robot manipulator using a vision-based human-robot interface," *Industrial Electronics, IEEE Transactions on*, vol. 52, no. 5, pp. 1206 – 1219, oct. 2005.

[14] ——, "Teleoperation of a robot manipulator using a vision-based human-robot interface," *Industrial Electronics, IEEE Transactions on*, vol. 52, no. 5, pp. 1206 – 1219, oct. 2005.

[15] L. Natale, F. Orabona, G. Metta, and G. Sandini, "Exploring the world through grasping: a developmental approach," in *Computational Intelligence in Robotics and Automation, 2005. CIRA 2005. Proceedings. 2005 IEEE International Symposium on*, june 2005, pp. 559 – 565.

[16] R. Pelossof, A. Miller, P. Allen, and T. Jebara, "An svm learning approach to robotic grasping," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 4, 26-may 1, 2004, pp. 3512 – 3518 Vol.4.

[17] A. Bicchi and V. Kumar, "Robotic grasping and contact: a review," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 1, 2000, pp. 348 –353 vol.1.

[18] A. Miller, S. Knoop, H. Christensen, and P. Allen, "Automatic grasp planning using shape primitives," in *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, vol. 2, sept. 2003, pp. 1824 – 1829 vol.2.

[19] A. Saxena, J. Driemeyer, J. Kearns, C. Osondu, and A. Ng, "Learning to grasp novel objects using vision," in *Experimental Robotics*, ser. Springer Tracts in Advanced Robotics, O. Khatib, V. Kumar, and D. Rus, Eds. Springer Berlin / Heidelberg, vol. 39, pp. 33–42.

[20] 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. [Online]. Available: http://ijr.sagepub.com/content/10/4/314.abstract

[21] B. Wang, L. Jiang, J. LI, and H. Cai, "Grasping unknown objects based on 3d model reconstruction," in *Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on*, july 2005, pp. 461 –466.

[22] G. Bone, A. Lambert, and M. Edwards, "Automated modeling and robotic grasping of unknown three-dimensional objects," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, may 2008, pp. 292 –298.

[23] D. Kragic and M. Bjorkman, "Strategies for object manipulation using foveal and peripheral vision," in *Computer Vision Systems, 2006 ICVS '06. IEEE International Conference on*, jan. 2006, p. 50.

[24] B. Calli, M. Wisse, and P. Jonker, "Grasping of unknown objects via curvature maximization using active vision," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, sept. 2011, pp. 995 –1001.

[25] M. Popović, D. Kraft, L. Bodenhagen, E. Başeski, N. Pugeault, D. Kragic, T. Asfour, and N. Krüger, "A strategy for grasping unknown objects based on

co-planarity and colour information," *Robot. Auton. Syst.*, vol. 58, pp. 551–565, May 2010. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2010.01.003

[26] S.-H. Hong, J. W. Jeon, and J. S. Yoon, "A robot graphic simulator for teleoperation," in *Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on*, vol. 1, 2001, pp. 195 –200 vol.1.

[27] J. Liu, J. D. Simeral, S. D. Stavisky, D. Bacher, J. Vogel, S. Haddadin, P. v. d. Smagt, L. R. Hochberg, and J. P. Donoghue, "Control of a robotic arm using intracortical motor signal by an individual with tetraplegia in the BrainGate2 trial," in *40th Annual Meeting in Neuroscience (SFN2010)*, 2010.

[28] T. Fong and I. Nourbakhsh, "Interaction challenges in human-robot space exploration," *Interactions*, vol. 12, no. 1, pp. 42–45, March 2005.

[29] S. Green, M. Billinghurst, X. Chen, and J. Chase, "Human-robot collaboration: a literature review and augmented reality approach in design," *International Journal of Advanced Robotic Systems*, vol. 5, no. 1, pp. 1 – 18, 2008/03/.

[30] L. Tsoukalas and D. Bargiotas, "Modeling instructible robots for waste disposal applications," in *Intelligence and Systems, 1996., IEEE International Joint Symposia on*, nov 1996, pp. 202 –207.

[31] Y. Inagaki, H. Sugie, H. Aisu, S. Ono, and T. Unemi, "Behavior-based intention inference for intelligent robots cooperating with human," in *Fuzzy Systems, 1995. International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium.,*

Proceedings of 1995 IEEE International Conference on, vol. 3, mar 1995, pp. 1695 –1700 vol.3.

