Pretouch Sensing for Robotic Grasping

Liang-Ting Jiang

A dissertation
submitted in partial fulfillment of
the degree of Mechanical Engineering

Doctor of Philosophy

University of Washington

2014

Supervisory Committee:

Joshua R. Smith, Chair

Martin C. Berg

Peter H. Dahl

Blake Hannaford

Joan E. Sanders

Program Authorized to Offer Degree:
Mechanical Engineering

University of Washington

Abstract

Pretouch Sensing for Robotic Grasping

Liang-Ting Jiang

Chair of the Supervisory Committee:
Associate Professor Joshua R. Smith
Computer Science & Engineering and Electrical Engineering

Robotic grasping of unknown objects is recognized to be a challenging problem; this is due to the uncertainty of object shape, caused by the imperfect perception capability of the robot. Vision and depth sensors are commonly used to sense objects before grasping. These sensors suffer from shortcomings such as occlusion, inaccurate sensor readings, and failures due to reflection and transparency. Grasping solely relying on an incomplete object shape can fail regardless of the grasp planning. On the other hand, tactile exploration is widely used to acquire local geometric information of the object and does not subject to occlusion. However, since touch sensing relies on physical contact between the manipulator and the object, it tends to unintentionally displace objects, particularly light objects.

This dissertation considers "pretouch", a sensing modality that is intermediate in range between long-range depth and tactile sensing. This is potentially beneficial for robotic grasping as it provides reliable geometric information in the last centimeter before contact.

In this dissertation, a novel pretouch technique, "seashell effect pretouch", is first presented. It is effective for a set of materials that other pretouch techniques fail to sense. This pretouch modality is inspired by the phenomenon of "hearing the sea" when a seashell is held to the ear; in particular, the observation that the "sound of the sea" changes as the distance from the seashell to the ear varies. It is because environmental noise is amplified the most (attenuated the least) at the cavity's resonant frequency, which changes as the cavity approaches an object. In order to turn the familiar seashell effect into a pretouch sensor,

I study the underlying acoustic principle, i.e., the acoustic radiation impedence changes caused by the object being close to the opening of the cavity. The sensor design, including the acoustic properties, hardware/software design, and signal processing, are discussed in detail. The resulting implementation is fully integrated into the finger of a Willow Garage PR2 robot. The sensor detects resonance frequency shifts in the spectrum of ambient sound, which occur when the finger approaches an object. The performance of the proposed sensor is characterized and evaluated, in terms of the sensing range, accuracy, and its material selectivity. This results in the ability to reliably detect the presence of the object within 5 mm. In addition, a new infrared optical pretouch sensor can be developed, with minimal modification on the proposed sensor system design.

The first explored application is detecting extremely compliant objects during grasp execution. In a pre-grasp execution experiment, the ability to detect compliant objects of the seashell effect pretouch sensor is compared with that of a pressure sensor. The results suggest advantages of seashell pretouch over tactile sensing.

The second application is pretouch-assisted grasp planning. When the pretouch sensor senses the object during a series of probing motions, it provides points collected by recording the position of end effector on the robot; these additional points augment the point cloud from the depth sensor. This method compensates for object shape, that is otherwise incomplete due to depth sensor failure or occlusion. Furthermore, a unified probabilistic framework is proposed to (1) identify shape uncertainty for the target object; (2) automatically explore the uncertain areas to reduce the uncertainty, resulting in a grasp with higher confidence. In the beginning, the robot is provided with only the incomplete object shape data acquired from a Kinect depth sensor—it does not have a model of the object. Next, combining the Kinect point cloud with prior probability distributions for occlusion and transparency, it makes inferences about unobserved portions of the object. Operating on the inferred shape of the object, an iterative grasp replanning algorithm decides whether further exploration is required, and where to explore in the scene. The information gathered by the exploration action is added directly to the environment representation in real-time

and hence considered in the next grasp planning iteration. Experimental results showed that, the robot is able to grasp partially transparent objects with a high success rate of 96%.

Finally, I propose to augment streaming point cloud data with the seashell effect pretouch information. This is inspired by the use case of haptic rendering in a telerobotic grasping scenario. The non-contact seashell-effect pretouch sensor fixed to the robot end effector is used to sense physical geometries within the vicinity of the sensor. Thus, the point cloud representation of an unknown environment, which may be sparse or poorly visible, is enhanced through telerobotic exploration/sensing in real-time. Furthurmore, real-time haptic rendering algorithms are applied on the augmented point clouds to create haptic virtual fixtures, and also provide haptic force feedback to the operator. This method provides the teleoperator with critical geometrical information about the grasp target, while preventing the robot end effector from collision. The augmented virual environment after the pretouch exploration represents more complete object shapes, which helps the operator align the gripper on the slave robot with the target object for grasping.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I would like to express sincere gratitude to my advisor, Dr. Joshua Smith, for giving me the opportunity and flexibility to explore interesting research topics. He has provided invaluable insight for my research. I could not have dreamed of a better advisor with such an inspiring style of supervision.

I am also thankful to Dr. Blake Hannaford for his mentorship in robotic kinematics and manipulataion; Dr. Peter Dahl for his advising about acoustics theory and design; Dr. Martin Berg and Dr. Joan Sanders for serving in my committee.

Finally and most importantly, I want to thank my dearest wife Li-Yen, for her encouragement, tolerance, patience, and love throughout my doctoral study. My thanks to my son, Neil, for all the happiness he brought, that helped me overcome the hardships I encountered in research. I am also grateful to my parents, Che-Ming and Yih-Chun for their constant support, which empowered me with courage for consistently pursuing my goal.

Chapter 1

INTRODUCTION

1.1 Motivation

Robotic grasping of unknown daily objects is an important task for personal/service robots. Recognizing objects can facilitate the grasping task, but creating complete object models for all possible objects in the world is intractable. It is long desired that robots have the capability to grasp arbitrary objects based on its shape. This is a challenging task, mainly due to the object shape uncertainty caused by the robot's imperfect perception. Vision/depth sensors are commonly used for sensing before grasping. Nevertheless, when it comes to execute a grasp, vision has its shortcomings. In particular, the robot's hand and arm often occlude a head-mounted camera's view of the object to be manipulated. A head-mounted camera is not in the same coordinate frame as the hand; thus, even if the camera has an unrestricted view of the object, uncertainties associated with actuation can lead to manipulation errors. Moreover, sensors based on optical modality are also known to fail to sense objects when they are transparent, or under poor lighting conditions. Lysenkov et al. [45] focused on pose estimation and object recognition of only modeled transparent objects from a single RGB-D image captured by a Kinect sensor. However, detecting transparent or partially transparent unknown objects in general remains an unsolved problem due to the strict training requirement. The grasping solely relying on an incomplete object shape can fail regardless of the grasp planning. On the other hand, tactile exploration is widely used to acquire local geometric information of the object [56, 59] and does not subject to occlusion. However, since touch sensing relies on physical contact between the manipulator and the object, it tends to unintentionally displace objects, particularly light objects.

"Pretouch" sensing has been proposed to address the above perception problems. Pretouch refers to sensing modalities that are intermediate in range between long-range depth and tactile sensing. The aim of pretouch sensing is to combine the desirable features of

touch and vision sensing: like touch, it provides information about the relative geometry of the hand and the object, in the hand's coordinate frame; like vision, it is non-contact and does not disturb the object's position. This is potentially beneficial for robotic grasping as it provides reliable geometric information in the last centimeter before contact.

Different pretouch sensing modalities has been proposed previously such as optical [29, 48] and E-field pretouch [49, 75, 81]. However, the optical pretouch has the same drawback as common depth sensors for its lack of capability to sense transparent and reflective objects. On the other hand, E-field pretouch only works for objects with conductive materials or materials with high dielectric constant. A new pretouch sensing modality is desired to complement the types of the materials the robot can sense. Furthermore, the exploration for uncertain areas using the pretouch sensors is not automatic in prior work. A better and more sophisticated method is required to let the robot decide when further exploration is required, and where to explore in the scene with pretouch.

1.2 Organization of the Thesis

Motivated by the facts described in section 1.1, the goal of this dissertation is to improve the robustness of robotic grasping for arbitrary objects based on the shape when their models are not available in advance. Specifically, I propose to improve the perception capability of the robot by utilizing pretouch sensing modalities. Two aspects of the researh are performed:

- Develop a novel pretouch sensing modality which has desirable characteristics to extend the set of materials that the robot can sense.

- Develop new methods of using pretouch sensor information for improving the grasping robustness. Different algorithms and frameworks are proposed, and the efficacy are shown by experiments.

The chapers in this dissertation are organized as follows:

In chapter 3, a novel pretouch technique, "seashell effect pretouch", is presented. This is effective for a set of materials that previous pretouch techniques fail to sense. As the name suggets, this pretouch is inspired by the phenomenon of "hearing the sea" when a

seashell is held to the ear; in particular, I made the observation that the "sound of the sea" changes as the distance from the seashell to the ear varies. In order to turn the familiar seashell effect into a sensor for robotic manipulation, I built a sensor into the robot finger with a acoustic cavity and a microphone. The acoustic principles, sensor design, and the quantitative characterizations of the sensors are studied in detail. Moreover, the seashell effect pretouch sensor is implemented and fully integrated into the Willow Garage PR2 robot platform [1]. The setup is able to detect the presence of the object reliably in the close distance up to 5 mm.

In chapter 4, initial experiments are performed to explored two primary applications: (1) reactive grasp control — I demonstrated the advantages of pretouch over tactile sensing for objects that are too compliant during the grasp execution. (2) pretouch-assisted grasp planning — when the pretouch sensor sensed the object during a series of probing motions, it provided points collected by recording the position of the robot's end effector; these additional points augmented the point cloud from the PR2's depth sensor. This method compensated for the areas on the object that are missing from the point cloud either due to depth camera failure or occlusion.

For the latter application, in order to make the robot automatically determine the uncertain areas to sense for pretouch-assisted grasp planning, a unified probabilistic framework is proposed in chapter 5. The goals are: (1) identify shape uncertainty for the target object, and (2) enable automatic exploration with a pretouch sensor to reduce the uncertainty toward a grasp with high confidence. The robot first uses the information acquired by the depth sensor as initial data in a probabilistic mapping representation. Next, it uses priors on occlusion/transparency to make inferences about possible occluded and transparent portions of the object. An iterative grasp replanning and exploration algorithm decides when further exploration is required, and where to explore in the scene. The robotic grasping experiments are performed to evaluate the efficacy of the proposed framework.

---

[1] PR2 is a hardware platform for robot researchers designed by Willow Garage, Inc. It operates under the open-source Robot Operating System (ROS) framework. However, the research presented in this dissertation is general, and is not limited to a particular platform.

In chapter 6, I present an application in teleoperated robotic grasping scenarios. A method of augmenting streaming point cloud data with the seashell effect pretouch information for the purposes of haptic rendering in a telerobotic grasping task is presented. The non-contact seashell-effect pretouch sensor fixed to the robot end effector is used to sense physical geometries within the vicinity of the sensor. Thus, the point cloud representation of an unknown environment, which may be sparse or poorly visible, can be enhanced through telerobotic exploration/sensing in real-time. Furthurmore, real-time haptic rendering algorithms are applied on the augmented point clouds to create haptic virtual fixtures, and also provide haptic force feedback to the operator. This method provides the teleoperator with critical geometrical information about the grasp target while simultaneously preventing the robot end effector from colliding with it.

Finally, I summarize and conclude this dissetation and discuss potential future work in chapter 7.

Chapter 2

BACKGROUD

2.1 Robotic Grasping

In the last few decades, robotic technologies have been widely adapted in the manufacturing industry. Industrial robots are manipulators programmable and configured to execute various manipulation tasks, such as welding, assembly, pick and place, in predefined workspaces designed for specific tasks. Due to the repetitive nature of the tasks, the main focuses of the industrail robot are on the motion speed and accuracy. As the computational resources become more and more affordable, the interests in robotic system development in both research and industry have been migrating into general purpose personal/domestic robots operating in unstructured everyday environment. The manipulataion capability of these robots have the potential to improve the quality of life of human beings. However, robotic manipulation is an extremely challenging task mainly due to the unstructured environment. In such environment, perception capabilities and algorithms for reasoning about sensor information need to be considered. In addition, the motion of the manipulator needs to be planned and executed dynamically to adapt to uncertainty and avoid collision.

Among different manipulation tasks, robotic grasping has been one of the most sought-after capabilities. Grasping is essential to enable the manipulator to complete desirable tasks in daily life, e.g., picking up objects, cleaning up the table, or delivering objects. A lot of previous research efforts have been focused on the robotic grasping. Nonetheless, it still remain a challenging problem. Besides the inaccuracy of the end-effector motion control, a key difficulty is the lack of reliable perception, due to sensor calibration errors, occlusions, or simply the limitation of sensors.

2.1.1 Grasp Planning

Autonomous grasp planning for robotic hands is a difficult problem because of the huge number of possible hand congurations. To plan appropriate grasping poses for objects, different grasp synthesis algorithms have been proposed for different gripper designs. Sahbani et al. [69] and Shimoga [73] performed extensive reviews of the existing algorithms in terms of the four aspects of the synthesized grasps: dexterity, stability, equilibrium, and the dynamic behavior. Grasp synthesis algorithms are analytical or empirical algorithms to look for grasp configurations for achieving the immobilization of the grasped objects (equilibrium) againts the external disturbance. There are two methods of defining the equilibrium of the grasp:

- Form closure – the equilibrium is established soley based on a set of contact points on the object preventing all motion of the object [18, 77]. Mishra et al. [52] were the first to place an upper bound on the number of contact points needed for a form closure grasp on a frictionless object using a multifingered dexterous hand.

- Force closure – the equilibrium is established by contacts maintained by externally applied forces from the robot hands [7, 8].

When applying grasp synthesis algorithms, the object shape has to be first represented in certain forms on which the algorithms can run. Miller and Allen proposed to model an object as a set of shape primitives, such as spheres, cylinders, cones and boxes [51]. Graspit!, a robotic grasping simulator focusing on fouce-clousre grasps was then used to provide the best grasp. The simulation framework has been widely used by the robotic community in grasp planning and evaluation [50].

Although grasp synthesis is an effective way for grasp planning, a common pitfall is the uncertain object shape caused by imperfect robot perception, since synthesis algorithms assume complete knowledge of the object shape. The planned grasps solely based on the incomplete observed data usually lead to failures during the actual execution. Furthermore, the algorithms rely on the kinematics of the robot hands. Often multifinger robot hands are required to establish either the form closure or force closure equilibrium. When using

a simpler gripper, e.g., a 2-jaw parallel gripper, the theoretical equilibrium can not be established.

2.1.2 Object Recognition

The problem of incomplete object shape may be addressed by model-based object recognition approaches. Various studies have shown that objects may still be recognized based on partially-observed shape from the sensor data. For instance, Lamdan et al proposed an algorithm assuming affine approximation to the prospective viewing transformation to recognize partially occluded objects [40]. Aldoma et al. developed a fast and accurate 3D feature, Clustered Viewpoint Feature Histogram (CVFH), for use in object recognition and pose estimation for rigid objects given partially occluded point cloud obtained from a depth sensor [1].

In the cases that multiple possible object models in the database are matched with the target object, probabilistic methods can be used to select grasps that are robust to incorrect object recognition as well as other factors, such as motion error. Hsiao and Brook proposed a method using a Bayesian model to estimate the probability of success of each grasp, combining information from multiple object recognition and grasp evaluation algorithms [11, 28].

When recognizing objects for the purpose of grasping, an additional step is required – finding good grasping poses for each model. It can usually be done offline using grasp synthesis methods in similation (e.g. Graspit!) for each model. The designated best grasps for each object model can then be used to execute the actual grasping online after the target is matched with a model.

In order to achieve higher object recognition rate, training is usually required to be done on a broad coverage of the objects in order to match the target with the correct object model. Otherwise, wrong models may be matched and wrong grasping poses will be assigned. Methods of learning graspable features from the training objects have been researched to generalize the learned models for novel objects which are never seen before, and thus reduce the need of massive training data. Saxena et al. [70] showed a learning

algorithm to enable robots to grasp previously unseen objects. The method learns a classifier for finding grasping points as a set of pixels in 2D images. Furthermore, the training phase can potentially be facilitated by taking advantages of the recent progress in big data and cloud computing technologies. The amount of data about different object models on the cloud is abundent and can be used for offline training. Lai and Fox [39] showed a method to signicantly reduce the need for manually labeled training data by leveraging data sets available on the internet. The objects from Googles 3D Warehouse are used to train an object detection system for 3D point clouds sensed by robots. Additionally, domain adaption was performed to deal with the different characteristics of the web data and the real robot data. Kehoe et al. [36] presented a system utlizing the Google object recognition engine hosted on the cloud to scale the training and the recognition process. Their initial prototype can achieve 80% recognition rate.

Nevertheless, all of the methods mentioned above can still be very sensitive to the view point of the camera/depth sensor. For instance, in a case where the handle of a mug is on the backside and is occluded by itself, no graspable features may be found by the classifier.

2.1.3 Augmented Perception

Another area of research to improve the grasping performance is to improve both the quality and quantity of the information provided by sensors. Depth sensing RGB-D camera systems, including textured stereo cameras and structured infrared depth cameras, have dramatically improved recently. However, these sensors still frequently fail to provide data, either because of geometric difficulties (such as occlusion), or material difficulties (transparency or specularity).

Due to the limitation of the depth sensors, local tactile or pressure sensors have been widely used in the robotics community for the manipulation tasks in order to acquire a more complete object shape. Hsiao et al. proposed the use of contact sensor on partially visually sensed unknown objects [27]. A reactive adjustment approach is also implemented to deal with uncertainty in object location and shapes. They showed this approach can correct for a fair amount of uncertainty in the measured position or

shape of the objects, and increasd the success rate on grasping objects with a variety of shapes using a simple 2-jaw parallel gripper. Petrovskaya et al. demonstrated the use of tactile sensors for global object localization, and proposed an efficient Monte Carlo algorithm for realtime applications [56, 57]. A software architecture for grasping of household objects was developed [13], which tried to address the uncertainty due to sensor and excution error from the software engineering aspect by tighting up the coupling between perception (visual and tactile) and manipulation. However, it is still desirable to have more sensor information in order to have more complete shape of the objects.

These approaches rely on tactile sensing are subject to difficulties such as unintentional displacement of the object by the robot manipulator. In cases of involving high-value or delicate physical structures, it is imperative that the operator is able to make the robot avoids unintended contact with the object based on non-contact sensor data.

2.2 Pretouch Sensing

"Pretouch" refers to a short range, non-contact sensing modalities integrated into robot manipulators that are intermediate in range between long-range depth and tactile sensing. The aim of pretouch sensing is to combine the desirable features of touch and vision sensing: like touch, it provides information about the relative geometry of the hand and the object, in the hand's coordinate frame; like vision, it is non-contact and does not disturb the object's position. This is potentially beneficial for robotic manipulation as it provides reliable geometric information in the last centimeter before contact.

Different pretouch sensing modalities has been proposed previously such as optical [29, 48] and E-field pretouch [49, 75, 81]. However, the optical pretouch has similar drawbacks as common depth sensors for its incapability to sense transparent and reflective objects. On the other hand, E-field pretouch only works for objects with conductive materials or materials with high dielectric constant. A new pretouch sensing modality is desired to complement the variety of the materials the robot can sense.

Chapter 3

SEASHELL EFFECT PRETOUCH SENSING

In this chapter, I introduce a novel pretouch technique, "seashell effect pretouch", which is effective for a set of materials that previous pretouch techniques fail to sense. This pretouch modality is inspired by the phenomenon of "hearing the sea" when a seashell is held to the ear; in particular, the observation that the "sound of the sea" changes as the distance from the seashell to the ear varies.

This section is organized as follows: in section 3.1, related work about a similar sensing principle, "acoustic resonant shift", are reviewed. The acoustic theory behind the seashell effect is discussed in section 3.2. The sensor design and implementation including the acoustic, hardware, software design, and signal processing are discussed in detail in section 3.3. The performance the proposed sensor in terms of the sensing range, accuracy, and the material selectivity are characterized and evaluated in section 3.4. Finally, in section 3.5, I discuss the modularity and flexibility of the proposed sensor system design. A optical pretouch sensor can be implemented with minimal modification on the exsisting hardware and software.

3.1 Related Work

The idea of acoustic resonant shift is widely used in highly sensitive mass sensors for chemical and biological environment [4]. A typical acoustic mass sensor uses the fact that the resonant frequency of an acoustic-wave resonator changes in response to the mass load applied on the resonator's surface. For example, Zhang et al. [83, 84] devised a micromachined film bulk acoustic resonator mass sensor built on a micromachined silicon-nitride diaphragm with a piezoelectric thin film and Al electrodes that can operate in vapor and liquid. Its resonant frequency drops linearly with added mass on the surface. The shift of the acoustic resonant frequency is measured from the longitudinal standing wave existing between two faces of

the electrodes sandwiching a piezoelectric film, and the mass load and be inferred from the frequency shift.

In the design of the seashell effect pretouch sensor, I design an acoustic system using a cylindrical pipe with one end open and the other end closed with a hard boundary (a closed-open ended cylindrical pipe), in which the resonant frequency is shifted by the reactive radiation impedance (small vibrating air mass) change at the open termination of the pipe caused by the obstacle.

3.2 Acoustic Theory

The seashell effect is the phenomenon of "hearing the sea" that is observed when a seashell is held to the ear. The sound is the ambient noise amplified (attenuated) with the seashell cavity's acoustic frequency response. Inspired by the fact that the sound of the sea changes as the distance from the seashell to the head varies, a pretouch sensor is essentially an acoustic cavity (an closed-open pipe in our case) attached to a microphone that detects the change in the ambient sound spectrum that occurs when the pipe approaches an object. When an object approaches the pipe opening, the sound field between the surface of the object and the pipe opening causes a change in the effective (acoustic) length of the pipe. Perhaps counterintuitively, the total effective length of the pipe increases as the sensed object approaches; thus the resonant frequency of the pipe decreases as an object approaches. Similar effects were also studied for the woodwind musical intruments which have keys (buttons) hanging above the tone holes [14]. The acoustic theory is summarized in this section.

3.2.1 End Correction of Cylindrical Pipes

The shift of the resonant frequency caused by the object can be best explained by using the terminologies of acoustic impedance and end correction. The acoustic impedance (Z) is defined as:

$$Z = \frac{P}{U} \tag{3.1}$$

where P and U are the amplitude of the sound pressure and volume velocity, respectively. For an ideal closed-open pipe, the closed end is a rigid termination, and the acoustic impedance at the closed end (Z_c) is infinity ($Z_c = \infty$). At the open end, the sound wave is small compared to the atmospheric pressure, so the open end acts as a release termination at which the pressure vanishes ($P = 0$), and results in zero acoustic impedance ($Z_o = 0$). The reflection coefficient at the interface is defined as:

$$R = \frac{Z_o - Z_{air}}{Z_o + Z_{air}} \tag{3.2}$$

where Z_{air} is the characteristic acoustic impedance of the air ($Z_0 = 415\ Ns/m^3$ for air at 20 °C). With $Z_o = 0$, the reflection coefficient is -1, meaning that all the sound pressure is reflected at the open end and back to the closed end. This results in a total reflection in anti-phase. However, in reality, the open end is in fact terminated by a small finite radiation impedance as opposed to zero impedance. The effect of radiation can be approximated by considering the wave in the pipe accelerating the final layer of the air back and forth as a small mass. The radiation impedance (Z_r) seen by this final layer has a general form of:

$$Z_r = \frac{Z_{air}}{S}(R + jX) \tag{3.3}$$

where $S = \pi a^2$ is the area of the pipe [m^2], where a is the pipe radius; R and X are functions of the wave number k, the pipe radius a, and most importantly, the geometric configuration in the environment around the opening. R and X represent the real part of the impedance (acoustic resistance) and the imaginary part of the impedance (acoustic reactance) to the characteristic impedance, respectively. The imaginary part corresponds to the impedance of the small mass resulting from a volume of $S\Delta L$, and shifts the position where antiphase reflection occurs to a virtual plane outside the tube by ΔL, which is usually called an *end correction*. The end correction for a cylinder pipe with infinite flange can be solved analytically using Rayleigh integral for the Helmholtz equation [37]:

$$\Delta L = 0.8488a \tag{3.4}$$

where a is the radius of the pipe. This serves as the upper bound for the end correction of cylindrical pipes. The end correction for cylindrical unflanged pipes was solved under plane

wave assumption (for frequency lower than the first cut-off frequency of the pipe) [44]:

$$\Delta L = 0.6133a \tag{3.5}$$

which serves as the lower bound. End correction for other complicated geometries and at different configurations are usually found by numerical and experimental methods [74].

In this study, I am interested in knowing the effects of obstacles presented close to the opening of a closed-open pipe, specifically, the distance between the object and the opening. In addition to the regular end correction term ΔL (caused by the radiation impedance at the opening), the presented object further restricts the space for sound wave propagation and results in more end correction effects. Dalmont et al. [15] presented an emperical formula of this additional end correction term ΔL_{obj} caused by the presenting obstacle:

$$\Delta L_{obj} = \frac{a}{3.5(h/a)^{0.8}(h/a + 3w/a)^{-0.4} + 30(h/d)^{2.6}} \tag{3.6}$$

where a is the radius of the pipe; h is the distance between the obstacle and the pipe opening; w is the thickness of the pipe wall; d is the width of the object. Considering this additional end correction, the effective length of the pipe becomes:

$$L_{eff} = L + \Delta L + \Delta L_{obj} \tag{3.7}$$

In other words, the object presented close to the opening of the closed-open pipe will increase the effective length of the pipe by ΔL_{obj}. Given that the parameters a, w, d in equation (3.6), the pipe length L, and the radiation end correction ΔL are all fixed for a specific object and a pipe, L_{eff} becomes a function of only the object distance h, and is approximately inversely proportional to h:

$$L_{eff} \propto\!\!\!\approx \frac{1}{h} \tag{3.8}$$

This approximation gives the intuition that the effective pipe length is actually longer when the object is closer to the pipe opening.

3.2.2 Resonance Frequency

The goal of this work is to utilize this radiation impedance change at the open end (or the effective pipe length) to develop a sensor which can find the distance h. Because the

effective pipe length ΔL_{eff} changes when an object is presented near the opening, it is intuitive to think that by monitoring the change of ΔL_{eff}, the distance of the object h from the pipe opening can be inferred. However, it is not possible to measure the change of ΔL_{eff} directly. Alternatively, I make use of the fundamental resonance frequency of the standing wave in a closed-open pipe f_0 given by:

$$f_0 = \frac{c}{4L_{eff}} \tag{3.9}$$

where c is the speed of sound (343.2 m/s at 20 °C) [38]. From equations (3.8)(3.13), we can conclude the distance h is approximately proportional to the resonance frequency f_0:

$$h \lessapprox \frac{1}{L_{eff}} \propto f_0 \tag{3.10}$$

Based on this relation, the distance of the object h can be inferred directly from the fundamental resonance frequency f_0. The signal processing methods and the algorithms used for monitoring the resonance frequency in real-time will be discussed extensively in section 3.3.5.

3.3 Sensor Design

In this section, the design of this seashell effect pretouch sensor is discussed in detail. It involves considerations on acoustics, hardware (electronics), software/hardware interface, and signal processing. Given the complexity of the system and the large number of design variables, performace metrics are defined in 3.3.4 and are used to determine appropriate parameters in the design process, and are also used to characterize the resulting sensor in section 3.4.

The main sensor hardware consists of a brass open-ended pipe, and a microphone attached to one side of the pipe to form the closed end. The microphone is used to collect the sound pressure at the closed end filtered by the acoustic cavity (i.e., the closed-open pipe). To avoid being confounded by features in the raw (unfiltered) ambient audio itself (including loud ambient sounds), a reference microphone is used to collect environmental sound which is not filtered by the pipe; this background spectrum is subtracted from the

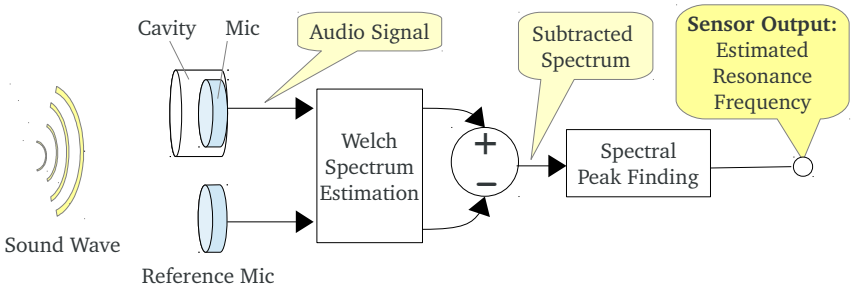


Figure 3.1: Illustration of the seashell effect pretouch sensor system.

spectrum of the actual pretouch sensor channel. This noise cancellation approach substantially improves sensing accuracy. Figure 3.1 illustrates the signal flow of the seashell effect pretouch sensor. The schematics for the printed circuit board, and source code for the microcontroller/soft-processor used in this work are open-sourced, and can be found in the appendix in this thesis.

3.3.1 Acoustic Design

The goal of the acoustic design is to engineer the acoustic system, specifically, the geometry of the closed-open pipe, such that the resonance frequency shift due to the presenting object is large enough to be monitored while remaining within the suitable frequency band. The larger swing of the resonance frequency can increase the sensing resolution (larger frequency shift when the object moves). At the same time, the geometry of the pipe should be compact enough to fit into a robot's finger for robotic applications.

During the acoustic design, equations (3.5), (3.6), (3.7), and (3.13) are employed throughout the process. Although the end correction term due to the pipe geometry (3.5) is under the assumption of a plane wave, and the end correction term (3.6) due to the presenting object is based on approximation as opposed to exact solutions, they serve as useful guiding references.

The sensor acoustic characteristics are mainly determined by two geometric parameters: the pipe length L and the pipe radius a. Some considerations should be taken when choosing the two parameters:

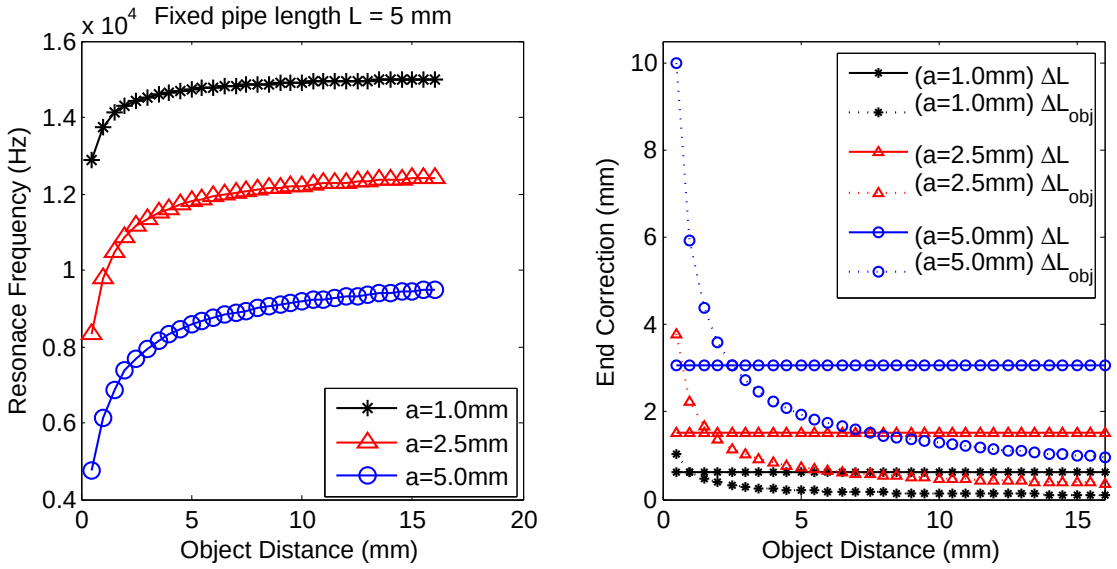


Figure 3.2: (Left:) resonance frequencies of the pipe at different distances with fixed length (L=5 mm) and various radius size. (Right:) end correction of the pipe at different distances with fixed length and various radius size.

1. As a rule of thumb, the plane wave assumption should hold in order to have predictable acoustic behavior. It requires the resonance frequencies in the working range of the sensor to be lower than the first cut-off frequency in the circular pipe ($ka < 1.8412$), so that only the fundamental mode will propagate.

2. From (3.6), the end correction caused by the object (ΔL_{obj}) is approximately inversely proportional to the ratio h/a. It means that using a larger pipe radius a could increase the amount of end correction changes at different object distances, and thus achieve better frequency resolution given a fixed pipe length L. Figure 3.2 shows the end correction values and the estimated resonance frequencies calculated based on equations (3.6) and (3.13). When the object moves from 5 mm to almost 0 mm (just before contact), the resonance frequency shifts approximately from 8500 Hz to 5000 Hz (change of 3,500 Hz) and from 14,500 Hz to 13,500 Hz (change of 1000 Hz) for the pipes with a=5 mm and a=1 mm, respectively. The frequency shifts more when using a larger pipe opening, and thus increase the sensing resolution. This is becuase

the fixed end correction term determined by the pipe opening ΔL and the varying term determined by the object distance ΔL_{obj} are both larger for the pipes with larger opening radius. However, with a larger opening radius, the lateral resolution will be reduced. The larger pipe also makes it harder to integrate the sensor into the robot's gripper. This trade-off should be taken into account when selecting the pipe radius.

3. In the case of a fixed pipe radius a (in other words, fixed end correction ΔL and ΔL_{obj}), using a shorter pipe length L results in larger frequency shift. This can be explained by equations (3.7) and (3.13):

$$f_0 = \frac{c}{4L_{eff}} = \frac{c}{4(L + \Delta L + \Delta L_{obj})} \tag{3.11}$$

Assuming the resonance frequency shifts from $f_{0,m}$ to $f_{0,n}$ when the object moves from distance m to n, the amount of resonance frequency shift Δf_0 is:

$$\Delta f_0 = f_{0,n} - f_{0,m} = \frac{c}{4(L + \Delta L + \Delta L_{obj,n})} - \frac{c}{4(L + \Delta L + \Delta L_{obj,m})} \tag{3.12}$$

Since the end correction change during the movement is independent of L, the amount changed is also fixed when the pipe radius a is fixed. Therefore, Δf_0 is approximated inversely proportional to the pipe length L:

$$\Delta f_0 \propto\!\!\!\sim \frac{1}{L} \tag{3.13}$$

Figure 3.3 shows that the shortest pipe (L=2.5mm) has the largest resonance frequency shift (the best dynamic range in frequency). However, the cavity is considered as a lumped-element (a mass) when the nominal dimension (the length) is short compared to the wavelength ($kL_{eff} << 1$) [9]. The design should avoid this region, since it will make the resonance behavior different from the theoretical analysis presented here and become hard to predict in practice.

According to the above three design considerations and some experimental verification, a pipe length L=5 mm and radius a=2.5 mm is selected, which has a compact size to be

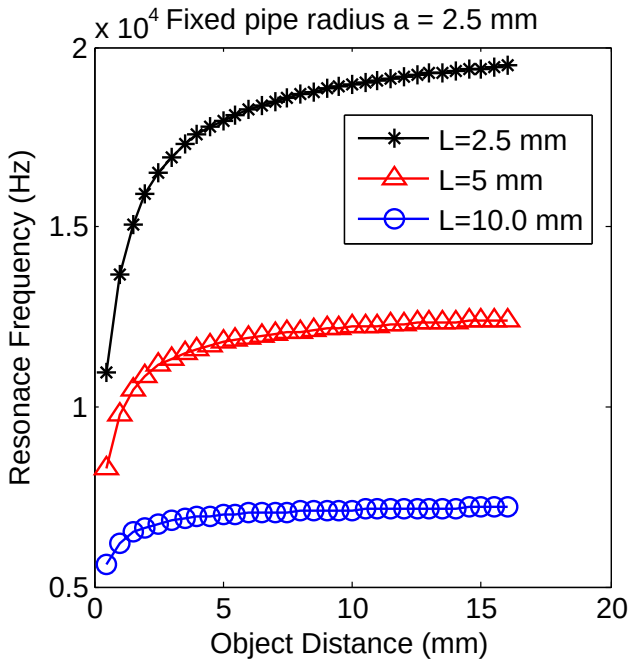


Figure 3.3: The computed resonance frequencies of the pipe at different distances with fixed radius size (a=2.5mm) and various pipe length L.

integrated into the PR2 robot's fingertip. The frequency shift is guaranteed to be in the range of $[6000, 12000]$ Hz for the sensing distance range of $[0, 10]$ mm.

3.3.2 Hardware Design

A printed circuit board (PCB) is designed using a EDA software (Altium) for the required analog circuit and electronic components of the sensor. The schematics, PCB layout, and the detail electronic components are included in appendix A.1 and A.2.

The sensor finger's mechanical structure is designed in a 3-D CAD software (Solidworks) and fabricared by a 3-D printer. It holds all of the electronics and mechanical components, including the microphone and the pipe. The cavity used in our system is a 2.5 mm radius / 5 mm length closed-open end cylindrical brass pipe attached to a 2.5 mm radius / 3 mm length microphone, which is embedded in the PR2 gripper's fingertip. Figure 3.4(b) shows the PCB and fingertip fixture with all the components attached, including a microcontroller, a microphone, an acoustic cavity, and other electronic components. Figure 3.4(a) shows the

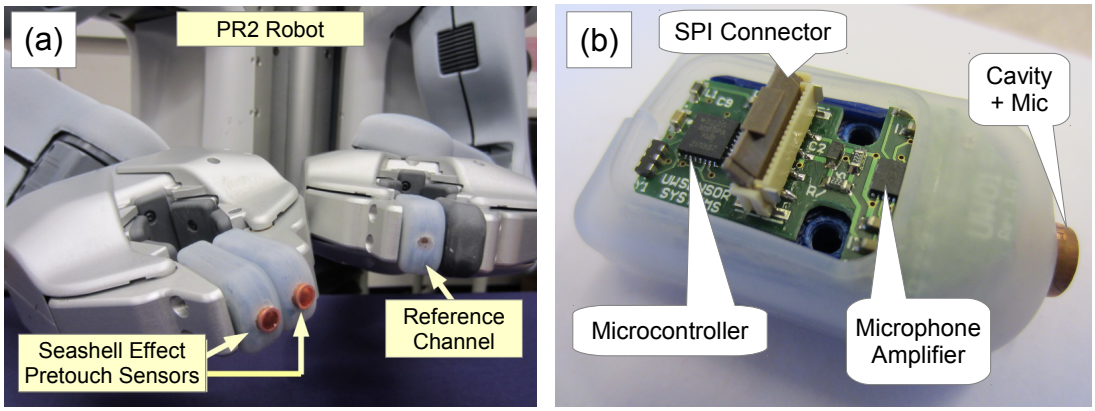


Figure 3.4: (a) The seashell effect pretouch sensors installed on the Willow Garage PR2 robot grippers. (b) The sensor system consists of a PCB, a microcontroller, electronic components, a microphone, and an acoustic cavity integrated into the PR2 robot's fingertip.

completed pretouch sensing fingertip installed on the Willow Garage PR2's gripper.

In the current implementation, there are two fingers on one of the grippers as the actual sensing channels, and a reference channel on another gripper for spectrum subtraction. The only difference between the design of the actual sensing channels and the reference channel is that the microphone on the reference channel is not attched to the pipe (acoustic cavity), so it simply collects the ambient (unfiltered) sound. Unlike the first prototype of the sensor [33], the new sensor is completely integrated into the Willow Garage PR2 robot; all external cables and electronics have been eliminated. The sensor uses the PR2s existing hardware interfaces for power and SPI data transmission. The embedded sensor design eliminates the constraints on robot arm motion caused by the external wires and electronics, and thus broadens the applicability of the sensors.

3.3.3 Hardware/Software Interface

The sound signal path implementation is shown in figure 3.5. The pipe cavity filters the ambient noise, and the sound signal collected by an electret microphone (Panasonic WM61-A) is amplified by 40 dB through a low-noise microphone amplifier (Maxim MAX9814),

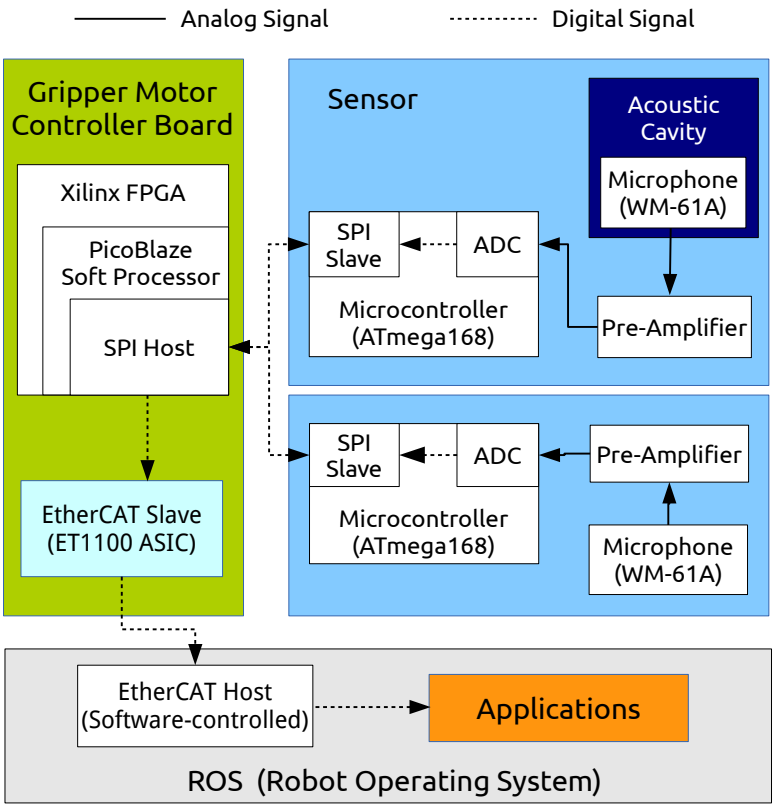


Figure 3.5: The system architecture and the signal flow of the seashell effect pretouch sensor system.

and is sampled by the 8-bit Analog-To-Digital Convertor (ADC) on a 8-bit microcontroller (Atmel ATmega168PA). The sampled data is then transmitted from the microcontroller to the 8-bit soft soft processor (Xilinx PicoBlaze) residing in the FPGA inside the PR2's gripper via the Serial Peripheral Interface (SPI) communication protocol. Finally the sampled sound data is accessible from the FPGA to Robot Operating System (ROS) in the PR2 robot through the EtherCAT interface. The sampling rate F_s for this whole signal path is 44,100 Hz for each channel, resulting the Nyquist frequency of 22,050 Hz, which is well beynod the designated sensing range of $[6000, 12000]$ Hz. The firmware of the microcontroller and the PicoBlaze soft-processor used for the seashell effect pretouch sensor is included in appendix A.3 and A.4, respectively.

3.3.4 Metrics for Sensor Performance

Given the complexity of the design variables, two performace metrices are defined here. They are used to determine appropriate parameters in the signal processing step, and are also used to characterize the final sensor performance.

The Contrast-To-Noise Ratio (CNR) is defined as:

$$CNR := \frac{\overline{f_{0,n}} - \overline{f_{0,1}}}{\overline{\sigma(f_{0,h})}}, h = [1, n] \tag{3.14}$$

where h is the object distance in millimeter, n is the farthest distance thich the sensor can sense, f_0 are the measured resonance frequencies, and σ is the standard deviation. This is the frequency difference between the farthest (10mm) and closest (1mm) distance devided by the average of the standard deviation at each distance in a interval of 1 mm within the range. It measures the ability of the sensor to distinguish between the farthest and closest distance, subject to sensor variability. This metric is useful for characterizing ground truth performance of the sensor, since one main use case of the sensor is to be used to discriminate whether an object is presented close enough to the sensor or not.

The second indicator Signal-To-Noise Ratio (SNR) is a measurement of how prominently the peak frequency stands out in the subtracted power spectral density (PSD) compared with the noise level of the environment. It is defined as:

$$SNR := p_{max} - \overline{p_i} \tag{3.15}$$

where p_{max} is the value of maximum power densiry in the spectrum, and $\overline{p_i}$ is the averaged value of thepower spectral density (dB/Hz) of the whole spectrum. The unit of the power The higher the SNR is, the more likely the peak (resonance) frequency can be detected precisely from the spectrum.

3.3.5 Signal Processing

The sensor designed until this section is just a pair of microphones enclosed in a carefully designed acoustic cavitiy. Signal processing steps are essential to convert the raw sound data collected by the microphones into useful information (the end results – sensor readings). In

this section, the detailed signal processing steps are presneted, including the description of the algorithms and the parameter selection. The appropriate parameters are selected experimentally using a grid search approach with the help of the metrics defined in section 3.3.4.

Power Spectral Density Estimation

The power spectral density of the sound signal from both the sensing channel and the reference channel are estimated using Welch spectrum estimation, a method based on time averaging over short, modified periodograms [80]. The method divides the time series data into overlapping segments, computes a modified periodogram using Discrete Fourier transform (DFT) for each data segment after applying a window (e.g., Hanning or Hamming data taper), and then averages the resulting power spectrum density estimates over segments. The method is known for its capability of reducing the variance of the estimated PSD resulting higher SNR.

Noise Reduction

A power spectral subtraction method is used to reduce the undesired acoustic noise [6, 10]. The PSD estimate of the signal collected from the reference channel is subtracted from the PSD estimate of the signal collected from the sensoing channel before peak finding. It avoids the effect of loud sounds, outside of the sensor's frequency range, misleading the peak tracking.

Peak Detection

To reduce the artifacts caused by the spectrun subtraction, a moving average low pass filter is used to further smooth the differential spectrum in the frequency domain. The resonance frequency is initially estimated by finding the frequency bin which has the maximum power density in the discrete subtracted PSD:

$$p_k = max(|p_i|), i = [1, \frac{N_{FFT}}{2}] \tag{3.16}$$

$$k = argmax(|p_i|), i = [1, \frac{N_{FFT}}{2}] \tag{3.17}$$

where N_{FFT} is the number of DFT points used for computing the discrete spectrum, which defermines the resolution of the resulting spectrum. The peak frequency f_{peak} is then refined by using a fast and accurate peak detection algorithm [31], which provides finer peak frequency estimates without the need for increasing the number of sampeles for PSD estimates. The refined peak frequency is given by:

$$k_{peak} = k + \tau \tag{3.18}$$

and

$$f_{peak} = \frac{k_{peak} F_s}{N} \tag{3.19}$$

where

$$\tau = \frac{K \times (|X_{k+1}| - |X_{k-1}|)}{(|X_k| + |X_{k-1}| + |X_{k+1}|)} \tag{3.20}$$

where X_k is the magnitude of the k-th distcrete frequency bin in the power spectrum, and K is a constant determined by the data taper used (e.g., $K = 1.36$ for the Hanning data taper).