[32] Y. Horiguchi, T. Sawaragi, and G. Akashi, “Naturalistic human-robot collaboration based upon mixed-initiative interactions in teleoperating environment,” in *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, vol. 2, 2000, pp. 876 –881 vol.2.

[33] A. Okamura, “Methods for haptic feedback in teleoperated robot-assisted surgery,” *The Industrial robot*, vol. 31, no. 6, pp. 499–508, dec. 2004. [Online]. Available: http://ukpmc.ac.uk/abstract/MED/16429611

[34] T. Fong, C. Thorpe, and C. Baur, “Multi-robot remote driving with collaborative control,” *Industrial Electronics, IEEE Transactions on*, vol. 50, no. 4, pp. 699 – 704, aug. 2003.

[35] G. Hirzinger, J. Heindl, K. Landzettel, and B. Brunner, “Multisensory shared autonomy - a key issue in the space robot technology experiment rotex,” in *Intelligent Robots and Systems, 1992., Proceedings of the 1992 lEEE/RSJ International Conference on*, vol. 1, jul 1992, pp. 221 –230.

[36] J. Glassmire, M. O’Malley, W. Bluethmann, and R. Ambrose, “Cooperative manipulation between humans and teleoperated agents,” in *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS ’04. Proceedings. 12th International Symposium on*, march 2004, pp. 114 – 120.

[37] J. Rosen, B. Hannaford, M. MacFarlane, and M. Sinanan, “Force controlled and teleoperated endoscopic grasper for minimally invasive surgery-experimental performance evaluation,” *Biomedical Engineering, IEEE Transactions on*, vol. 46, no. 10, pp. 1212 –1221, oct. 1999.

[38] Y. Cheung, *Adaptive Semi-autonomous Teleoperation of a Multi-agent Robotic System*. Stevens Institute of Technology, 2009. [Online]. Available: http://books.google.com.sg/books?id=4uDLjwEACAAJ

[39] D. Lee, “Semi-autonomous teleoperation of multiple wheeled mobile robots over the internet,” *ASME Conference Proceedings*, vol. 2008, no. 43352, pp. 147–154, 2008. [Online]. Available: http://link.aip.org/link/abstract/ASMECP/v2008/i43352/p147/s1

[40] C. Guo, T.-J. Tarn, N. Xi, and A. Bejczy, “Fusion of human and machine intelligence for telerobotic systems,” in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 3, may 1995, pp. 3110 –3115 vol.3.

[41] C. C. Kemp, C. D. Anderson, A. J. Trevor, and Z. Xu, “A point-and-click interface for the real world: laser designation of objects for mobile manipulation,” in *In Proceedings of the 3rd ACM/IEEE international*, 2008, pp. 241–248.

[42] A. Jain and C. C. Kemp, “El-e: an assistive mobile manipulator that autonomously fetches objects from flat surfaces,” *Autonomous Robots*, vol. 28, no. 1, pp. 45–64, 2009. [Online]. Available: http://www.springerlink.com/index/10.1007/s10514-009-9148-5

[43] A. Petrovskaya, O. Khatib, S. Thrun, and A. Ng, "Bayesian estimation for autonomous object manipulation based on tactile sensors," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, may 2006, pp. 707 –714.

[44] K. Hsiao, S. Chitta, M. Ciocarlie, and E. Jones, "Contact-reactive grasping of objects with partial shape information," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, oct. 2010, pp. 1228 –1235.

[45] A. Leeper, K. Hsiao, E. Chu, and J. K. S. Jr., "Using near-field stereo vision for robotic grasping in cluttered environments," in *ISER*, 2010.

[46] K. Hsiao, P. Nangeroni, M. Huber, A. Saxena, and A. Y. Ng, "Reactive grasping using optical proximity sensors," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, may 2009, pp. 2098 –2105.

[47] H. Kim, J. Biggs, W. Schloerb, M. Carmena, M. Lebedev, M. Nicolelis, and M. Srinivasan, "Continuous shared control for stabilizing reaching and grasping with brain-machine interfaces," *Biomedical Engineering, IEEE Transactions on*, vol. 53, no. 6, pp. 1164 –1173, june 2006.