Parameters Selection

There are several parameters in the Welch's PSD estimation, namely the number of samples N, number of DFT points N_{FFT}, the number of overlapped samples $N_{overlap}$, and the choice of the data taper. The best practice suggests 70% overlap for transient signal data due to the fewer available data and the desire of low variance. It results in $N_{overlap} = 0.7 \times N$ samples. Hanning window is chosen as the data taper, which tapers each data segment fully to zero on both ends to remove any discontinuity between data segments. It has less side lobe leakage while still maintaining a narrow main lobe. These characteristics are important for sucessfully estimating narrow band peaks in the signal.

The goal now is to find a combination of the remaining parameters N and N_{FFT} which can result in the best sensor accuracy. Besides the accuracy, the sensing rate is also an very important factor in practice. Therefore, a figure of merit (FOM) based on both the resulting sensing accuracy and the sensing rate is defined:

$$FOM := \sqrt{F} \times CNR \tag{3.21}$$

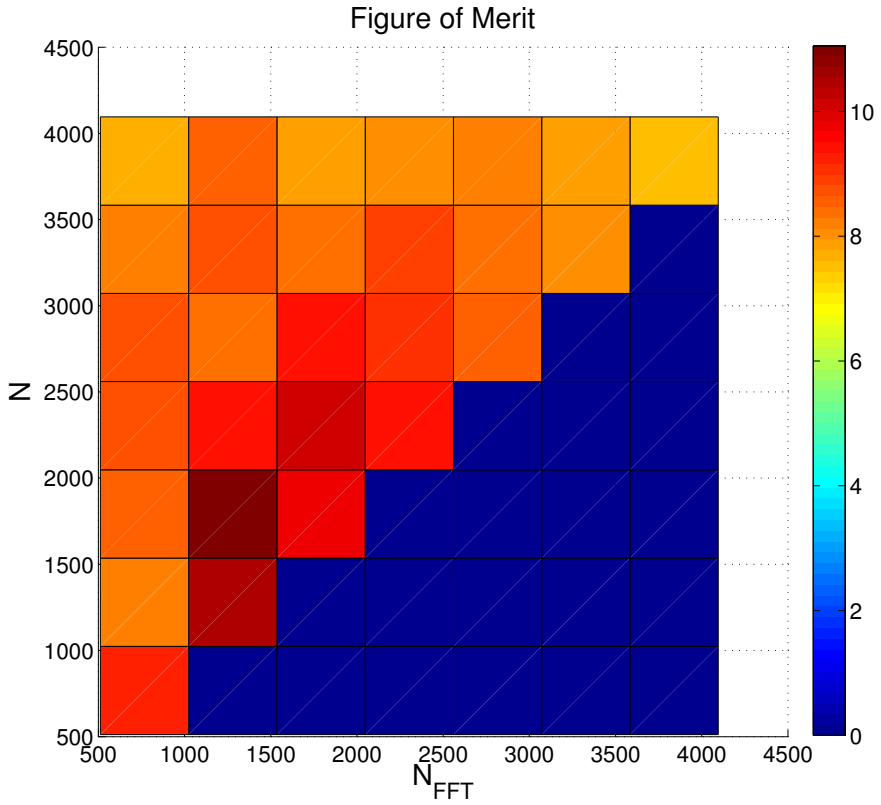


Figure 3.6: The figure of merit of the sensor based on the sensing accuracy CNR and the sensing rate F when using different parameters (N and N_{FFT}) in the Welch's power spectral density estimation.

where F is the number of sensor readings per second (Hz) computed by dividing the sound data sampling rate F_s by N:

$$F = \frac{F_s}{N} \tag{3.22}$$

and CNR is the sensing performance defined in equation (3.14). The FOM represents the overall capabilities of the sensor when applying different Welch's parameters . A grid search is then performed on the FOM with N and N_{FFT} being the tunable parameters.

The results of the grid search by tuning N and N_{FFT} is shown in figure 3.6. The parameters are selected to be $N = 2048$ and $N_{FFT} = 1024$, with the 70% overlap ratio and the use of Hanning data taper, which has the best FOM value of around 11. With the

sound data sampling rate F_s = 44,100 Hz of the system, the sensor reading rate F in this setup is around 20 Hz.

Example Results

Figure 3.7 shows a set of generated power spectral density spectrum during the whole signal processing pipeline from the sensor with a object presented. Fig 3.7(a)(b) show the original PSD estimated by the Welch's estimator using the selected parameters for the reference channel and the sensing channel, respectively. The spectrum of the sensing channel (Fig 3.7(b)) has a significant bump in the range of [6000, 12000] Hz, as opposed to a flat spectrum in the same range on Fig 3.7(a). As the low frequency contents are much more prominent in a regular daily-live environment as shown in Fig 3.7(b), the peak detection algorithm applied will actually pick up the lowest frequency as opposed to the peak of our interests caused by the amplification of the acoustic cavity. By applying the Boll's spectrum subtraction described in 3.3.5, the difference caused by the cavity can be clearly picked up in figure 3.7(c). To reduce the artifacts caused by DFT and the spectrum subtraction, a simple moving average is applied and a smooth spectrum can be obtained (figure 3.7(d)). The peak detection can then be applied to find the resonance frequency f_0 of the current acoustic system. In this particular case, f_0 is around 10,500 Hz.

3.4 Sensor Characterization

In this section, a set of experiments are performed to characterize the actual resulting sensor after the design process in section 3.3, including the resonance frequency shift when an object is presented at different distances, the effect of the object materials, and the effect of the ambient noise (the power density and spectral contents).

3.4.1 Resonance Frequency Shift

The resonance frequency change at different object distances is evaluated by collecting 1000 sensor readings (the resonance frequencies detected by the peak detection algorithm) at various distance from 1 mm to 10 mm with an increment of 1 mm. A box-and-whisker

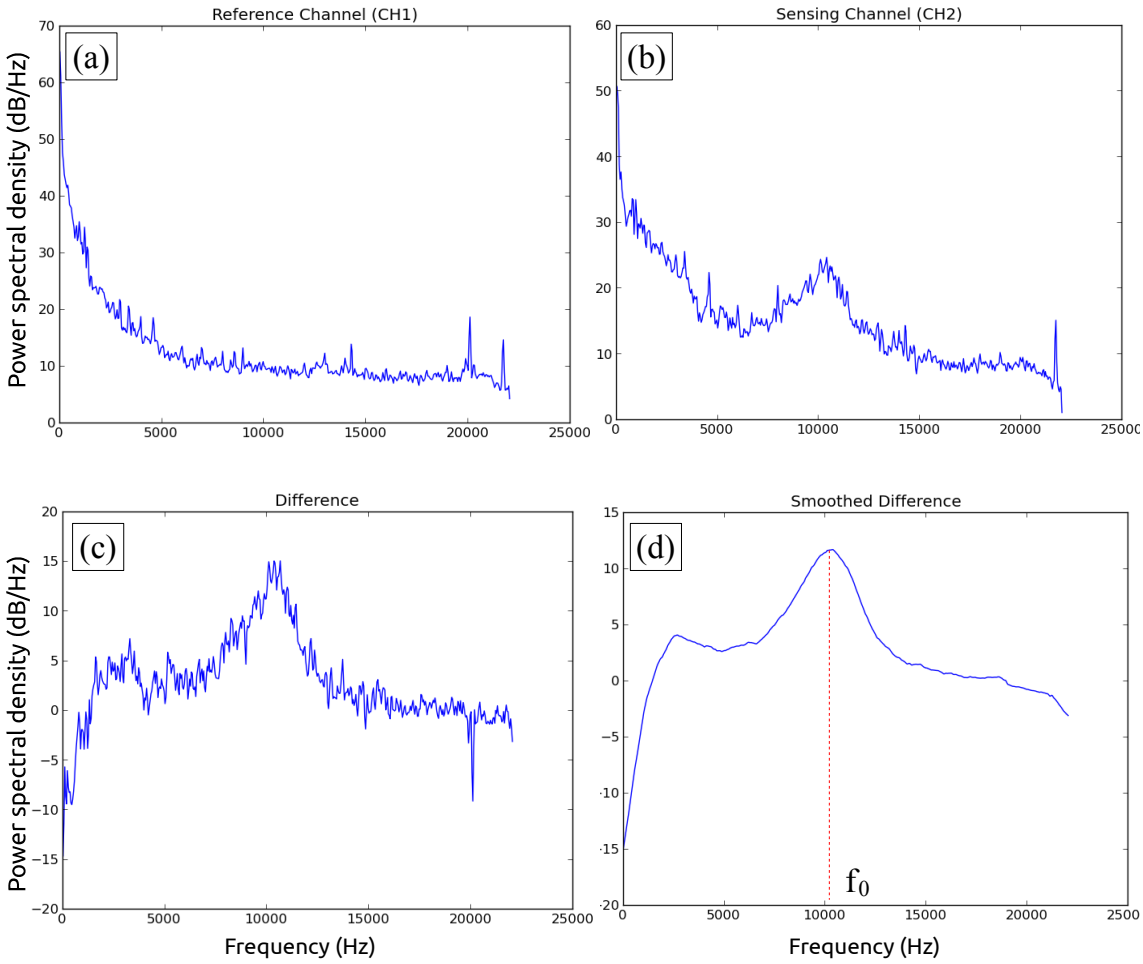


Figure 3.7: Example spectrums generated during the signal processing pipeline. (a) power spectral density of the reference channel. (b) power spectral density of the sensing channel. (c) power spectral density after noise reduction. (d) power spectral density after smoothing.

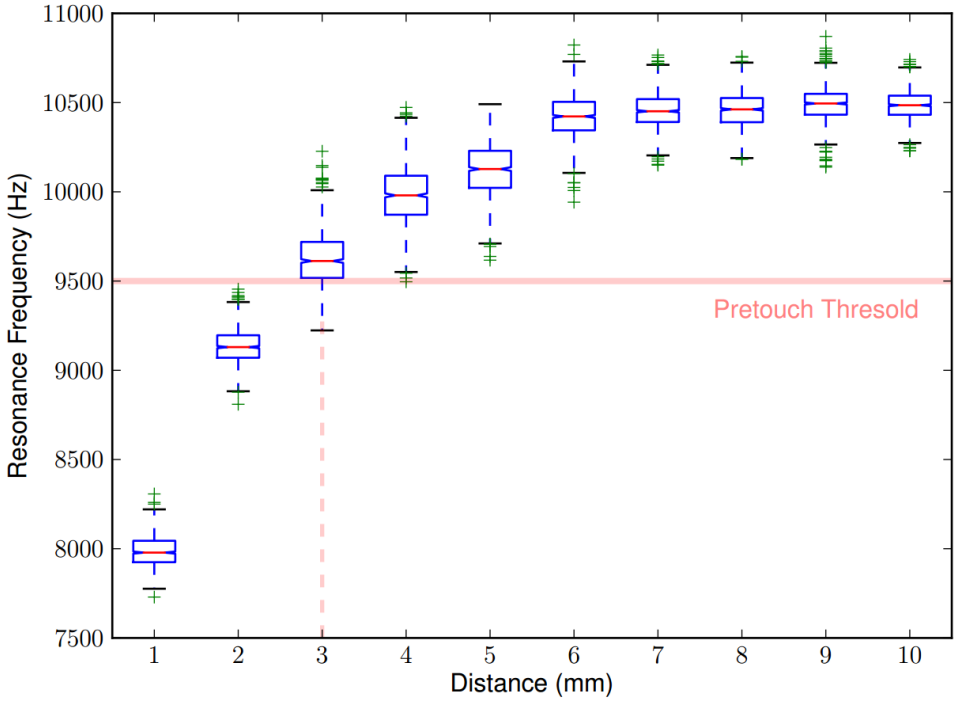


Figure 3.8: The box-and-whisker plot of 1000 estimated resonance frequencies at each distance. It represents the sensor characteristics of the sensor with length L=5 mm and radius a=2.5 mm integrated on the robot fingertip. The green crosses are outlier samples.

plot is used to present the performance of the sensor (figure 3.8). The resonance frequency increases monotonically from closer distances to farther distances, and becomes saturated at around 6 mm.

Based on this experimental data, a linear regression model can be fitted to obtain a mapping between the object distance and the resonance frequency. The sensor can then be used as a range sensor at close range below 5 mm.

Alternatively, for the application of detecting the presence of objects, a threshold at 9,500 Hz can be selected based on the experimental data. It is the lower quartile of the sensor readings at 3 mm. Since it is below the minimum at 4 mm, it can be very certain that the distance is below 4 mm when the sensor reading is less than 9,500 Hz.

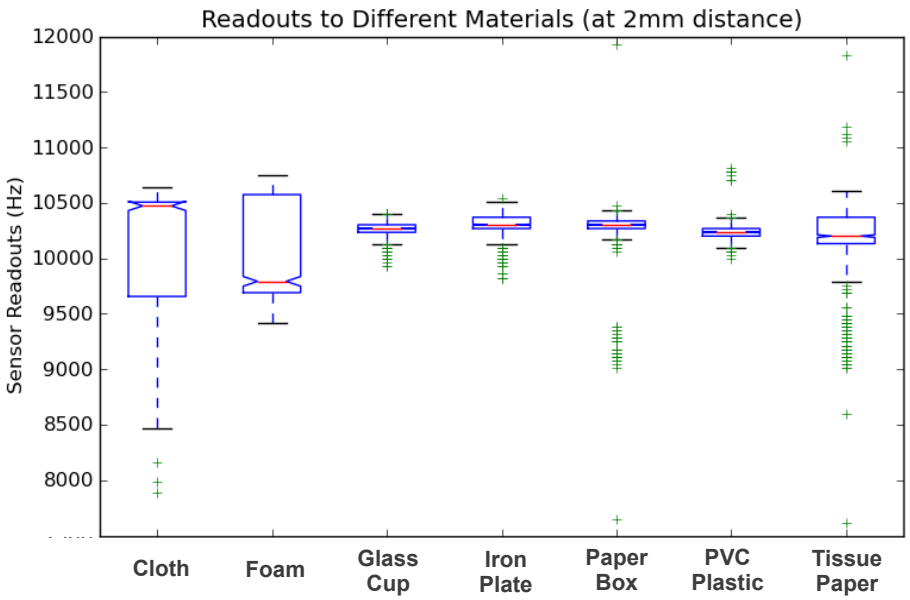


Figure 3.9: The box plot of the sensor readouts to different materials. 1000 readouts are measured for each object at 2 mm distance.

3.4.2 Material Sensitivity

An important motivation for developing the seashell effect pretouch sensor is its independence of optical and electrical material properties, which makes it a sensing modality complement to long range optical depth sensors and other existing pretouch sensors. Ideally, the sensor should only depends on mechanical (acoustic) properties. For example, seashell effect pretouch can sense highly transparent, reflective, or light-absorbing materials, which are difficult for optical sensors. In this section, I compare the sensor output for several different materials at the same distance.

1000 readings are measured for each object at a distance of 2 mm from the object's surface. The collected data is plotted in figure 3.9. The results show that the readings from materials with more porosity, such as cloth and foam, are more noisy than others. I hypothesize that these materials may acts as absorption or transparent materials depending on their thickness.

3.4.3 Effect of Ambient Sound

It is natural to wonder whether seashell effect pretouch could be improved by actively generating sound. Although the passive scheme has the advantage of easy integration, it is desirable to understand how much improvement could be achieved if extra sound is actively generated. In this section, I systematically investigate the effects of deliberately generated sound; I will examine sensor performance as a function of the sound's spectral contents and power density. Finally, I describe an adaptive active sound generation scheme that maintains consistent sensor performace regardless of ambient sound conditions.

The first factor is the spectral contents of the sound. Two waveforms with different spectral contents are experimented and compared. The first waveform, white noise, has an uniform power spectral density distribution over its frequency band from 0 to 22,050 Hz. Considering the fact that the sensor's resonance frequency is always within a certain range, it is intuitive to hypothesize that providing bandlimited white noise may be sufficient, or even more efficient. Therefore, the second wave form has the uniform spectral density limited within 6000 - 12000 Hz (the relevant range, as determined in our previous experiments). The lower and upper bounds are determined by experimental data: the sensor readout is confined to this frequency band for objects in its working range (0 - 10 mm). The waveform is generated by processing the white noise waveform with a bandpass filter. Figure 3.10 shows the spectral density of the two different waveforms used in this experiment.

The second factor investigated is the power level of the added sound. During the experiment, I measure the averaged power density of the sound for 30 seconds without a stimulus, and again for 30s with the stimulus. (The average is computed first over time, and then over all discrete frequency bins.) The difference is computed to find the power increase due to the stimulus. The difference value in units of dB/Hz is the average power density level increased due to the generated sound stimulus. When the waveform is played, the amplitude is adjusted such that the total power received by the microphone is at the target level. Three sound levels are tested: without external stimulus sound, 0.5 dB/Hz stimulus, and 1.5 dB/Hz stimulus. The microphone selected for the sensor system has flat frequency response from a very low frequency (20 Hz) to its highest frequency (20000 Hz),

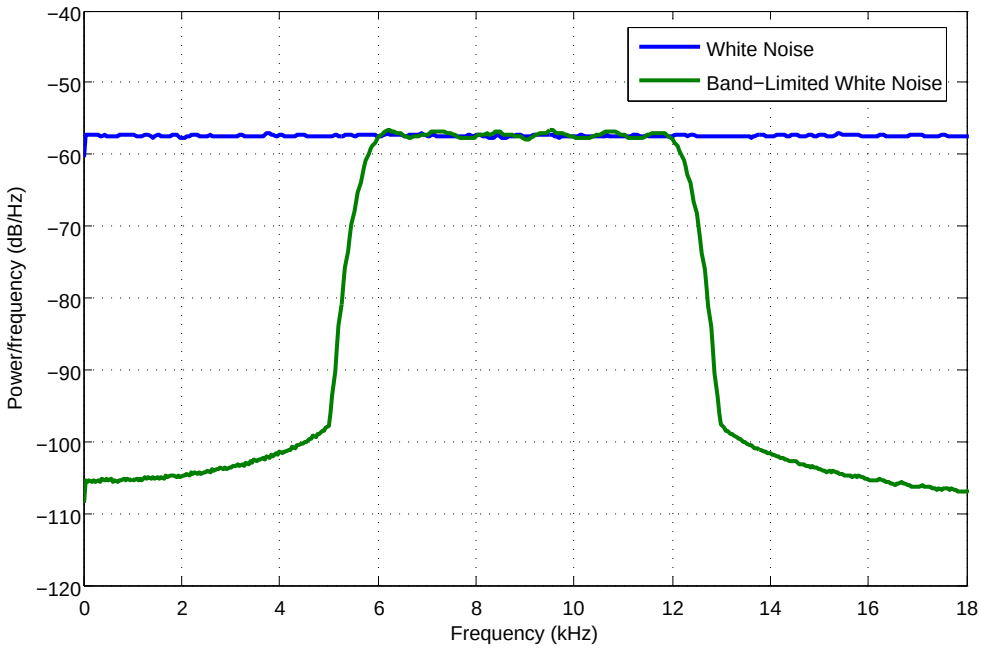


Figure 3.10: The spectrum of the two waveforms used for external sound characterization. The bandlimited white noise is the white noise processed by a bandpass filter that attenuates below 6000 Hz and above 12000 Hz.)

and the microphone amplifier has flat frequency response over 400 - 20000 Hz according to the manufacturer's datasheet. Our seashell effect pretouch sensor works in the range of 6000 - 12000 Hz, so the effects of the system response of the microphone and amplifier can be considered uniform for both waveforms in this experiment.

Using the combination of two different waveforms and three different sound levels, five experimental trials are performed: no stimulus, 0.5 dB/Hz broadband stimulus, 0.5 dB/Hz bandimited stimulus, 1.5 dB/Hz broadband stimulus, and 1.5 dB/Hz bandimited stimulus. For each trial, 1000 readouts are measured at each distance from 1 to 10 mm (in increments of 1 mm). Figure 3.11 shows the power spectral densities and the computed SNR for the five trials. Whithout providing extra sound stimulus, the peak power is weak compared with the environment sound, so any other noise bursting in the ambient sound can mislead the peak frequency estimation. By providing extra sound, the peak power at around 10,700 Hz is more prominent, so any other smaller peak appearing in the spectrum will not affect the peak frequency estimate. For the same stimulus power constraint, all the power in the bandlmited white noise stimulus falls in the sensor's working range (6000 - 12000 Hz), while the broadband stimulus essentially wastes some power (that outside the sensor's working region). Thus for the same allowed stimulus power, the bandlimited stimulus increases the sensor SNR more. Thus, the bandlimited white noise is more efficient, since I want to provide the smallest noise possible, both to avoid annoying nearby people who might hear the stimulus, and to prevent the robot / sensor system from expending unncessary electrical power.

The sensor readouts at each distance from 0 to 10 mm for each of the five cases; the CNR is computed for each case. Figure 3.12 shows that without providing external stimulus in a quiet environment, the sensor readings are spread out with no prominent peaks. When bandlimited white noise is provided at the average power density of 1.5 dB/Hz, the CNR is improved from 0.9 to 10.88 compared to the case without extra noise.

Table 3.1 shows the performance indices for all the experiment sets. From these results, I can conclude the following: (1) appropriate ambient noise spectrum and level is essential for good sensor performance. (2) the bandlimited white noise is more efficient to imporve the SNR and CNR compared with the white noise at the same power level. (3) CNR is

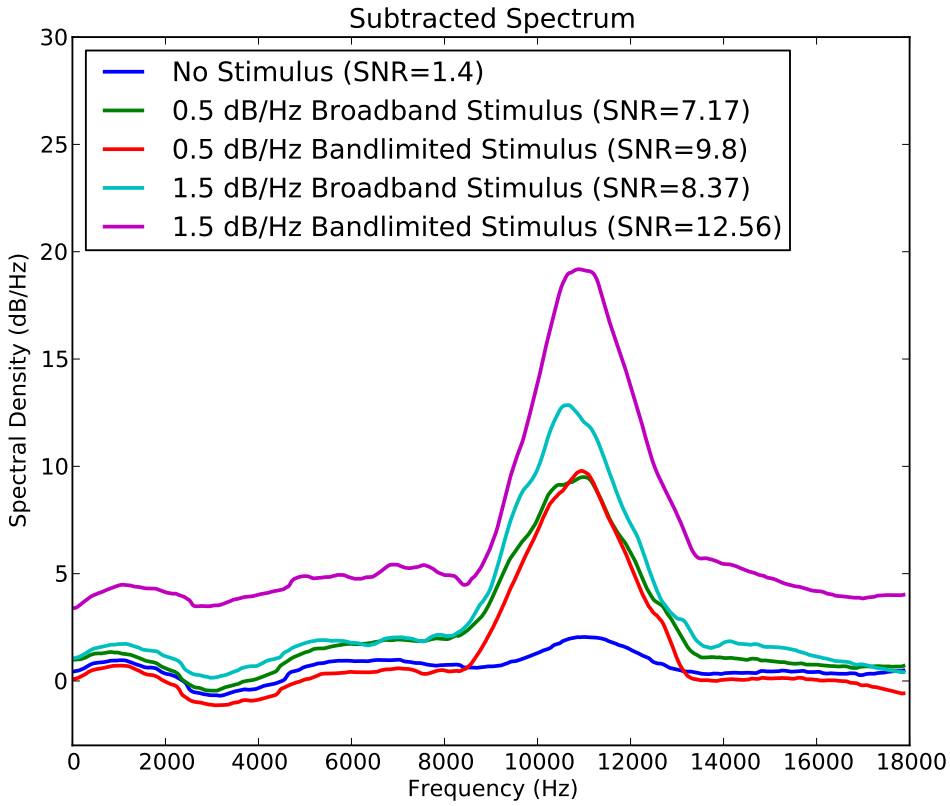


Figure 3.11: The subtracted spectrum (spectrum of sensing channel - spectrum of reference channel) at 10mm distance when extra noise is played with different waveforms and different power.

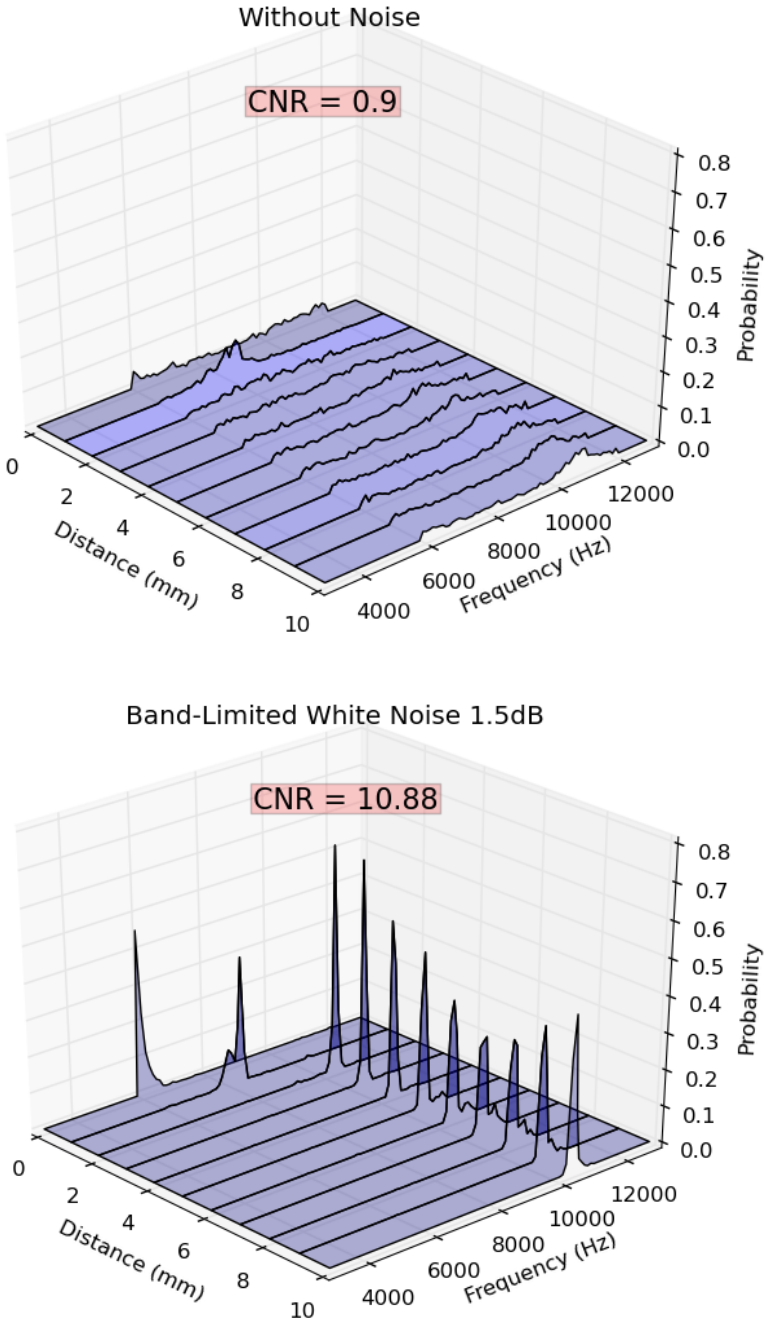


Figure 3.12: The sensor model created by collecting 1000 sensor readouts at each distance from 0 to 10 mm for the two cases: (a) In a quiet room without actively providing extra noise; (b) the extra bandlimited white noise is provided at 1.5 dB/Hz averaged spectral power density.

Table 3.1: Effect of the Extra Noise Spectrum on Sensor Performance

Waveform	Stimulus Noise Average Power Density (dB/Hz)	CNR	SNR
No Extra Noise	0	0.90	1.40
White Noise	0.5	6.57	7.17
Bandlimited White Noise	0.5	9,37	9.80
White Noise	1.5	8.45	8.37
Bandlimited White Noise	1.5	10.88	12.56

hightly correlated with SNR, which matches the intuition that the more prominent peak will facilitate the peak estimation, and thus the sensor measurement is more accurate. Based on these findings, I propose to implment an adaptive stimulus noise generation scheme described in the following section.

3.5 Modularity and Flexibility

The sensing system is designed with modularity and flexibility in mind. The same architecture can be used for developing other types of sensing modality with minimal modification on subsystems. One example is an infrared (IR) optical beam pretouch sensor developed to provide accurate detection of delicate objects. The sensor contains four IR emitter/receiver pairs, which create four IR light beams orthogonal to the surface of fingers. Objects can be detected between the pairs when one or some of the IR beams are interrupted (broken) by objects. The IR sensor is suitable for applications requiring detection of objects which the pressure and seashell effect pretouch sensors may not be able to sense due to the compliance, softness, porosity, and acoustic absorptivity.

One of the example applications of the IR beam sensor is on the robotic autonomous towel/cloth folding. Previously, this task has been shown to be feasible on a Willow Garage's PR2 robot [47]. However, the time spent to complete the task is still far beyond the desirable duration. One of the bottlenecks is to position the robot's finger at a desired location relative

to the object (the towel or clothes), for example, aligning the robot's finger with the corner of the towel. This task is in fact extremely difficult when using a tactile sensor, since the touch is usually too gentle. A simple, fast and reliable sensor for this particular task is desirable to speed up the whole process.

The IR beam optical pretouch sensor is implemented by simply modifying a portion of the seashell effect pretouch sensor system. Specifically, the sensor components are switched to photo/emitting diodes from microphones; a switch IC is added for selecting between the four IR beams (the schematic of the electronic circuit and the PCB layout are included in appendix A.5 and A.6); the firmware of the microcontroller (appendix A.7) and the PicoBlaze soft-processor (appendix A.8) are slightly modified to match the desired data rate. The system architecture after the modification is shown in figure 3.13. Figure 3.14 shows the resulting sensor installed on the PR2 robot's gripper. One side of the finger serves as the emitter to generate four IR light beams, and the other side of the finger serves as the receiver. Figure 3.15 shows the IR beam optical pretouch sensor sensing a compliant object. The object blocks on of the IR beams (channel 1), thus the presence of the object is detected.

3.6 Summary

In this chapter, I presented a short range proximity pretouch sensor based on the seashell effect inspired by the phenomenon of "hearing the sea" when a seashell is held to the ear. To the best of my knowledge, this effect has not previously been used to build proximity sensors. This sensor detects changes in the spectrum of ambient noise that occur when the finger approaches an object. The acoustic theory, design consideration (acoustic, hardware, software), signal processing, and the quantitative characterizations of the sensor are discussed in detail. The sensors are designed and integrated into the Willow Garage PR2 robot platform. The frequency shift of the sensor at different object distance, and the effect of ambient noise on the sensor performance, including the noise level and spectral content are characterized. A stimulus consisting of band-limited white noise with frequency content in the sensor's working frequency range is most effective, compared with a broadband stimulus, or random ambient sound.

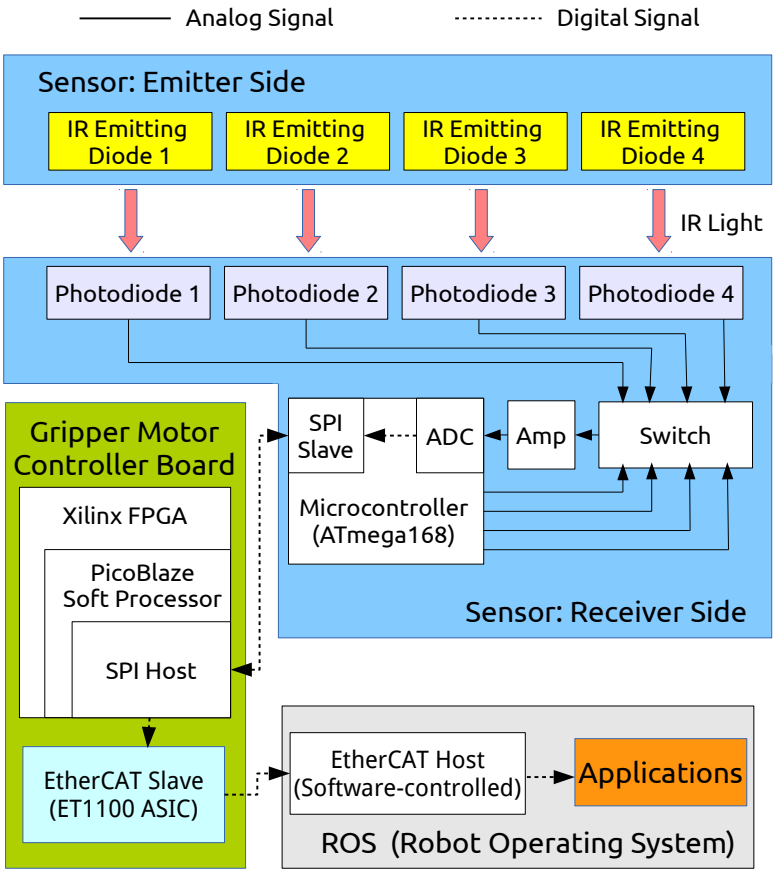


Figure 3.13: The system architecture and the signal flow of the IR beam optical pretouch sensor.

Figure 3.14: The IR beam optical pretouch sensor on the PR2 robot.

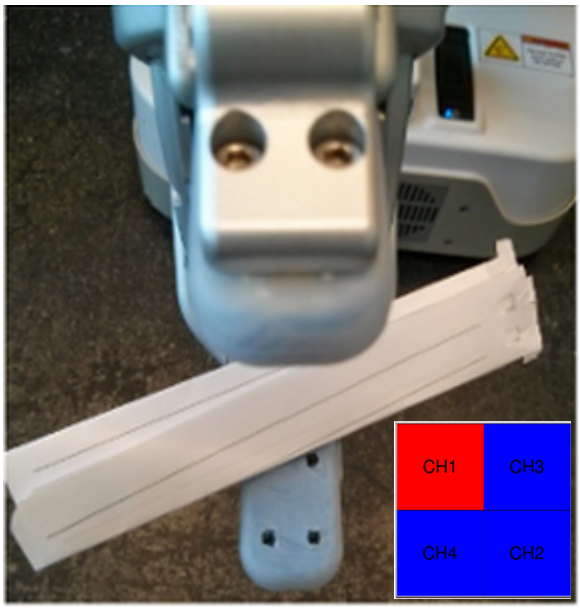


Figure 3.15: The IR beam optical pretouch sensor senses an object on one of the channels.

The setup is able to detect the presence of the objects reliably in the close distance up to 5 mm, which can provide a new source of information for robotic manipulation that complements long range depth sensors and contact-based tactile sensors. This sensing technique has a number of practical benefits compared to conventional sonar-based time of flight range sensors:

- Easy sensor integration — the sound stimulus source can be located anywhere. It does not have to be close to the sound receiver, which is an advantage for sensor integration into devices;

- Expensive hardware for short-interval time measurement is not required.

- Sensing accuracy is not affected by the notorious multi-path effects commonly seen in time-of-flight sensors.

Additionally, it can sense various types of objects depending on their mechanical properties. It makes the sensor a good complement to previously existed sensor such as optical, E-filed, and vision sensors [34].

Finally, the flexibility and modularity of the sensor system design enable us to devise other kinds of tip-mounted sensor with minimal modification on the existing system. An IR beam optical pretouch sensor is also development in this chapter as an example. The IR sensor is suitable for applications requiring the detection of object which the pressure and seashell effect pretouch sensors may not be able to sense due to compliance and the acoustic absorptivity, i.e., towels or cloths.

Chapter 4

APPLICATIONS OF SEASHELL EFFECT PRETOUCH SENSING

Initial experiments performed with the proposed sensors explore two primary applications: (1) reactive grasp control—the advantages of pretouch over tactile sensing for sensing objects that are too compliant during the grasp execution; (2) pretouch-assisted grasp planning—the point cloud from the PR2's texture-projection / narrow stereo camera is augmented with additional points collected by recording the position of the robot's end effector when the pretouch sensor detects the object during a series of probing motions. This method compensates for the areas on the object that are missing from the point cloud either due to depth camera failure or occlusion.

4.1 Reactive Grasping of Compliant Objects

Tactile sensors have been widely researched for local sensing of objects by monitoring the contact force [27, 57, 61]. One of the cases in which this approach might fail is when the object is too compliant, so the pressure sensor can not detect it.

The first application of the seashell pretouch sensor is exploring the feasibility of sensing these extremely compliant objects during grasp execution. A pre-grasp execution experiment is performed, in which the pressure sensor built in to the PR2's gripper is compared to the seashell effect pretouch sensor for the ability to detect the objects. The two sensors are applied in a pre-grasp task to four different objects with different compliance. The objects used are a cookie box, a disposable cup, a folded paper box, and a folded aluminum foil box.

In the force sensing trial, the PR2's pressure sensor fingertips are installed on both right and left fingers on the PR2 right gripper. The standard PR2 gripper sensor controller is used to detect contact with objects. In order to increase the sensitivity of the pressure sensors, I lowered the contact detection threshold from the default values (0.1 N for the

Table 4.1: Results of the sensing for compliant objects experiments.

(∨: success; ×: failed)		
Objects	Pressure	Seashell pretouch
cookie box	∨	∨
disposable cup	∨	∨
folded paper box	×	∨
folded aluminum foil box	×	∨

high-pass filtered pressure readings, and 1.2 N for the unfiltered pressure) to the lowest values (0.05 N for the high-passed pressure readings, and 0.4 N for the unfiltered pressure) such that false positive detection is avoided.

In the pretouch trial, the seashell effect pretouch sensor is installed on one of the fingers on the PR2 right gripper. The gripper is commanded to close until the fingertip is sufficiently close to the object according to the seashell sensor.

A trial is defined as successful if the gripper stops before sqeezing too much that it breaks the object. The pressure sensor is able to detect the contact for cookie box and disposable cup, but it is not able to sense the contact with the extremely compliant folded paper and aluminum foil boxes, thus breaks them. On the other hand, the seashell effect pretouch sensor is able to sense all four objects and stop the gripper at an appropriate distance from the object, which shapes the pre-grasp pose (figure 4.1 and table 4.1). An interactive mode in which the gripper dynamically adjusts its opening based on the seashell effect pretouch sensor has also been implemented. This experiment shows the pretouch sensor can be a good complement to tactile sensing when grasping compliant objects.

4.2 Pretouch-Assisted Grasp Planning

3D perception is becoming more and more important in robotics, as well as in other fields. Point cloud data structures are widely used to represent the sensed spatial and color in-

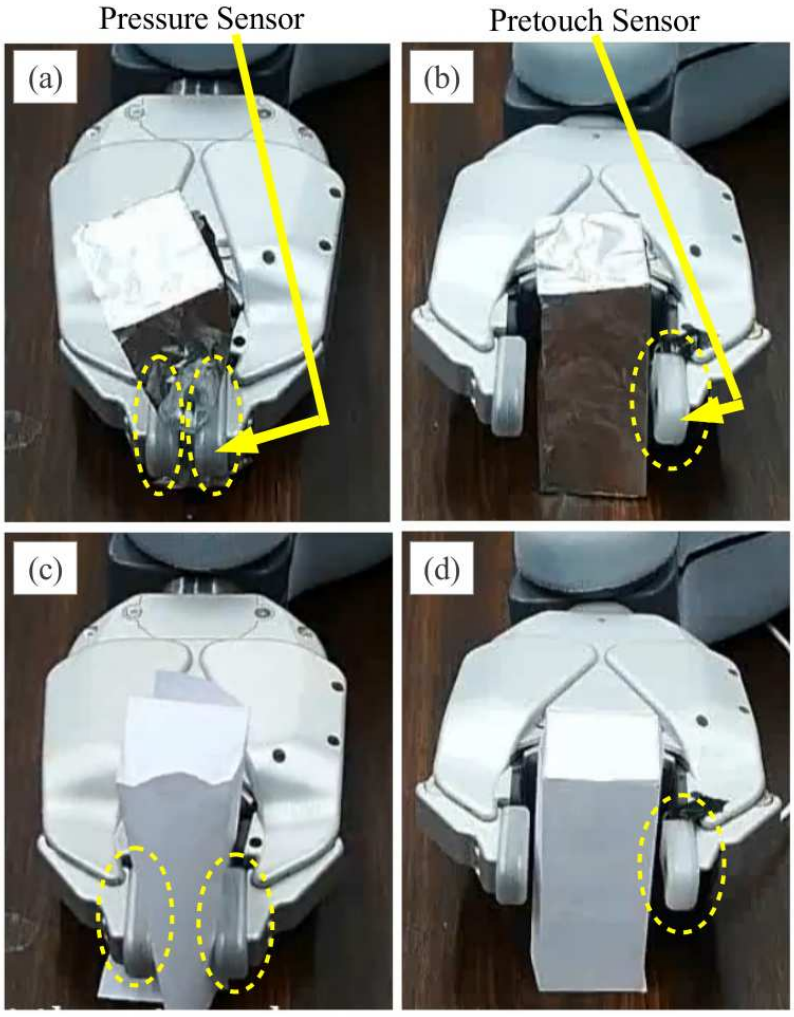


Figure 4.1: The results of pre-grasp execution experiments: (a) pregrasp a aluminum foil box with the pressure sensor; (b) pregrasp a aluminum foil box with the seashell effect pretouch sensor; (c) pregrasp a paper box with the pressure sensor; (d) pregrasp a paper box with the seashell effect pretouch sensor.

formation. A large community is developing a dedicated open source library, PCL (Point Cloud Library) [63], providing algorithms to process point clouds efficiently. In robotic grasping, an incomplete point cloud can result in poor grasp plans, simply because the planner is operating with incorrect information about the object geometry. Although depth sensing hardware has dramatically improved recently, it is still prone to dropping a many points, and there are still cases in which key object shape information cannot be collected by the depth camera because of geometric constraints; for example, in highly constrained environments, the robot may not be able to move its head enough to collect additional views.

I propose to use pretouch sensing in conjunction with the PR2's existing depth sensors, to provide data on portions of the object where information is missing, either because of occlusion or depth sensor failure. Given a partial point cloud collected by a depth sensor, the pretouch sensor (combined with robot kinematic data) will add additional points to the point cloud, prior to grasp planning. In this section, I will describe the methdology and how additional point cloud could be beneficial to robotic grasping. Some practical examples will also be discussed.

The grasping framework is built on top of the PR2 object manipulation framework proposed in [27]. I add two novel components to the framework: (1) a point cloud server to store both the point cloud generated from the depth sensor (stereo cameras in this case) and the pretouch point cloud I am generating later; and (2) a pretouch motion planner to command the robot end-effector to probe at the location of interest. Both components are implemented as ROS nodes.

The initial point cloud is generated by the depth data obtained from a single frame of the narrow-field-of-view stereo camera on the PR2. This point cloud is used to segment the objects from the table, and the point cloud cluster of the segmented object is then saved on the point cloud server. If pretouch point cloud is desired, the pretouch motion planner will use the point cloud to decide the initial probe location and then probe around the area.

The completed pipeline of the object grasping with pretouch point cloud is shown in figure 4.2. Two key components, pretouch motion planner and point cloud server, are added to the original PR2 grasping framework. The robot may choose whether to use the

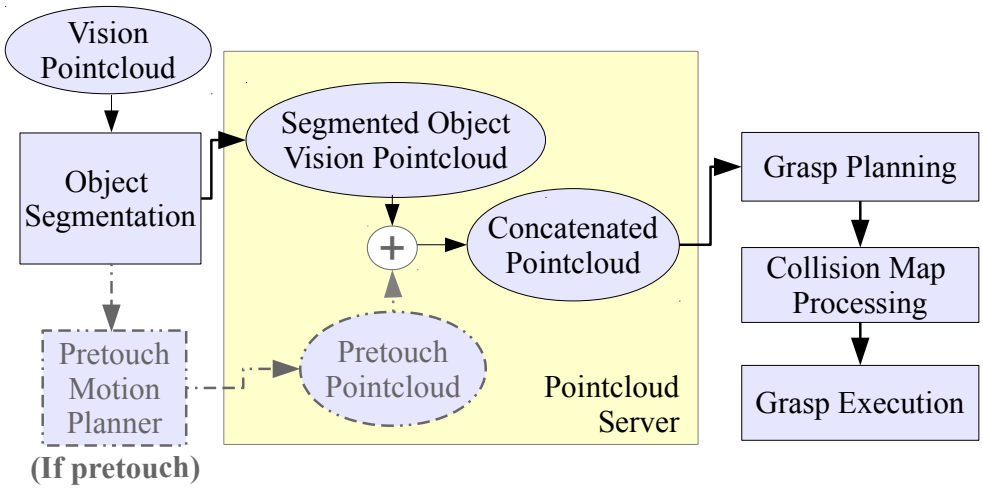


Figure 4.2: The pretouch-assisted grasping pipeline.

pretouch sensor or not. The augmented point cloud after the pretouch motion will be used for the final grasp planning, followed by the collision map processing and grasp execution. Each key step will be described in detail below.

4.2.1 Pretouch Motion Planning

The pretouch motion planner is a component that decides where the robot should move its end-effector to and look for the object surface. The current pretouch motion planning algorithm assumes the object is convex and fits an ellipic cylinder from the segmented object point cloud, and then the initial probe pose is determined by the farthest point the end-effector can reach in the unknown area of the object allowed by the PR2 arm's workspace. The algorithm works for most of objects with round and rectangular shapes. The fingertip always approaches in the direction normal to the periphery of the fitted ellipse with a constant velocity. When the pretouch sensor senses the distance closer than the threshold (when the estimated resonance frequency lower than the threshold 9500 Hz), W points are added on the point cloud server at the location translated by 3 mm in the X-axis from

the fingertip frame. The 3 mm offset is corresponding to the threshold 9500 Hz in sensor readings determined by the sensor characterization in figure 3.8. W is the weight of the point generated by the pretouch sensor. It determines how much the grasp planner will take these new points into account when doing grasp planning. Adding a single point at each location (W=1) will not alter the grasp plannng results much because the pretouch point cloud is sparse. The end-effector will continue lowering and probing with a fixed interval in the Z (vertical) axis before the gripper hitting the table. Once the height is too low before the robot arm hitting the table, the gripper moves back to the initial height and also moves backward to the direction of the robot body. The same process continues until the newly detected points are close enough to the existing point cloud generated from the camera.