[48] M. Teichmann and B. Mishra, "Reactive robotics i: Reactive grasping with a modified gripper and multifingered hands," *I. J. Robotic Res.*, pp. 697–708, 2000.

[49] OpenNI-Organization. (2011) Openni user guide. [Online]. Available: http://www.openni.org/images/stories/pdf/OpenNI_UserGuide.pdf

[50] G. Gallagher. (2011) Hand detection. [Online]. Available: http://www.ros.org/wiki/mit-ros-pkg/KinectDemos/HandDetection

[51] J. L. Hernandez-Rebollar, N. Kyriakopoulos, and R. W. Lindeman, "The acceleglove: a whole-hand input device for virtual reality," in *ACM SIGGRAPH 2002 conference abstracts and applications*, ser. SIGGRAPH '02. New York, NY, USA: ACM, 2002, pp. 259–259. [Online]. Available: http://doi.acm.org/10.1145/1242073.1242272

[52] P. Brook, M. Ciocarlie, and K. Hsiao, "Collaborative grasp planning with multiple object representations," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, may 2011, pp. 2851 –2858.

[53] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692

[54] A. W and Fitzgibbon, "Robust registration of 2d and 3d point sets," *Image and Vision Computing*, vol. 21, no. 13-14, pp. 1145 – 1153, 2003, british Machine Vision Computing 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0262885603001835

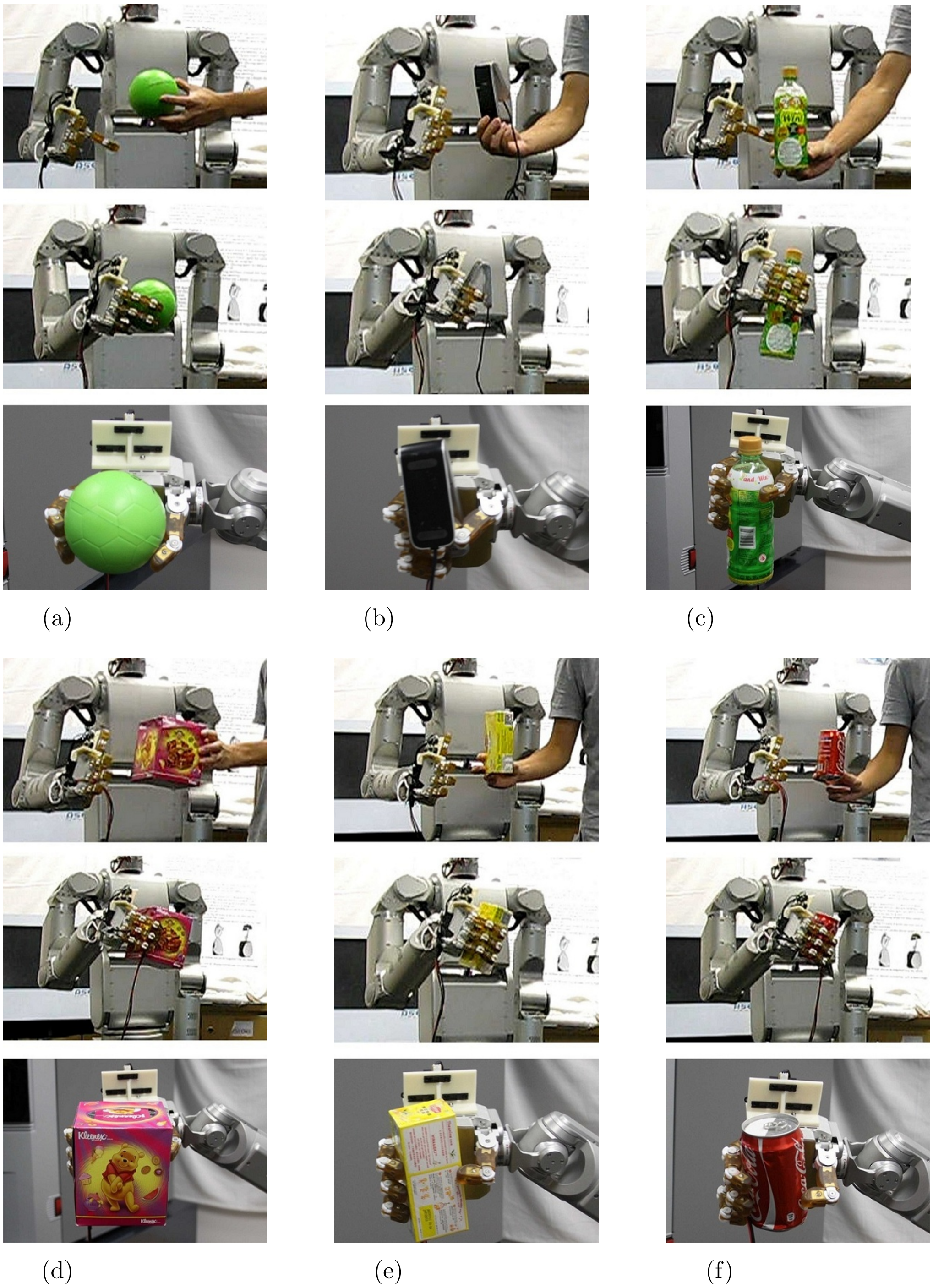
[55] A. Miller and P. Allen, "Graspit! a versatile simulator for robotic grasping," *Robotics Automation Magazine, IEEE*, vol. 11, no. 4, pp. 110 – 122, dec. 2004.