In this particular implementation, the Point Cloud Library (PCL) [63] is used for the processing of point clouds, and a Jacobian inverse controller is used as a low level controller to control the end-effector pose and velocity in the cartesian space. Figure 4.3(a) shows an example of the PR2 probing a Coke bottle. Figure 4.3(b)(c) shows the virtual 3D point cloud representation of object after the probing. The red points are sensed from the robot's depth sensor (stereo camera), and the yellow points are generated by the pretouch sensor during the pretouch motion. The incomplete point cloud due to the transparency and reflection is filled by the pretouch sensor.

4.2.2 Grasp Planning and Execution

The grasp planner relies solely on the observed point cloud of an object versus fitting object models. The observed point cloud can be either with the pretouch points added or not. The algorithms presented in detail in [27]. This planner uses a set of heuristics, such as hand alignment with the cluster bounding box, size of the bounding box compared to gripper aperture, and number of observed points inside the gripper, to assess the quality of grasp, and also to populate a list of grasp candidates.

When the point cloud generated by the pretouch sensor is provided, it is concatenated with the point cloud from the cameras before being fed to the grasp planner. If pretouch point cloud is not provided, only the pre-stored point cloud from camera will be used to do

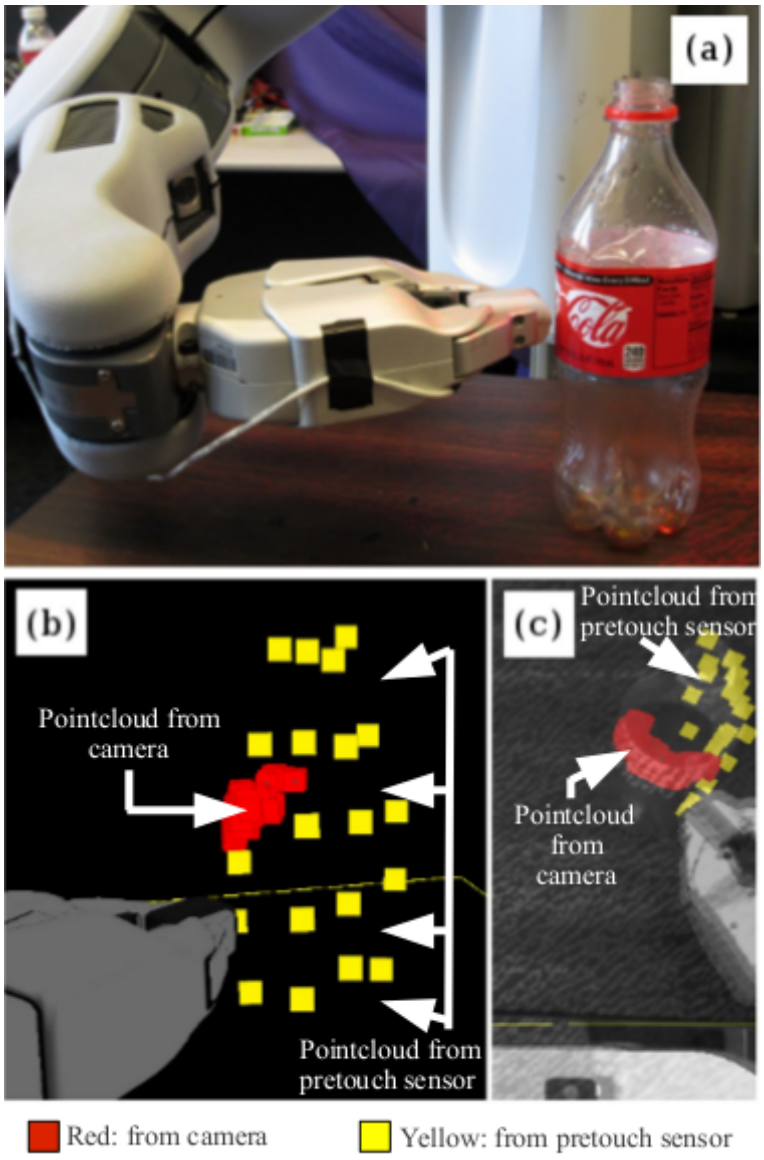


Figure 4.3: (a) PR2 robot probing the unseen area of the bottle and adding pretouch point cloud before grasp planning. (b) 3-D visualization in rviz. (c) camera image overlayed with the visualization markers.

the planning. The grasp candidate with the highest quality is used to execute the object grasping. After collision map processing, the robot will try to grasp the object in the planned pose.

4.2.3 Experiment and Results

In order to understand how the additional point cloud added by the pretouch sensor can affect the planned grasps, object grasping experiments are performed on 4 objects both with and without using the pretouch-assisted grasp planning approach. Figure 4.4 show the example grasping results on each of the objects for both cases. For each case, the 3D visualization in rviz, the image seen from the head-mounted narrow stereo camera overlayed with the visualization markers, and the planned grasp pose in the real world are shown. The red point cloud is generated by the stereo camera; the yellow point cloud is from the pretouch sensor; the green block is the fitted rectangular bounding box given the point cloud for grasp planning; and the gripper model shows the final planned grasp to be excuted.

(a) *Coke bottle:* Without pretouch sensor, the narrow-view stereo camera only captures the point cloud of the wrapper due to the transparency and the reflection of the light. Also the right and back sides of the red wrapper is missing because of self-occlusion. Since the grasp is planned only using the point cloud from the vision (red), the robot do not know the complete shape of the bottle, thus the planned grasp is off-centered. This particular grasp failed becuase the robot tries to grasp the bottle from the inclined portion of the bottle. After adding pretouch point cloud (yellow points), the fitted bounding box now matches the actual shape of the bottle better. The final grasp planned using the concatenated point cloud is centered and thus successful. This example shows the pretouch point cloud is useful for grasp planning when the object is self-occluded due to the view of camera.

(b) *Lego blocks:* The robot intends to grasp the yellow block on the table. In this configuration, ideally the a front-sided grasp should be more safe and reliable. However, there is another block in front of the target, which prevents any front grasp. When the robot is forced to do a side or overhead grasp, the lack of information on the backside of the object may result in the wrong decision. in this example, the robot decides to grasp

from the top and it locates the gripper without knowing the bumps in the back, thus hits the bump during the grasp execution. With the help of the pretouch point cloud, the robot fingers are located on the side walls in the overhead grasp, which avoids dealing with the bumpy backside. In this example, the depth sensor works properly, but due to the collision constraints, the additional information provided by the pretouch sensor plays an important roll in grasp planning.

(c) *milk box:* Due to the particular inclined shape and the view of the camera, the entire back side of the milk box is occluded. Similar to the case of the lego block, there are chances that the planned grasp will collide with the backside of the object without knowing there are things on the back. The added pretouch point cloud again helps plan a centered grasping pose in this case.

(d) *Snapple bottle:* The transparency and reflection of the plastic material cause the missing information on most of the portions except the wrapper on the bottle. Also the opening of the bottle is outside of the camera's view, so the initially planned overhead grasp is off-centered. The pretouch sensor adds additional useful information and shift the grasping pose to the center, which makes the grasp sucessful. This example demonstrates the pretouch point cloud is useful when the object is self-occluded and partially out of the view.

In figure 4.4(a) and figure 4.4(d), some portions of the incomplete point cloud in the transparent portions is filled by the pretouch sensor. In figure 4.4(b) and figure 4.4(d), some portions of the incomplete point cloud in the backside is filled by pretouch sensor. The experimental results demonstrate general applicability of the approach on some critical cases that the pretouch point cloud becomes useful to correct the grasp planning by reducing the uncertainty in object's geometry in the cases of occlusion, depth sensor failure, or when in constrained environment.

4.3 Summary

In this chapter, two applications are proposed and evaluated experimentally. The first application, reactive grasping, explores the possibility of detecting and locating extremely compliant objects during the grasp execution, which can not be done by using the traditional

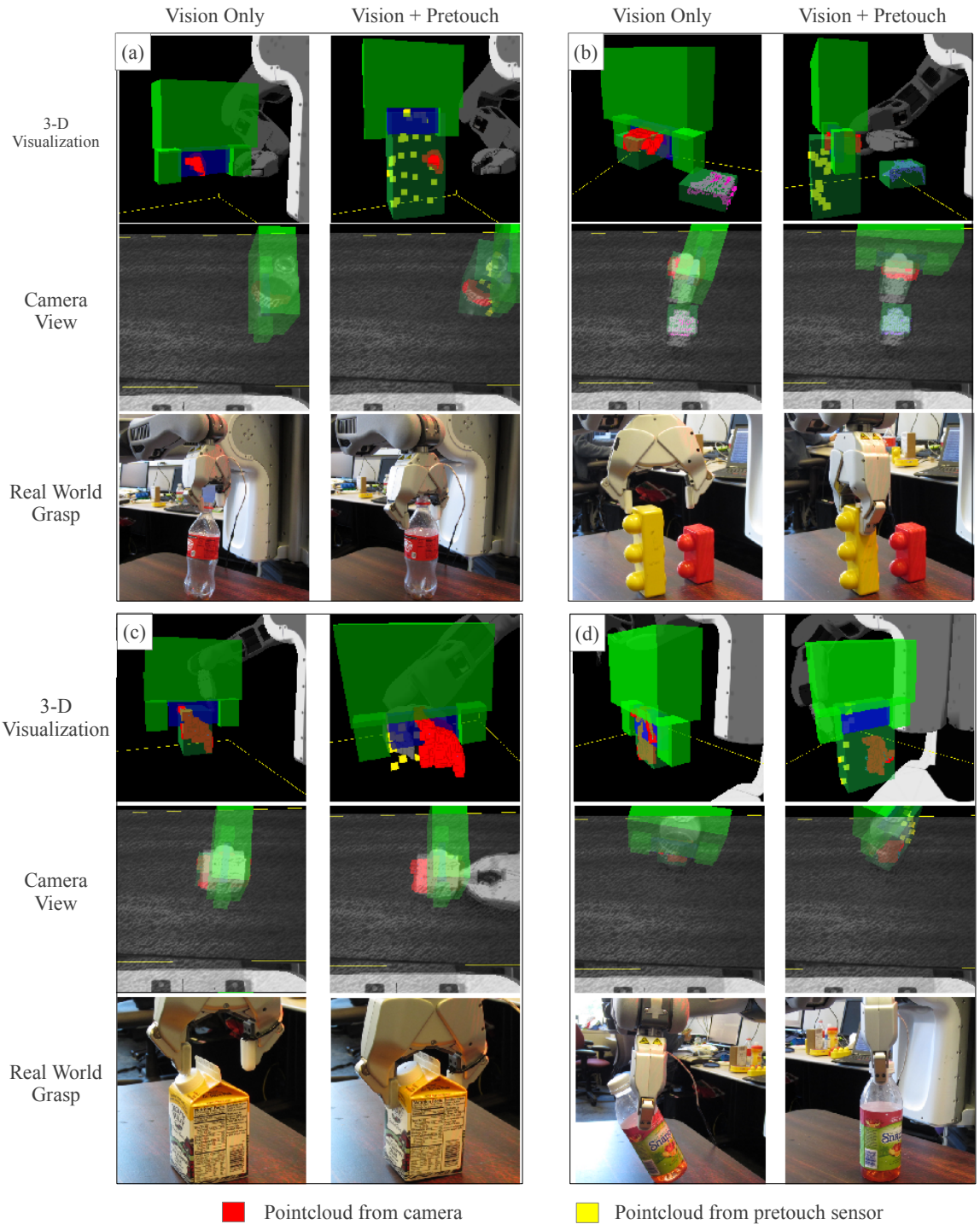


Figure 4.4: Examples of pretouch-assisted grasping. Compare the planned grasp with and without the additional point cloud generated using the pretouch sensor on 4 objects: (a) Coke bottle, (b) Lego blocks, (c) milk box, (d) Snapple bottle.

pressure sensors on the PR2 fingertip.

The second applicaion, pretouch-assisted grasp planning, shows the possibility of augmenting the incomplete point cloud. The pretouch sensing added points to a point cloud collected by a depth camera, and enable improved grasp plans. Seashell effect pretouch can be used to collect spatial information in regions where optical depth sensing has failed. Unlike existing pretouch modalities, seashell effect pretouch complements optical depth sensing because they rely on fundamentally different physical mechanisms: mechanical/acoustic in the case of seashell effect pretouch, and electromagnetic/optical in the case of depth cameras, electric field pretouch, and optical pretouch. In the next chapter, the application of pretouch-assisted grasp planning will be further extended to a more effective generalized framework.

Chapter 5

A UNIFIED FRAMEWORK FOR SHAPE ACQUISITION AND GRASPING

In order to have the robot utilize pretouch more effectively for grasping unknown objects, a unified probabilistic framework is proposed with the goal to (1) locate shape uncertainty for the target object, and (2) enable automatic exploration with a pretouch sensor to reduce the uncertainty toward a grasp with high confidence. The framework is an extension of the pretouch motion planning application decribed in section 4.2 with improvements in many aspects. This chapter starts from reviewing the related work about robotic grasping and planning under uncertainty. The framework and the algorithm are then described in detail.

5.1 Related Work

A common way to deal with grasping under uncertainty is to take a sequence of actions (sensing, pushing, etc) which reduces the uncertainty about the object. Planning an optimal sequence of these actions can be modeled as a partially observable Markov decision process (POMDP) problem, which is known to be PSPACE complete [62]. Therefore, POMDP-based action planning is usually used on abstracted state and action spaces [30].

Information-theoretic optimal action selection is widely applied to this set of problems. Hebert et al. [24] use Next-Best-Touch to estimate the pose of object. The Next-Best-View in-hand Kinect sensing [60] has the drawback of the difficulty of integration to a manipulator. Dogar et al. [19] demonstrated a push-grasp algorithm that addresses the uncertainty about object poses in clutter. Recently Platt et al. [58] proposed to simultaneously localize and grasp known objects under uncertainty by creating plans in belief-space, the space of probability distributions over the underlying state of the system. The algorithm estimates the state of the system and attempts to reach a goal region with high probability. Javdani et al. [32] proposed a method to select a sequence of information gathering actions to localize and grasp known objects by considering many hypotheses of object pose and greedily select

actions expected to disprove the most hypotheses. This formulation is shown to be adaptive submodular.

In contrast to the above work, this framework aims to grasp unknown or novel objects without any prior training. For known objects (ones where a model of object shape is available), the problem can be approximately reduced to a pose and state estimation problem. In this framework, the object's shape is not available to the robot in advance, and pretouch sensing collects additional information about object shape, rather than being used to estimate object pose. Dang et al. [16] utilized tactile feedback to predict the stability of a robotic grasp without knowing the object's model, but the action after identifying instability is not mentioned in their work. Our framework combines grasp planning and shape sensing natually: the grasp planner automatically determines when and where to do the pretouch sensing exploration, and the information gathered during exploration will be considered during the next iteration of the grasp planning, i.e. the shape uncertainty is reduced in an attempt to a better grasp.

5.2 Framework and Environment Representation

The framework consists of two main subsystems. The robot first locates and infers the incomplete object shape from Kinect data using some general priors under a probabilistic occupancy map representation. The constructed occupancy probability distribution is later used by the iterative grasp replanning algorithm to decide when further exploration is required, and where to explore in the scene. The robot executes the grasp when the algorithm terminates. The flowchart of the framework is shown in figure 5.1.

To let the robot plan under uncertainties, an probabilistic environment representation is required. Hollinger et al. [26] used Gaussian processes implicit surfaces to represent the uncertainty and fomulate the avtive sensing problem to a problem of minimizing the average variance. However, it can not represent arbitrary shapes. I use OctoMap [82], a compact and efficient probabilistic 3-D grid environment representation which stores the occupancy probability at each grid cell, to represent the occupancy probability (uncertainty) around the tabletop and target objects. OctoMap has become the standard the standard collision map environment for online motion planning. The occupancy probability at each grid can

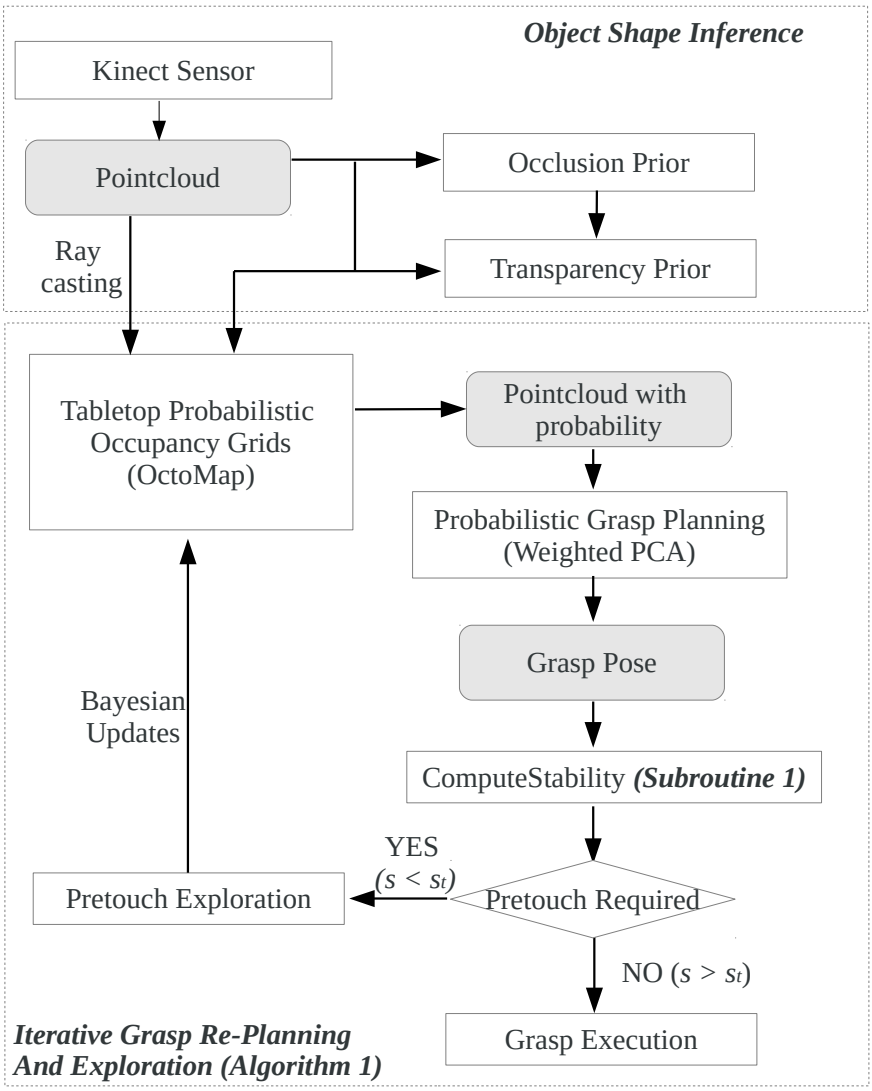


Figure 5.1: The flowchart of the probabilistic framework for pretouch exploration and grasping under uncertainty

be updated easily by Bayesian updates with sensor models. Furthermore, it provides easy conversion between occupancy grids and point clouds, which makes it easy to use various algorithms in the Point Cloud Library (PCL) [63].

5.3 Inferring Object Shape from Kinect Data

Glover et al. [20] proposed to learn probablistic generative models of a class of objects from 2-D image, and the learned models can be used to infer the hidden parts to complete the object contour before manipulation. I take a different approach by using priors inspired from the sensor characteristics to infer the object shape in 3-D for the purpose of further pretouch sensing and grasping. No model training is requred in this framework. I first construct an initial occupancy distribution from Kinect data, and then infer the object shape using prior probability distributions for occlusion and transparency in the 3-D space. Specifically, all occupancy grid cells are initialized with the uniform distribution ($p = 0.5$), and are updated using ray-casting from the Kinect point cloud to build the initial occupancy distribution. To make the robot aware of the uncertainty of the object shape, I use occlusion and transparency priors to extend the occupied grid cells detected by Kinect, inferring potentially occluded and transparent parts of the object.

5.3.1 Ray casting from the Kinect Point Cloud

A beam-based inverse sensor model of the Kinect is used to update the occupancy grid. The underlying ray-casting algorithm is a 3D variant of the Bresenham algorithm [2]. The occupancy of the grid cells along the ray are updated by Bayesian updates using the Kinect's beam-based inverse sensor model ($p_{kinect,hit} = 0.7$ and $p_{kinect,miss} = 0.4$).

5.3.2 Occlusion Prior

Occlusion is a common problem in object grasping. Often it causes underestimates of the object depth. To deal with the occlusion problem, the robot should have doubt about the backside of the object. Based on the physics of the depth sensor, grid cells along the ray behind each point sensed by the Kinect are unknown and possibly occupied. In the the

most general case, occluded portions of an object could appear anywhere from the detected point to the maximum range of the sensor. In the tabletop environment, I can truncate the distribution at the table surface. For the occupancy prior (along the ray from a sensed point to the table top), I use an exponential distribution. This corresponds to the simple assumption that for a grid cell i and its neighbor along the ray $i-1$, the conditional probability of occupancy $p(i|i-1)$ is a constant. Another interpretation is that the exponential distribution models the distribution of the distance from the actually detected point to the first non-occupied grid cell along the ray given the conditional independence mentioned above. The inferred occupancy probability along the ray is given by:

$$p = p_0 \times e^{-\lambda_{occ} x} \tag{5.1}$$

where p_0 is the occupied probability on the actually detected point, x is the distance in meters away from the detected point, and λ is the exponential decay rate. In this work, $\lambda_{occ} = 4.0$ is selected experimentally to fit with the realistic conditions for the objects on the tabletop. For example, if $p_{kinect,hit}$ are 1.0, then the occupancy probability would have decayed to 0.5 at a distance of 17 cm. Essentially the decay rate in the occupancy prior encodes the robot's initial assumptions about the size of the object.

5.3.3 Transparency Prior

Transparency and non-Lambertian objects are another common error source for depth sensors. One important observation when inspecting the Kinect data for transparent objects is that although the transparent parts are not detected by the sensor, the transparent parts still block the ray so nothing can be detected behind the transparent cell along the ray. Based on this observation, I search for unknown areas in the tabletop scene, where points should be observed but nothing is detected. If a ray is casted from the location of Kinect sensor to the pixels having no sensor data, there are likely to be transparent parts somewhere along the ray.

There are several possible ways to look for these suspicious areas. In the tabletop environment, I do point cloud processing to find the holes (shadows) on the table, and then cast rays to those shadow points from the Kinect sensor frame. I also cast rays upward and

downward from occupied grid cells, and if any of the rays intersects with the rays cast from the sensor to shadow points, the intersecting grids are potential transparent cells. I made a similar assumption that the likelihood of finding transparent grids decays exponentially with distance away from occupied grid cells. The occupancy probability associated with transparent grid cells is thus defined as:

$$p = p_0 \times e^{-\lambda_{trans} x} \tag{5.2}$$

where p_0 is the occupancy probability on an occupied grid cell, x is the distance from the grid cell, and λ is the exponential decay rate. In this work, $\lambda_{trans} = 4.0$ is selected experimentally to fit with the realistic conditions for the objects on the tabletop.

5.3.4 Object Shape Inference Results

Figure 5.2(a)(b) shows the environment representation before and after the object shape inference using the above uncertainty priors. The original point cloud detected by the head-mounted Kinect (figure 5.2(a)) is missing the transparent portion of the bottle, and also underestimating the depth of the object. During the inference, the transparency and occlusion priors are used to extend the initial occupancy data obtained from Kinect. Figure 5.2(b) shows the visualization of the resulting constructed probabilistic occupancy map after inference. The occupancy distribution is extended toward the potential occluded and transparent portions of the object. In this particular example, the transparency prior matches the the actual object shape even at the nozzle of the bottle. I can also see that grid cells are less likely to be occupied the further they are from the Kinect point cloud.

5.4 Iterative Grasp Replanning and Pretouch Exploration

After constructing the occupancy map by inferring the object from Kinect data, an iterative grasp replanning and exploration algorithm is used to decide when further exploration is required, and where to explore in the scene using a pretouch sensor. The information gathered by the exploration action is added directly to the robot's environment representation and is considered automatically in the next grasp replanning iteration. The algorithm will terminate and execute the grasp when it finds a grasp with high stability for both the left

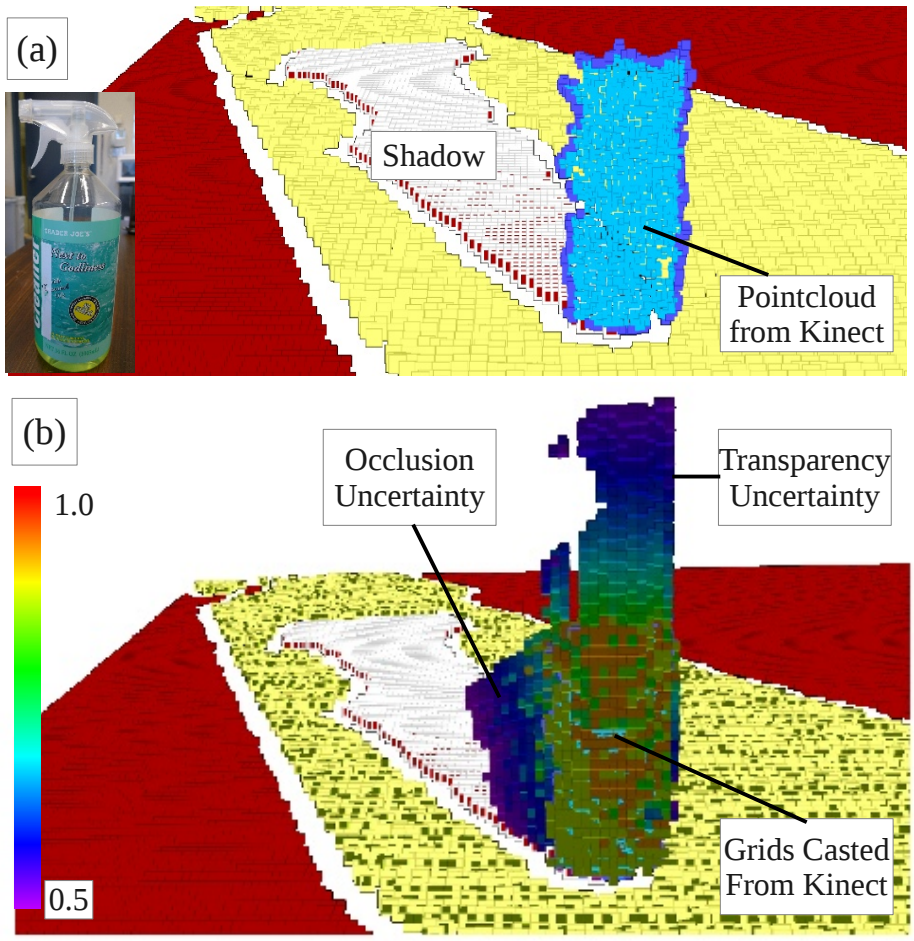


Figure 5.2: (a) The incomplete point cloud detected by Kinect. (b) The probabilistic map representation after the tabletop uncertainty processing. (The color bar represents occupancy probability between 0.5-1.0 in the log-odd space.)

finger and the right finger. Figure 5.1 outlines the schematic and flow of the framework. In the following subsections, I will introduce the individual pieces required for the algorithm.

5.4.1 Probabilistic Grasp Planning

The grasp planner plans grasps based on the occupied grids inferred from the objects in the probabilistic occupancy map. It is a variant of the cluster-based grasp planner for unknown objects described in [27]. The original planer first finds a oriented bounding box of the object using principal component analysis (PCA) from the object's point cloud, and then finds grasps aligning the PR2's parallel gripper with the principal axes. Our modified version of this grasp planner takes the occupancy probability of each point in the object point cloud into account. The probabilistic occupancy map is first converted into a probabilistic point cloud with probability associated with each point for the grids with occupancy probability $p > 0.5$. The grasp planner then finds the object bounding box by computing the weighted PCA according to the probabilistic point cloud. Figure 5.3(b) shows the different object bounding boxes computed using the regular PCA and weighted PCA, and on the original point cloud from Kinect, and the probabilistic point cloud converted from the occupancy map. The PCA bounding box computed from the original point cloud (red) does not cover the transparent parts of the object, and the dimension in the depth direction is too short due to the occlusion. In contrast, the weighted PCA bounding box computed from the probabilistic point cloud (green) takes the occupancy probability in the potentially transparent and occluded portion into account, thus provides the grasp planner the opportunity to plan a grasp around those areas.

5.4.2 Grasp Stability Evaluation

To decide whether the planned grasp is sufficiently certain to be excuted or not, the grasp stability in the sense of object shape uncertainty is evaluated.

First, the potential contact areas on the object is defined by finding the k-nearest neighbor points to the left and right robot fingertips of the given grasp in the probabilistic occupancy map. The pink markers in figure 5.3(c) and figure 5.3(d) represent the locations

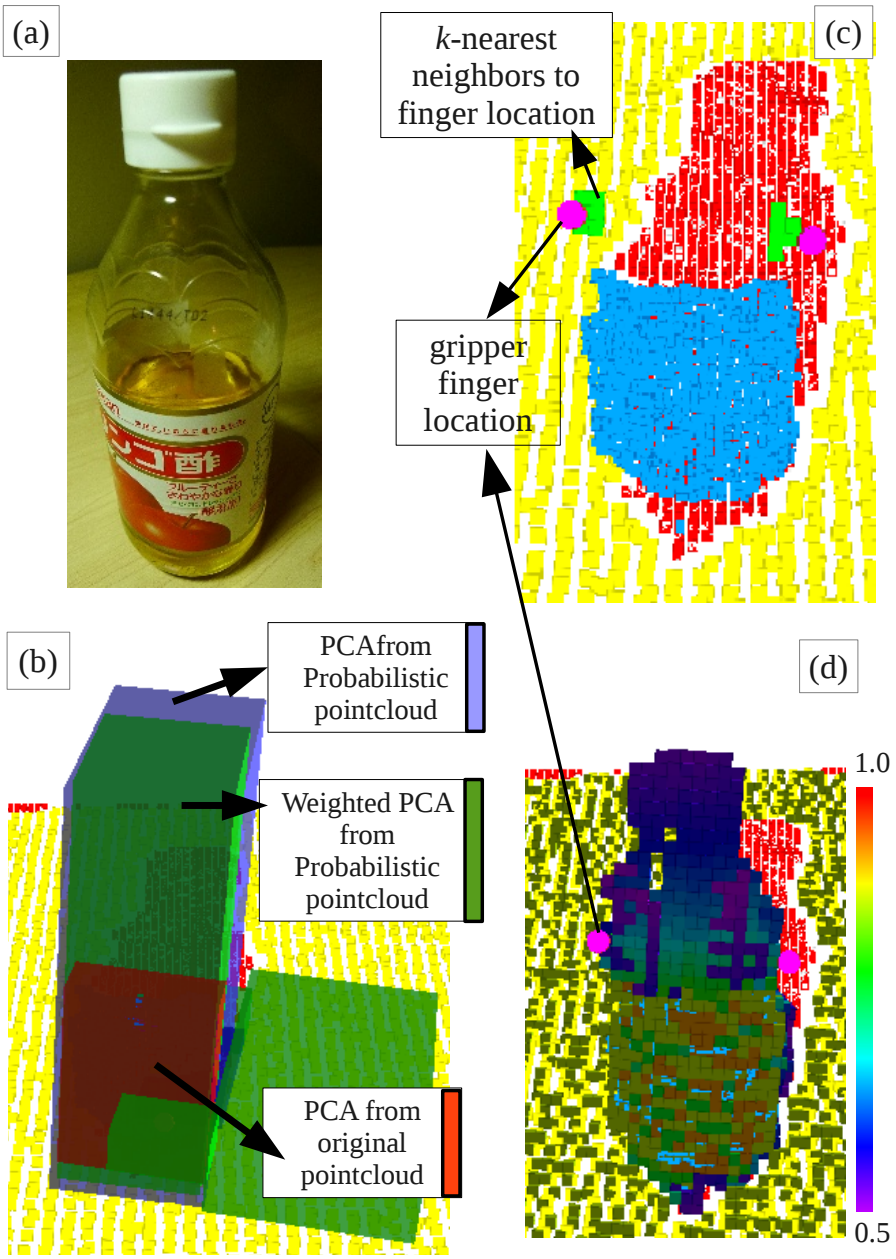


Figure 5.3: An example of the grasp planning and grasp stability evaluation: (a) the target object; (b) bounding boxes computed during the grasp; (c) the object point cloud overlayed with the finger locations of the planned grasp and the potential contact areas (k-nearest points); (d) the probabilistic occupancy map overlayed with the finger locations of the planned grasp.

of the left and right robot fingertips, and the green markers in figure 5.3(c) represent the potential contact areas on the object for the given grasp represted by the k closest points for each side.

Secondly, I define s_l and s_r as the stability for a given grasp at the left finger and the right finger, respectively, to be the mean occupancy probability:

$$s_m = (\sum_{i=0}^{k-1} p_{m,i})/k \tag{5.3}$$

for $m = l$ or r, where $p_{m,i}$ are the occupancy probabilities of the points for $i = [0, k-1]$, in the probabilistic point cloud to the left and right fingertip surface centroid.

Finally, a stability thresold s_t is defined. If either s_l or s_r is smaller than the thresold s_t, it indicates the planned grasp is attempting to make contact with the object from an uncertain area. The robot would need to explore that area using the pretouch sensor to reduce the uncertainty instead of excuting the grasp. The subroutine to compute the stability of a planned grasp is described in Subroutine 1.

5.4.3 Pretouch Exploration

When the planned grasp is not certain, pretouch exploration is required to reduce the uncertainty (or, increase the grasp stability by the definition in section 5.4.2) before grasping. It is desired that the robot can automatically decide the uncertain areas to sense which is crucial for the grasping task. Inspired by the fact that the final goal of the entire task is to grasp the object with confidence, it is reasonable sense the uncertain contact area of the previous planned grasp for the purpose of reducing uncertainty for grasping. Using this concept, the desired pose for the robot to probe using the tip-mounted pretouch sensor is be computed from the previous planned grasp.

The pose is pointing to the gripper closing direction of the planned grasp, and pointing at the centroid of the k-nearest neighbor points. During the pretouch probe execution, the robot moves the sensor to the pretouch pose, and moves the end effector toward the potential contact area at constant speed. Figure 5.4 illustrates an example pretouch probe configuration. While probing, the robot continuously uses the pretouch sensor's inverse

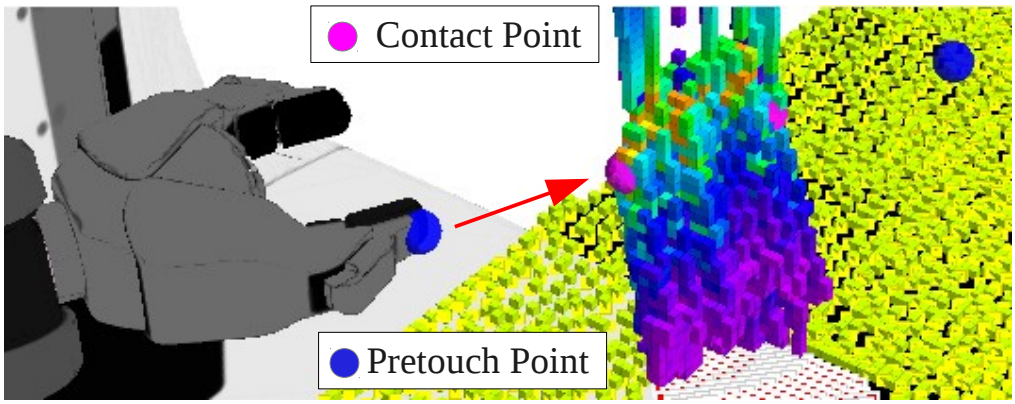


Figure 5.4: An example pretouch pose and direction. The robot uses the seashell effect pretouch sensor to probe toward the potential contact area on the object.

sensor model to update occupancy probability of the grids within the sensing range of 4 mm ($p_{pretouch,hit} = 0.95$, $p_{pretouch,miss} = 0.48$). Figure 5.5 shows an example of how a single probe succesfullty updates the occupancy when it detects the object.

Subroutine 1 `ComputeStability` $(m, Grasp)$

Input: m: the side of the finger (l or r)

Input: $Grasp$: a planned grasp from the grasp planner

Output: s_m: the contact stability at finger m

1: Find k-nearest neighbors in the probabilistic point cloud to finger m on $Grasp$

2: **for** $i \leftarrow 0$ to $k-1$ **do**

3: $\quad p_{m,i} \leftarrow$ occupancy probability of point i

4: **end for**

5: $s_m \leftarrow (\sum_{i=0}^{k-1} p_{m,i})/k$

6: **return** s_m

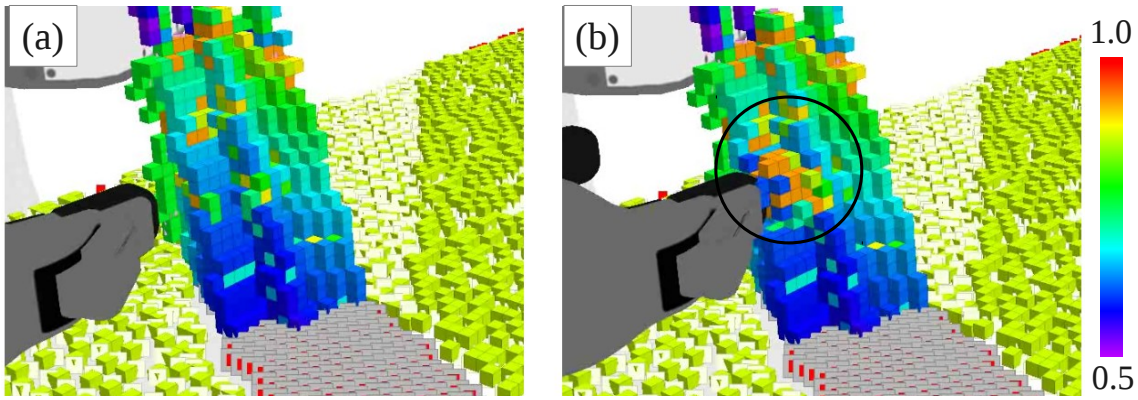


Figure 5.5: (a) the probabilistic occupancy map before the pretouch sensor detects the object; (b) the probabilistic occupancy map after the pretouch sensor detects the object and updates the occupancy probability around using its sensor model. The color map represents occupancy probability between 0.5-1.0 in the log-odd space.

5.4.4 Iterative Grasp Replanning and Pretouch Exploration

The complete procedure is described in Algorithm 1. The robot first uses the weighted PCA cluster-based planner to find a grasp candidate. The stability of the planned grasp is computed using Subroutine 1. For each side of the gripper's ufinger, If the stability is lower than the stability threshold s_t, the robot uses pretouch sensor to explore the potential contact areas, and then replan a grasp based on the updated occupancy grids. The algorithm terminates and execute the planned grasp when a stable grasp is found.

5.5 Experimental Results

I evaluate the proposed framework on the PR2 robot using the seashell effect pretouch sensor decribed in chapter 3 for object grasping. The framework's parameters used in this set of experiments are: $s_t = 0.7$, $p_{kinect,hit} = 0.7$, $p_{kinect,miss} = 0.4$, $p_{pretouch,hit} = 0.95$, $p_{pretouch,miss} = 0.48$, $\lambda_{trans} = 4.0$, $\lambda_{occ} = 4.0$, and $k = 40$ for the grasp stability threshold, kinect's inverse sensor model, pretouch sensor's inverse model, the occlusion and transparency prior's decay rates, and the number of neighbor points to evaluate the contact stability. The grasp stability s_t is determined by testing with the combination of other parameters. By using a higher threshold, the framework is more robust, but this also

Algorithm 1 Iterative Grasp Replanning and Exploration

```
1: s_t ← stability threshold
2: StableGraspFound ← False
3: while not StableGraspFound do
4:     Grasp ← probabilistic grasp planning results
5:     for m = l, r do
6:         s_m ← ComputeStability(m, Grasp)
7:         if s_m < s_t then
8:             execute pretouch probe to the contact area
9:         end if
10:    end for
11:    if no pretouch is excuted then
12:        StableGraspFound ← True
13:    end if
14: end while
15: execute Grasp
```

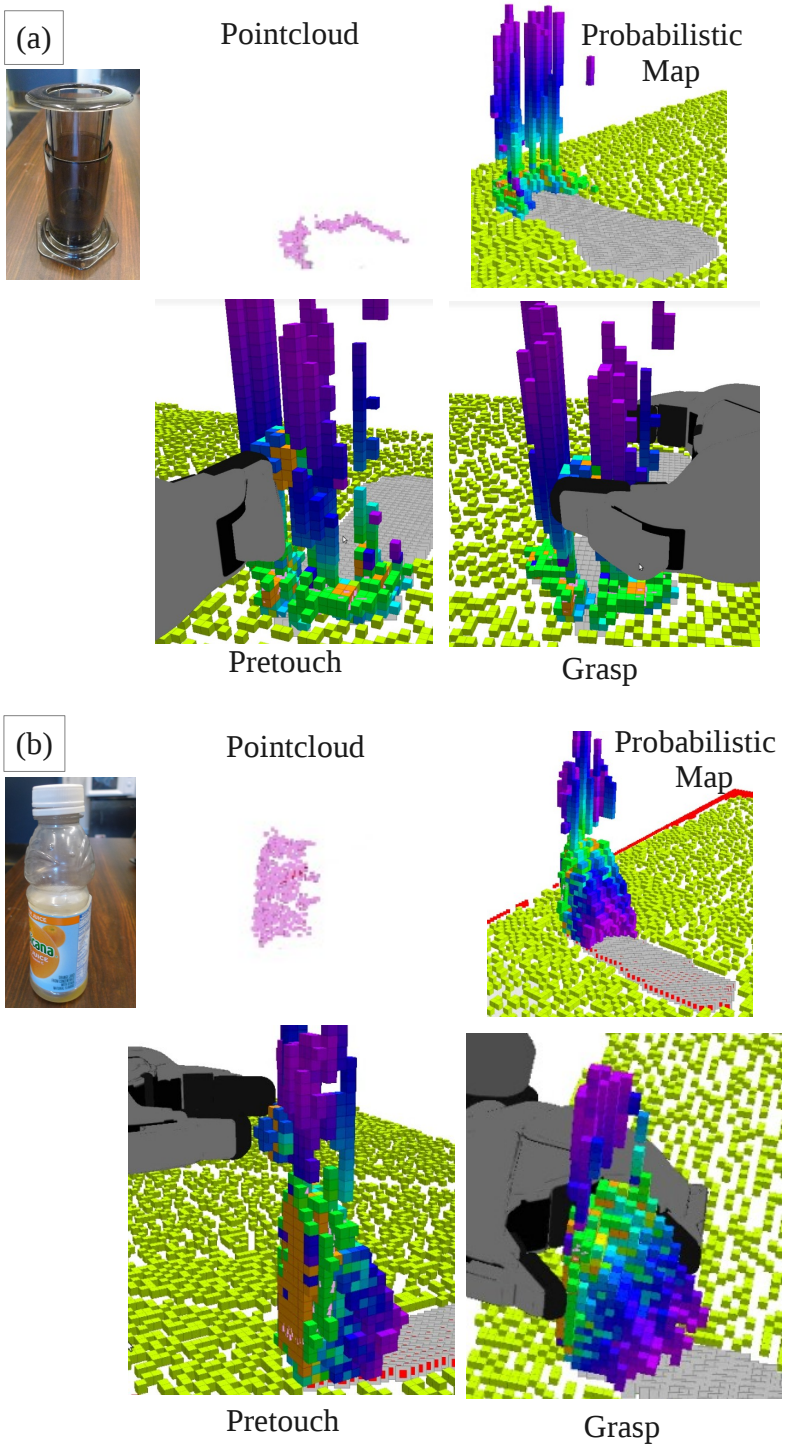


Figure 5.6: A set of example system states during the progress of grasping using the proposed framework for the two objects (a) a coffee press; (b) a partially transparent juice bottle.

makes the algorithm slower to converge. The OctoMap resolution I use for the tabletop environment is 5 mm.

Two objects, a coffee press and a partially transparent bottle, are tested multiple times. Figure 5.6 shows a set of example system states during the progress of grasping for the two objects. The coffee press is almost completely transparent except right at the bottom. Therefore, the point cloud from Kinect is only at the bottom of the object as in figure 5.6(a). Most of the grasp planners would fail in this case. Using this framework, the initial probabilistic map after inferring the object shape from the Kinect data successfully extends the occupancy prior from the bottom to the upper transparent portion. The grasp planner can thus plan a grasp based on this prior occupancy distribution as I introduced in section 5.4.1. However, since the extended occupancy probability is lower than the occupancy probability at actual sensed grids, the planned grasp is often required to contact with some hypothesized portion of the object. In consequence, algorithm 1 determines to probe those areas and thus reduces the uncertainty in the system at each iteration. In the example shown in 5.6(a), the pretouch probe sucessfully increases (verifies) the occupancy probability of grids near the grasping contact area, and the followed planned grasp is planned and execute at the verified location.

The partially transparent bottle, on the other hand, requires much less pretouch sensing before grasp execution. The point cloud from Kinect for the bottle is usually sufficient to plan a stable grasp, because the grasp planner plans the grasp in favor of the area with higher occupancy probability using weighted PCA. In contrast, the grasp planner is forced to plan grasps on uncertain areas for the coffee press because there are very few grids with high occupancy probability.The extended occupancy prior also makes the grasp planner tend to favor front grasps in terms of the Kinect's coordinate frame, which could significantly reduces the errors arising from underestimating the object depth while doing side grasps. Figure 5.6(b) shows an example of succesful front grasp.

Table 5.1 summarizes the results of the grasping experiments on the two objects. The grasping tasks are mostly successful except one case of inverse kinematics failure. Not surprising, the averaged number of the pretouch probes before grasping is much higher when grasping the transparent coffee press (1.54) than when grasping the partially transparent

Table 5.1: Results of grasping trials using the proposed framework

	Averaged number of pretouch probes before grasping	Trials	Success Grasp	Success Rate
Coffee Press	1.54	12	12	100%
Bottle	0.40	10	9	90%

bottle (0.40). Nonetheless, the iterative grasp-replanning and exploration algorithm terminates within 5 pretouch probes because of the decreasing shape uncertainty at each of the iterations. After the iterative algorithm terminated, the robot can grasp the object with high confidence, which results in high success rates.