[56] R. Diankov, "Automated construction of robotic manipulation programs," Ph.D. dissertation, Carnegie Mellon University, Robotics Institute, August 2010. [Online]. Available: http://www.programmingvision.com/rosen_diankov_thesis.pdf

Appendix A

Appendix

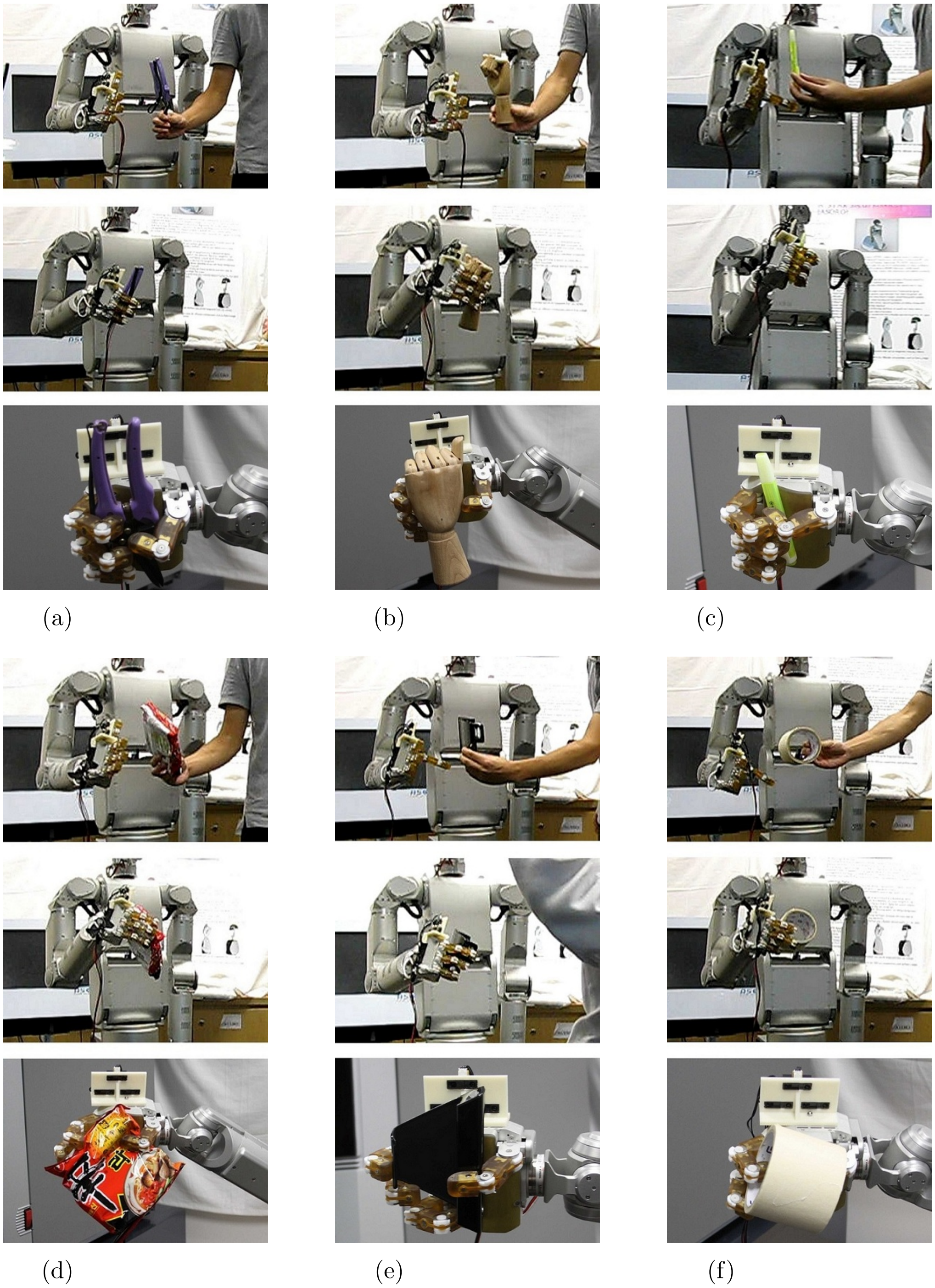
A.1 Appendix 1



(a) (b) (c)

(d) (e) (f)

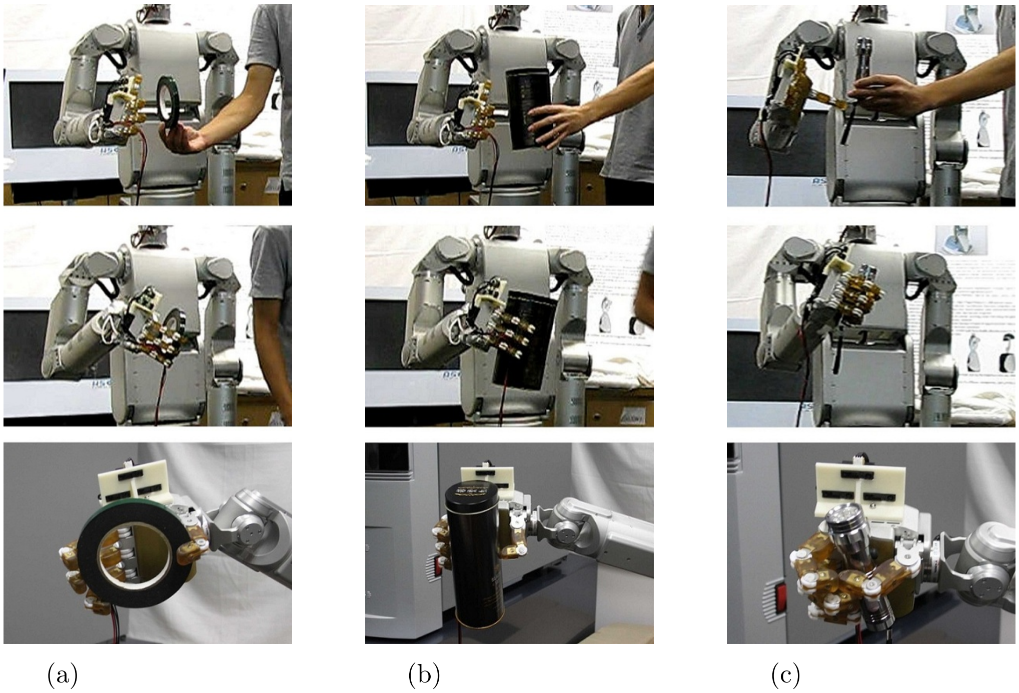
Figure A.1: Grasp experiments: for every object, the top picture shows the initial position of the sensor detecting an object. The middle and the bottom pictures show the final grasp from front view and side view respectively.



(a) (b) (c)

(d) (e) (f)

Figure A.2: Grasp experiments: for every object, the top picture shows the initial position of the sensor detecting an object. The middle and the bottom pictures show the final grasp from front view and side view respectively.



(a) (b) (c)

Figure A.3: Grasp experiments: for every object, the top picture shows the initial position of the sensor detecting an object. The middle and the bottom pictures show the final grasp from front view and side view respectively.