5.6 Summary

In this chapter, I proposed a framework which enables automatic exploration with a pretouch sensor to reduce the uncertainty toward a grasp with high confidence. The robot first makes inference based on initially available information to identify potential uncertain areas of the target object. An iterative grasp-replanning algorithm is then operating on the prior occupancy distribution to actively explore those areas. The algorithm starts from planning a grasp for the target object given the shape represented in the inferred probabilistic occupancy map. To decide whether the planned grasp is sufficiently certain to be excuted or not, the grasp stability in the sense of object shape uncertainty is evaluated. When a planned grasp is not certain, pretouch exploration is preformed to reduce the uncertainty. The process iterates until a grasp with high confidence is found.

There are several key advantages of this framework:

- The sensing is performed with the goal of finding a high confidence grasp as opposed to other previous methods in where exhausted exploration was performed to complete the object model. It can significantly speed up the exploration process.

- The uncertainty of the object shape is reduced after each iteration of the pretouch exploration, and the information is updated in the probabilistic occupancy map in real-time. The next grasp planning is based on the more defined object shape, and thus are more robust.

- The exploration is via pretouch, so the object is not disturbed (moved). In previous related work of tactile exploration, iterative closest point (ICP) algorithm is usually required to register the objects in the scene before and after each of the touch sensing due to the movement of the objects;

- The robot only executes the grasp when a grasp with high confidence is found.

The experimental results show that, by using pretouch exploration with the seashell effect pretouch sensor under this framework, the PR2 robot is able to grasp partially transparent objects with high success rates (96%), which would not have been possible with previous methods. For objects with less shape uncertainty, the point cloud detected by Kinect is usually sufficient to plan a stable grasp, as the grasp planner tend to use areas with higher occupancy probabilities due to the weighted PCA model. In contrast, in the extreme cases that very few shape information is available, the grasp planner is forced to plan grasps on uncertain areas identified from the inference process, and thus requires more iterations of pretouch exporation.

Chapter 6

FUSING PRETOUCH SENSING AND HAPTICS FOR TELEROBOTIC GRASPING

Robotic teleoperation is the task where the robot is controlled directly and remotely by a human operator. Among different teleoperation tasks, the telemanipulation task interacting with target objects is considered as an important and extremely difficult task due to the limited sensor information provided to the operator. In such tasks, it is important to gather sufficient geometrical information about the target object. Additionally, in many scenarios, it is of interest to gather these information without physical contact with the object.

In this chapter, I present a method of augmenting streaming point cloud data with pretouch proximity sensor information for the purposes of haptic rendering in a telerobotic grasping task. The non-contact seashell-effect pretouch sensor fixed to the robot end effector introduced in chapter 3 is used to sense physical geometries within the vicinity of the device. Thus, the point cloud representation of an unknown environment, which may be sparse or poorly visible, can be enhanced through telerobotic exploration/sensing in real-time. Furthurmore, real-time haptic rendering algorithms are applied on the augmented point clouds to create haptic virtual fixtures, and also provide haptic force feedback to the operator. This method provides the teleoperator with critical geometrical information about the grasp target while simultaneously preventing the robot end effector from colliding with it.

6.1 Background

6.1.1 Teleoperation

In remote robotics, oftentimes autonomy is not sufficient to complete challenging tasks in unstructured environments. One of the examples is grasping unknown or poorly perceived objects in clutter without the complete geometric information of the objects. For the case

of automation or semi-automation, heuristic grasp planning algorithms can be implemented to handle the uncertainties of the object [27], but do not work well in clutter. In such cases, teleoperation which takes human into the loop has the potential to increase the success rate by combining the cognitive skills of the human operator with the robotic autonomous behaviors [42].

Teleoperation refers to controlling a robotic device/manipulator, the *slave*, by a human operator provided a control input *master* device (e.g., a joystick or a smaller manipulator) *from a distance* [25]. The operator is usually provided visual or audio feedback based on the sensors equipped on the slave robot in order to know the remote environment where the actual task is performed. A broad set of research topics related to teleoperation have been performed including the appropriate control theories [5, 54, 68], the effect of latency [3, 72], and human-machine interaction/collaboration [21, 79].

6.1.2 Haptic Feedback for Teleoperation

One problem raised in the teleoperation setup is the operator's lack of perception of the remote workspace. Since the operator generates the control input solely based on the virtual environment, the completeness of the data about the states of the remote environment is crucial for proper task execution. More complete data can improve the performace and the achievement of the tasks.

In recent years, inexpensive RGB-D cameras have provided non-contact means of collecting geometrical information of an observed scene. However, the visual information collected from the remote sensors still suffers from incomplete or poorly visible geometries of the observed scene due the limited field of view, occlusion or sensor limitation.

Kinesthetic feedback, such as haptic or tactile feedbacks, have thus been proposed to increase a operator's perception of the remote workspace to stablize the human-machine interaction loop in various teleoperation tasks [17, 76]. For example, if the slave possess force sensors, it can transmit or reflect back to the master reaction forces from the task being performed, which enters into the input torque of the master, and the teleoperator is said to be controlled *bilaterally* [25]. Analytical method was proposed by Hannaford

[23] to provide useful characterizations of kinesthetic feedback in the bilateral teleoperation scenario. The haptic feedback has been shown to beneficial to various teleoperation tasks. Lee et al. [41] showed a method of converting the obstacle range information measured from an onboard sensor into force reflected to the operator specifically for a teleoperated navigation task on a mobile robot. The addition of haptic feedback reduces the collisions and increases minimum distance between the robot and obstables during a navigation task. Okamura [55] showed that by providing haptic feedback for the teleoperated surgical robot manipulator, the surgens can accomplish some tasks which were previously not possible. The master operator can also use haptic feedback to provide intelligent assistants, generating virtual fixtures that support various manipulation tasks performed by the surgeon. However, the study is only performed on teleoperated surgical robots as opposed to general robotic manipulators. Tsetserukou et al. [78] showed that haptic feedback provides safe physical interaction with environment and enables high level of maneuverability of robot arm in unstructured environment. The slave robot is covered by tactile sensors for detecting contact point, and is also equipped with torque sensors distributed into each joint for measurement of applied force. One drawback of this approach is that the haptic feedback relies on physical contacts, and thus undesired collision can not be avoided. In cases of involving high-value or delicate physical structures (for example bomb defusing or surgery), it is imperative that the operator is able to make the robot avoids unintended contact with the object based on non-contact sensor data.

6.1.3 Haptic Rendering with Non-contact Sensors

Ryden et al. [66, 67] proposed real-time haptic rendering methods for streaming point clouds from a RGB-D depth sensor (Kinect). A proxy tracking method is introduced used to control the motion of the slave's end effector, in which a user's position and orientation inputs to the master haptic device correspond to scaled position and orientation of a haptic interaction point (HIP) within a virtual point cloud representation of the environment. Kinesthetic haptic feedback is rendered by a virtual coupling methods for proxy-HIP movement algorithms. Furthermore, in [64], forbidden-region virtual fixtures were used to

enforce a minimum distance between the robot end effector and sensed object surfaces.

However, real-time streaming point cloud information from current RGB-D cameras is obtained at the cost of sparse scenes littered with 'holes' occurring from occlusions, optical properties, glancing incidence angles and other factors. This is undesirable for applications in which accurate and dense 3D information retrieval is essential for the task at hand, such as grasping. In teleoperated robotic graspig, the occlusion problem is even more prominent because of the slave robot manipulator blocking between the head-mounted depth sensor and the target object. For instance, in [43], a method for constraint-aware-teleoperation is proposed whereby the user provides an end pose goal and the algorithm generates a motion path under the collision constraints. However, the collision constraints are captured from a Kinect camera and is thus prone to occlusion when teleoperating the manipulator around. Local features, particularly in close proximity of the target object, are thus desirable from other sources of sensing techniques.

6.1.4 Augmenting Point Clouds

Several approaches have been taken to produce denser point clouds from depth sensors and to increase coverage of point cloud data. In [53], a global implicit surface representation of an indoor scene is constructed and built up in real-time from multiple scans using the same Kinect sensor. While this method provides a dense reconstruction of a room-sized surface, localization of scans can be problematic at different scales. Moreover, it requires the movement of a single RGB-D camera around the target surface, a stipulation that may not be feasible in telerobotics or constrained environments.

In another method, multiple RGB-D cameras were used to observe a scene from several angles, followed by registration and stitching of the multiple sources. [46] used separate high frequency motion patterns for each camera, while [12] utilized a simpler technique consisting of offset-weight vibration motors. In [71], it was shown that different point cloud sets with varying sensor characteristics were registered with each other using enhanced iterative closest point algorithms for the purposes of metrology. While these methods provide ways to integrate homogenous scans from multiple sensors and increase the amount of informa-

tion, strategically placing and registering two or more separate RGB-D cameras may be troublesome or not realistic on a space-limited teleoperated robot.

Due to the above reasones, end effector mounted sensors are more amenable to augment the point cloud in grasping tasks than 'arena' sensors such as RGB-D cameras. The tool-tip fixed sensors have two advantages:

- In contrast of the sensors mounted on the robot's head or body, the measurements from the tool-tip are not likely to be occluded by the end effector itself.

- The accuracy and precision of the sensor measurements are left mainly to the robot kinematics without the need of computationally expensive registration with respect to the robot reference frame.

Tool-tip mounted sensors can be characterized to two types: tactile and non-contact. Both types of the sensors may require several probes or palpations of the target object in order to build a point cloud representation.

In the case of tactile sensing, contacts are required for getting sensor readings, as shown in [56]. It may potentially displace the object of interest, and as more data are acquired, the obtained point cloud may become more and more distorted as objects and geometries move [22]. Moreover, in some cases contact in a delicate or sensitive environment could be disastrous.

Pretouch depth information gathered from the robot tool-tip can be used to populate sparse areas of interest as introduced in chapter 3, 4, and 5. The seashell effect pretouch sensor possesses many desirable properties. It provides reliable geometric data within the last one cm prior to touch, and it can sense a diverse variety of materials compared to electrical or optical types. Using this sensor, depth and geometry information within the vicinity of the robot end effector are easily collected during and prior to grasp.

Most importantly, the non-contact nature of pretouch is in line with one of the main purposes of providing haptic feedback in teleoperation– collision avoidance. The non-contact measurements can be used to render the haptic force and assist the operator for exploring the unknown portion of the target object without collision. As the robot end effector

explores, the point cloud representation becomes denser in the explored regions proximal to the object. Finally, the denser virtual point cloud representation of the scene after the teleoperated pretouch explorarion is helpful for completing the grasping task.

In Chapter 5, the proposed iterative grasp replanning and exploration framework reduces object shape uncertainty using a pretouch sensor before grasping and increase the success rate. However, the method is only applicable for autonomous grasping tasks, not human-in-the-loop teleoperation grasping[35].

6.2 Method

To address the problems of incomplete object shape information, the main goal of this work is to integrate pretouch sensing into teleoperated object grasping tasks. A *teleoperated pretouch exploration* is done by the operator in order to augment the point cloud of target object for the unknown areas. A more complete object shape representation after exploration is thus available to the operator, and is helpful for determing the approporate grasping pose. Additionally, the performance of the exploration task is evaluated by comparing the resulting augmented point cloude with and without haptic feedbacks.

6.2.1 Teleoperated Pretouch Exploration

In section 5, an iterative grasp replanning and exploration framework with a pretouch sensor is shown to automatically determine where to sense around the target object, and thus reduce object shape uncertainty for the purpose of grasping. However, the method is only applicable for autonomous grasping tasks, and not applicable to the human-in-the-loop teleoperation grasping task.

In a teleoperated pretouch exploration task, the operator has direct control of the robot end-effector location and orientation at any given time during teleoperation. The teleoperator's goal is to explore an unknown, potentially sensitive object that is partially occluded in the point cloud gathered from the RGB-D camera unitil the denser virtual point cloud representation of the object is sufficiently dense for determining the grasping pose. All the decision making processes and the motion controls are done by the human cognitive skills.

6.2.2 Haptic Feedback in Teleoperated Pretouch Exploration

The purpose of the pretouch exploration is to obtain more complete object shape of the target object to facilitate the grasping task. However, the task requires the operator to control the robot's end effector such that the tip-mounted pretouch sensor is in close proximity to the target object without touching it. While the sensor provide reliable sensor readings, the challenges come from the response time of the operator, and the preciseness of the control. Therefore, providing haptic feedback to the operator to close the control loop can potentially reduce the possibility unintentional contact.

A *proxy* tracking method [66, 67] is used to control the motion of the PR2 grasper, in which a user's position and orientation inputs to the haptic device correspond to scaled position and orientation of a haptic interaction point (*HIP*) within a virtual point cloud representation of the environment:

- The proxy can be thought of the desired end effector position. In the case of the pretouch exploration, the proxy is modeled as a 3DOF sphere at the location of the tip-mounted seashell effect pretouch sensor.

- The HIP can be thought of as the position of an end effector in virtual space, which matches the position of the haptic device. When the user moves the haptic device, the HIP moves accordingly in the virtual environment.

The haptic force is then calculated as if there is a virtual spring-damper coupling between the HIP and the proxy. Futhermore, with the extension of spherical forbidden region virtual fixtures around points [64], either generated from the pretouch sensor or the Kinect, whereby the proxy, the desired grasper position, is restricted from entering within a certain distance of each point. Since the HIP can penetrate into the virtual fixtures and the robot grasper tracks the proxy, the deviation of the positions between proxy and HIP will result in force preventing the operator from further moving the grasper toward the virtual fixtures. By carefully designing the virtual fixrues, collisions can thus potentially be avoided during the pretouch exploration.

6.2.3 Experimental Setup

To evaluate this approach, teleoperator performances is compared in exploring the surface of an unknown grasp object that is not moving under the following two different user feedback conditions:

1. Point cloud and RGB information augmented with seashell pretouch sensor information

2. Point cloud and RGB information augmented with seashell pretouch sensor information and haptic feedback

Scenario 1 represents a baseline case in which the teleoperator may explore occluded or undetected regions of the grasp object with the seashell effect pretouch sensor, thus augmenting the visual feedback received by the operator. To protect the delicate grasp target or to prevent undesired motion during exploration, scenario 2 includes haptic virtual fixtures to prevent unintentional contact.

The experimental setup has a bilateral teleoperation arrangement. The remote/slave robot is a Willow Garages's PR2 robot with a head-mounted RGB-D camera (Kinect) as shown in figure 6.1. A seashell effect pretouch sensor introduced in section 3 is integrated into the PR2 gripper to sense geometries close to the object undetected by the Kinect and augment the incomplete point cloud from Kinect. As described preciously, the sensor has reliable sensing range within $[1, 5]$ mm. The registration of the pretouch sensor readings and point cloud is done through the PR2's kinematics, since both sensors are mounted to known robot frames.

On the master console side, the teleoperator manipulates a master haptic device, the Geomagic® Touch™. This master device both sends position and orientation commands to one of the PR2's arms as well as receives and displays 3DOF haptic force feedback commands when applicable. Figure 6.2 shows the setup of the master console. During teleoperation, the user received visual feedback in the form of an RGB image stream from the PR2 headmount Kinect and the virtual environment rendering (includes proxy location, HIP

(a) (b) (c)

Figure 6.1: The slave/remote side of the teleoperation setup. (a) PR2 with headmounted Kinect. (b) end effector seashell sensor. (c) PR2 grasping an object.

location, surface normal, current PR2 end effector pose, and registered point cloud). When using haptic rendering, the user received force feedback via the Geomagic® Touch™device.

With this setup, the teleoperator's goal is to explore a simple flat surface, the face of a cardboard box, in a specific trajectory as shown in Fig 6.3. The teleoperator is asked to sense an unknown surface of the box while attempting to avoid contact first by exploring along the top edge, and then returning to bottom corner across the diagonal of the box face.

6.3 Results

The teleoperated exploration task as described in section 6.2.3 is performed five times both with and without haptic virtual fixtures. During the trials, the points sensed by the seashell pretouch sensor are both rendered graphically and logged for post-trial quantitative analysis. At the end of each trial, a visual inspection of the point clouds is performed, and a sample of this is shown in Fig 6.4.

The visual results provided a qualitative measure of performance. Ideally, the sensed point cloud should resemble a flat planar surface, as describes a static box surface. With haptic virtual fixtures, the robot end effector is prevented from colliding with the surface, and the teleoperator's hand is pushed away from the sensed surface. In the case of visual feedback only, displacement of the object and less structured points are expected due to the

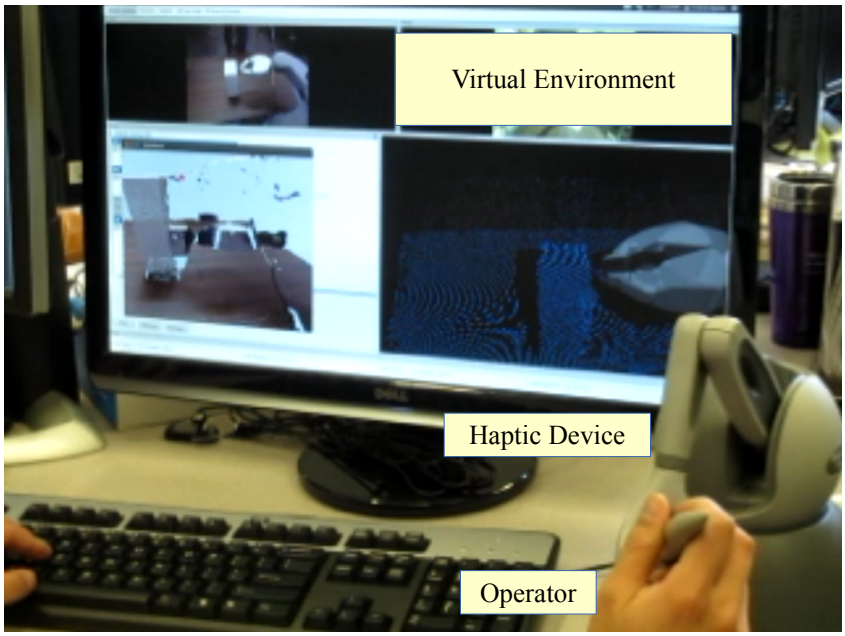


Figure 6.2: The master console of the teleoperation setup.

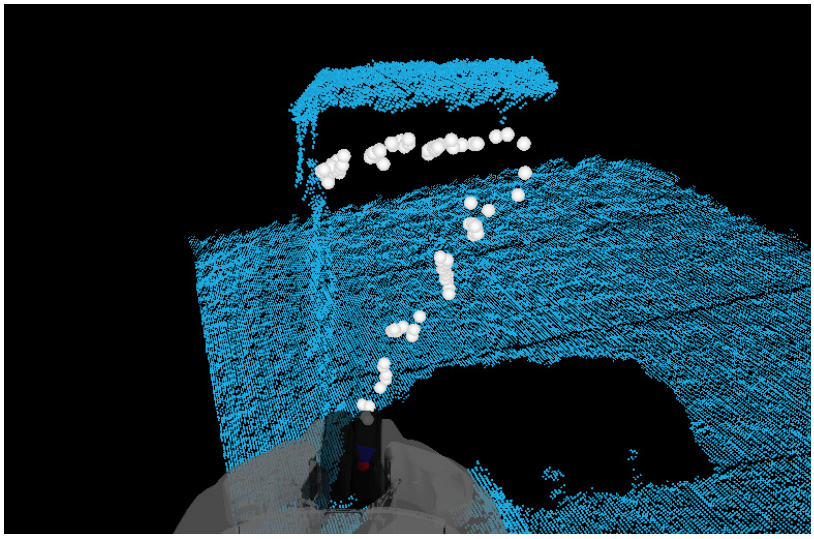


Figure 6.3: Desired exploration trajectory for unknown flat box surface

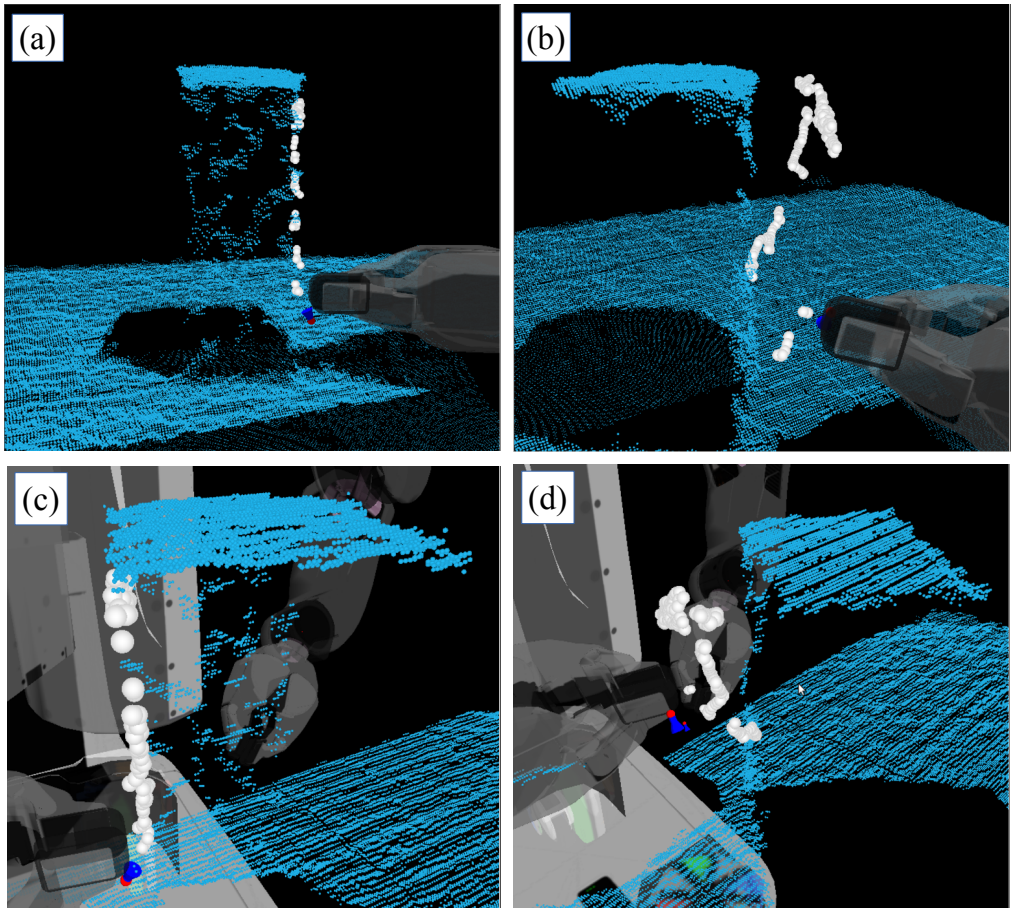
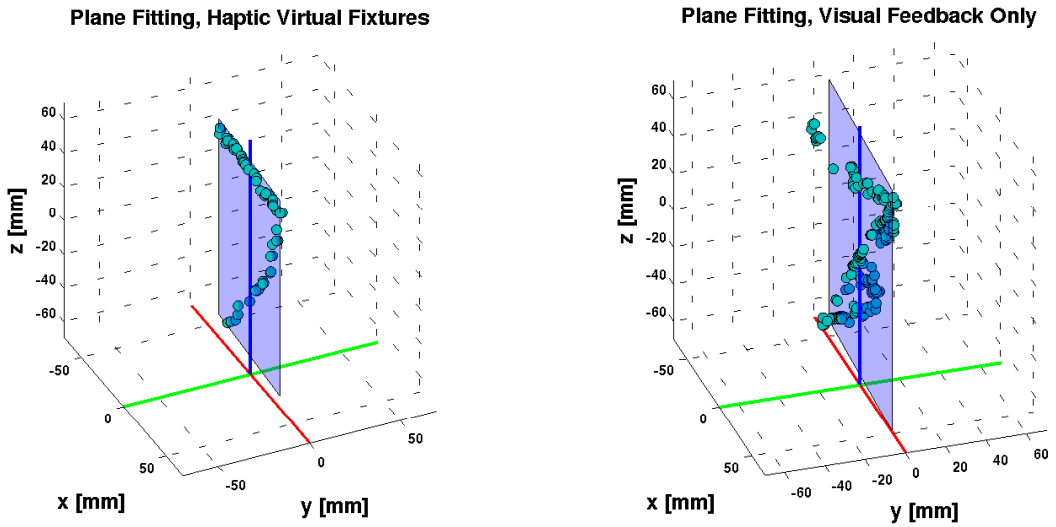


Figure 6.4: Seashell sensor point clouds (a) with haptic virtual fixture, (b) visual feedback only, (c) close-up with haptic virtual fixture, (d) close-up visual feedback only.

difficulty of teleoperating the end effector in the sub-centimeter range. This can be seen in figure 6.4.

A more quantitative comparison is performed by analyzing the recorded sensed point cloud after the exploration task. In particular, a two dimensional plane is fitted in a constrained least squares sense to each of the generated point clouds. The enforced constraint ensured that the plane fitting generated a plane perpendicular to the plane described by the robot base z axis (the explored box surface is approximately perpendicular to this plane). Figures 6.5 and 6.6 show plane fitting results from trials with points sensed with both visual and haptic feedback and those with visual feedback only.



(a) (b)

Figure 6.5: Plane fitting visual results of the teleoperated pretouch augmented points (a) with haptic virtual fixture, (b) visual feedback only.

The deviation from the fitted planes is calculated for each dataset. Additionally, the number of collisions, time to completion, as well as object displacement are measured for each test. Unless stated otherwise (i.e. preceded with 'total' or 'RMS'), the data in table 6.1 represent the mean value across five trials.

The experimental results show that haptic virtual fixtures around pretouch sensed points

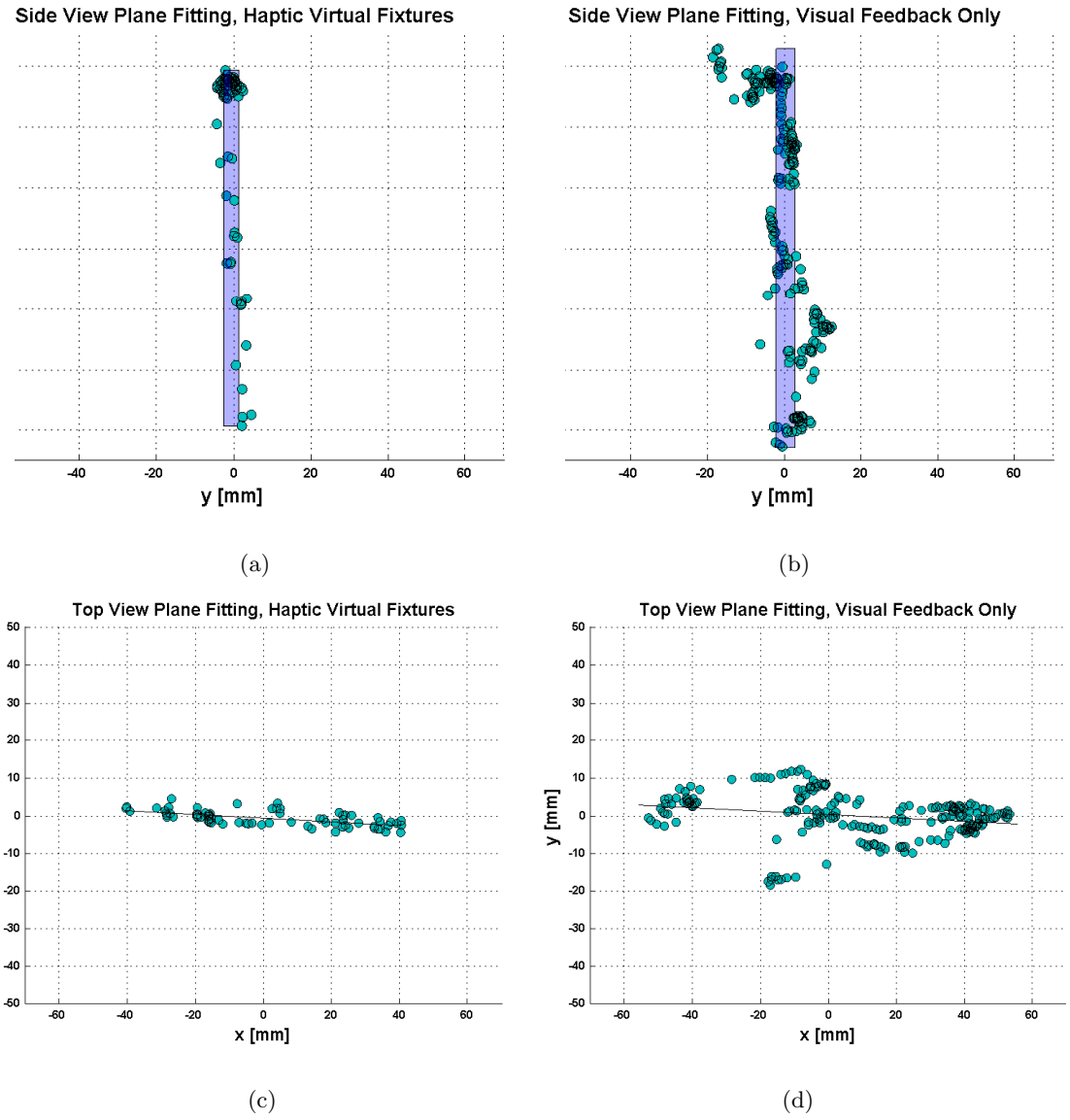


Figure 6.6: Side/top views of the plane fitting results of the teleoperated pretouch augmented points {(a),(c)} with haptic virtual fixture, {(b),(d)} visual feedback only.

can significantly reduce the number of collisions and object displacement, at the cost of less point density and longer completion time. This is an expected tradeoff, since the haptic virtual fixtures restrict the user's motion (longer time), and limit the robot end effector motion near the surface (less points). This combination of techniques can be used

Table 6.1: Experimental results of the exploration tasks

	Visual	**Haptic + Visual**
Total trials	5	5
Number of sensed points	295	1217
Task completion time [s]	185.4 ± 27.7	108.6 ± 17.3
Total collisions	1	71
Collisions	0.2± 0.4	14.2±1.6
Displacement [mm]	N/A	17.6± 7.1
Deviation from plane [mm]	1.468 ± 1.171	3.797 ± 3.033
RMS deviation from plane [mm]	1.878	4.860

in grasping tasks to improve the teleoperator's performance. It also allows for reaction to moving objects (e.g. to avoid unintended contact).

Teleoperated Grasping

The experiment performed above has shown the efficacy of haptic feedback during pretouch exploration task in order to augment the point cloud without contact. The augmented point cloud can provide very important information about the target object shape to the operator in the teleoperated grasping setup. An example is shown in figure 6.7, in which the goal of the operator is to teleoperate the robot to grasp the transparent bottle on the cluttered tabletop. In figure 6.7(b), the point cloud in blue color is sensed by the Kinect sensor, and it represents incomplete shapes of the two bottles in the virtual environment. Since the teleoperator relies solely on the virtual environment, it is likely the grasp to be

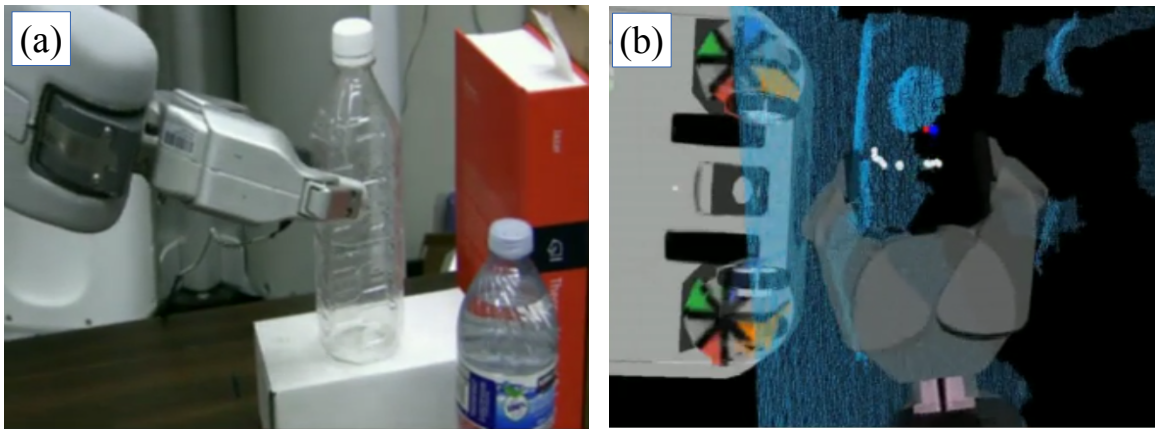


Figure 6.7: Teleoperated grasp after pretouch exploration. (a) The PR2 robot grasping the transparent bottle in clutter. (b) Align the gripper with the help of augmented point cloud

misaligned with the bottle. The white point cloud added during the pretouch exploration hence provides more complete object shape and helps the operator move the robot gripper into an aligned pre-grasp pose before approaching to grasp it. Figure 6.7(a) shows the PR2 robot succesfullty grasps the transparent bottle in a clutter environment using this mothod.

Limitations

Several factors limit the applicability of this work to specific scenarios (exploration in open space of convex object). The haptic virtual fixtures affect only a singular point, namely the HIP. For collision avoidance, this method is amenable only under the assumption of a convex grasp object without acute angles or neighboring objects. In other cases, the grasper itself may collide with the grasp target or nearby surfaces. While the seashell pretouch sensor can detect surfaces in great detail within the sub-centimeter range, a larger sensing range would allow for surface detection from greater distances and thus provide a safer distance from which to explore and avoid contact. Finally, this approach deals with nonmoving environments and objects, since the sensed points are nonmoving as well.

6.4 Summary

In this chapter, a method of fusing pretouch sensing with streaming point cloud data from the depth sensor is demonstrated for the purposes of both haptic rendering and refining the shape of the object without physical contact in a teleoperated grasping scenario.

A seashell effect pretouch sensor fixed to the robot end effector is used to sense physical geometries in the vicinity of the target object. The point cloud representation of an unknown environment, which may be sparse or poorly visible, can be enhanced through telerobotic exploration/sensing in real-time. Virtual fixtures suitable for pretouch sensing are designed; real-time haptic rendering algorithms are applied; experiments are conducted to verify the effectiveness of haptic feedback in teleoperated pretouch exploration. The experimental results show that haptic virtual fixtures around pretouch sensed points can significantly reduce the number of collisions and object displacement. This method provides the teleoperator with critical geometrical information about the grasp target while simultaneously preventing the robot end effector from colliding with it. Finally, the augmented virual environment after the pretouch exploration represents more complete object shapes, which helps the operator align the slave robot's gripper with the target object for grasping.

Chapter 7

CONCLUSION

This dissertation has focused on developing methods of pretouch sensing to improve the robotic grasping capability for novel objects, based on the shape when their models are not available in advance. It has been a challenging task due to the shape uncertainty caused by common sensor failures and noisy sensor data. The research is a combination effort of (1) improving the robot's perception capability with new pretouch sensors, and (2) developing novel algorithms and frameworks to make use of the sensor information efficiently for grasping.

The addition of pretouch sensing capability offers the potential to fundamentally change the way robots grasp objects. Effective means of the short range pretouch sensing in the vicinity of the target object will allow the robot to feel the shape at a distance without contact. Although the cognitive capability of the robots is not yet as good as human beings, with the help of this superior perception modality, the grasping capability can be much improved. The proposed methods are verified experimentally in both autonomous robotic grasping and teleoperated robotic grasping scenarios.

7.1 Dissertation Summary

The first focus in this work is on developing a novel pretouch sensing modality, "seashell effect pretouch". This pretouch based on acoustics is effective for a set of materials that previous pretouch techniques fail to sense and complements the set of materials that the robot can sense. To the best of my knowledge, this effect has not previously been used to build proximity sensors. In order to turn the familiar seashell effect into a short range proximity (pretouch) sensor, I study the underlying acoustic principle, i.e., the acoustic radiation impedence changes caused by the object being close to the opening of the cavity. The sensor design including the acoustic properties, hardware/software design, and signal

processing are discussed in detail. The resulting implementation is fully integrated into the Willow Garage PR2 robot's finger. The sensor detects changes in the spectrum of ambient noise when the finger approaches an object. The performance of the proposed sensor in terms of the sensing range, accuracy, and the material selectivity are characterized and evaluated The setup is able to detect the presence of the object reliably in the close distance up to 5 mm. The designed PR2 finger sensor board is open-sourced and is available to other PR2 users over the world. I also show that the flexibility of the proposed sensor system design enable development of a IR optical beam pretouch sensor with minimal modification.

The first explored application of the proposed sensor is detecting extremely compliant objects during grasp execution. A pre-grasp execution experiment is performed, in which the pressure sensor built in to the PR2's gripper is compared to the seashell effect pretouch sensor for the ability to detect compliant objects. The results suggest advantages of seashell pretouch over tactile sensing in this scenario.

The second application is pretouch-assisted grasp planning. When the pretouch sensor sensed the object during a series of probing motions, it provided points collected by recording the position of the robot's end effector; these additional points augmented the point cloud from the depth sensor. This method compensated for incomplete object shape due to depth sensor failure or occlusion. Furthermore, a unified probabilistic framework is proposed to (1) identify shape uncertainty for the target object; and (2) automatically explore the uncertain areas with a pretouch sensor to reduce the uncertainty toward a success grasp with high confidence. The environment perceived by the robot is represented by a probabilistic occupancy map. The robot starts with only the incomplete object shape data acquired from a Kinect depth sensor—it has no model of the object. Next, combining the Kinect point cloud with prior probability distributions for occlusion and transparency, it makes inferences about portions of the object unobserved by the Kinect sensor. Operating on the inferred shape of the object, an iterative grasp replanning algorithm decides whether further exploration is required, and where to explore in the scene. The information gathered by the exploration action is added directly to the robot's environment representation in real-time and is considered in the next grasp planning iteration. The experimental results showed that, the robot is able to grasp partially transparent objects with high success rates

(96%). I believe that combining pretouch sensing and a grasping-oriented active sensing exploration schemes is one of the effective methods to address the shape uncertainty problem for unknown objects.

Finally, I present a method of augmenting streaming point cloud data with the seashell effect pretouch information for the purposes of haptic rendering in a telerobotic grasping task. The non-contact seashell-effect pretouch sensor fixed to the robot end effector is used to sense physical geometries within the vicinity of the sensor. Thus, the point cloud representation of an unknown environment, which may be sparse or poorly visible, can be enhanced through telerobotic exploration/sensing in real-time. Furthurmore, real-time haptic rendering algorithms are appliedon the augmented point clouds to create haptic virtual fixtures, and also provide haptic force feedback to the operator. This method provides the teleoperator with critical geometrical information about the grasp target while simultaneously preventing the robot end effector from colliding with it. The augmented virual environment after the pretouch exploration represents more complete object shapes, which helps the operator align the slave robot's gripper with the target object for grasping.

7.2 Future Work

The potential future work for this dissertation can be in pretouch sensor development, advanced algorithms for using the pretouch sensors, and other applications which can benefit from utilizing pretouch sensing.

Pretouch Sensor Development

Short range pretouch sensing techniques are still at the early stage compared with matured depth sensing techniques such as laser scanners, stereo cameras, and RGB-D cameras. When developing personal robotic applications, people still tend to use standard off-the-shelf sensors. I belive that pretouch sensing has the potential to play an important role in improving the grasping capaility of personal robots. One common problem when promoting new sensing technologies is the lack of consolidation of efforts among the communities. Open source development model has been widely adapted in software development for a long time, and nowadays been applied to the robotics domain. The Robot Operating System (ROS), a

flexible open source framework for robot software, has been adapted by the robotic communities all over the world. Additionally, PR2 is a robotic platform designed with the vision of unifying hardware for research. These are both significant milestones. However, when it comes to developing individual sensors on the robot's hand, it becomes extremely hard to reuse the developed sensors. This is mainly due to the wide variety of robot hand designs.

The approach I have taken toward this goal in this dissertation is to make my sensor design modular, and adapt as much exsisting hardware interface on a common robotic platform (the PR2) as possible. I designed two pretouch sensors using the proposed flexible system design, namely the seashell effect pretouch sensor and the infrared optical beam sensor. In the future, the sensors can be distributed to all the PR2 users. For the users of other robots, my sensor may also be installed with minimal modification since I use a standard communication protocol (SPI) to transmit sensor data. In future work, a new version of the sensor board utilizing the USB interface for communication can be implemented to make it even easier to use.

Additionally, one can integrate multiple sensing modalities into the current finger sensor board given the high extensibility of the board. In particular, each of the E-field, optical, and seashell effect sensing method for pretouch has been studied extensively, and can be integrated with each other. A sensor combining the above three pretouch modalities possesses a broad spectrum in sensing range and characteristics (acoustic, mechanical, optical, and electrical). Furthermore, even tactile sensors may co-exist on the same fingertip sensor board.

Algorithms and Applications

In chapter 5, a unified probabilistic framework for shape acquisition and grasping with pretouch sensor is described. It provides a general framework for the robot to fuse sensor information from all different available equipped sensors, and grasp the target object with high confidence. Experiments using the Kinect depth sensor and the seashell effect pretouch sensing are performed to demonstrate the efficacy of the framework. However, the framework is not limited to pretouch sensors. In the future, additional sensor such as

tactile sensors, or an improved pretouch sensor with multiple modalities can possibly be added into the framework seamlessly, since it only requires additional Bayesian updates on the probabilistic occupancy map during the exploration. The computation complexity grows linearly with the number of sensors. Using multiple pretouch sensing modalities is expected to make each iteration during the exploration more informative (i.e., reduce more uncertainty).

Currently the inference step in the framework uses experimentally determined predefined exponential priors for both the potential transparent and occluded portions of the objects. Generalized priors for both cases may be learned from the point cloud data and the object models. While the current planner based on the weighted PCA already takes the object shape uncertainty into account, it may not be the best-suited grasp planner for "grasp replanning". For instance, a planner in favor of previously explored areas may speed up the convergence of the iterative grasp replanning algorithm. The scalability to multiple target objects can also be evaluated in the future. Finally, it is possible to combine the grasp planning and grasp evaluation with graspable features in the occupancy grids by matching against a large database [85].

In section 4.1, it is shown the seashell effect pretouch sensor can be used to sense extremely compliant objects. This application can be extended to the actual object grasping scenarios. The robot can actually close the loop and grasp the object when the compliant object is detected by the pretouch sensor as opposed to adjusting its gripper reactively based on sensor readings. The contact detection with the seashell effect pretouch sensor can be done by further studying the frequency contents of the acoustic cavity when a contact is made with the object. Alternatively, it can be done by analyzing the raw sound signal to monitor the sound when the sensor hitting the object.

Regarding the application in the teleoperated grasping with haptic feedback, an immediate step forward is the use of 6DOF haptic rendering algorithms based on the augmented point cloud for more robust collision avoidance, which was done in previous work [65]. In addition, a sensor array or a sensor with wider sensing region may provide more information in the general vicinity of the tool tip. These two future improvements together may extend the scope of this application to non-concave objects, and enable more applications.

The teleoperator may even be able to feel the object from the whole gripper during the grasping task without contacts.

BIBLIOGRAPHY

[1] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R.B. Rusu, and G. Bradski. Cad-model recognition and 6dof pose estimation using 3d cues. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 585–592, Nov 2011.

[2] J. Amanatides and A. Woo. A fast voxel traversal algorithm for ray tracing. In *Eurographics,*, Amsterdam, The Netherlands, August 1987.

[3] R. Anderson and M.W. Spong. Bilateral control of teleoperators with time delay. *Automatic Control, IEEE Transactions on*, 34(5):494–501, May 1989.

[4] L. Arapan, E. Anderas, I. Katardjiev, and V. Yantchev. Sensitivity features of thin film plate acoustic wave resonators. *IEEE Sensors Journal*, 11(12):3330–3331, 2011.

[5] Paolo Arcara and Claudio Melchiorri. Control schemes for teleoperation with time delay: A comparative study. *Robotics and Autonomous Systems*, 38(1):49 – 64, 2002.

[6] M. Berouti, R. Schwartz, and J. Makhoul. Enhancement of speech corrupted by acoustic noise. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '79.*, volume 4, pages 208–211, Apr 1979.

[7] A. Bicchi. On the Closure Properties of Robotic Grasping. *The International Journal of Robotics Research*, 14(4):319–334, August 1995.

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

[9] D. T. Blackstock. *Fundamentals of Physical Acoustics*. Wiley-Interscience, 1 edition, 2000.

[10] S. Boll. Suppression of acoustic noise in speech using spectral subtraction. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 27(2):113–120, Apr 1979.

[11] P. Brook, M Ciocarlie, and K. Hsiao. Collaborative grasp planning with multiple object representations. In *IEEE International Conference on Robotics and Automation*, 2011.

[12] D. Alex Butler, Shahram Izadi, Otmar Hilliges, David Molyneaux, Steve Hodges, and David Kim. Shake'n'sense: Reducing interference for overlapping structured light depth cameras. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1933–1936, New York, NY, USA, 2012. ACM.

[13] Matei Ciocarlie, Kaijen Hsiao, EdwardGil Jones, Sachin Chitta, RaduBogdan Rusu, and IoanA. ucan. Towards reliable grasping and manipulation in household environments. In Oussama Khatib, Vijay Kumar, and Gaurav Sukhatme, editors, *Experimental Robotics*, volume 79 of *Springer Tracts in Advanced Robotics*, pages 241–252. Springer Berlin Heidelberg, 2014.

[14] J. W. Coltman. Differentiating sonar reflections from corners and planes by employing an intelligent sensor. *Journal of the Acoustical Society of America*, 65:499–506, 1978.

[15] J.-P. Dalmont, C.J. Nederveen, and N. Joly. Radiation impedance of tubes with different flanges: Numerical and experimental investigations. *Journal of Sound and Vibration*, 244(3):505–534, 2001.

[16] H. Dang, J.Weisz, and P. Allen. Blind grasping: Stable robotic grasping using tactile feedback and hand kinematics. In *IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011.

[17] N. Diolaiti and Claudio Melchiorri. Teleoperation of a mobile robot through haptic feedback. In *Haptic Virtual Environments and Their Applications, IEEE International Workshop 2002 HAVE*, pages 67–72, 2002.

[18] B. Diziolu and K. Lakshiminarayana. Mechanics of form closure. *Acta Mechanica*, 52(1-2):107–118, 1984.

[19] Mehmet Dogar and Siddhartha Srinivasa. A framework for push-grasping in clutter. In *Proceedings of Robotics: Science and Systems*, Los Angeles, USA, 2011.

[20] J. Glover, D. Rus, and N. Roy. Probabilistic models of object geometry for grasp planning. In *Robotics: Science and Systems Conference*, 2009.

[21] Michael A. Goodrich and Alan C. Schultz. Human-robot interaction: A survey. *Found. Trends Hum.-Comput. Interact.*, 1(3):203–275, January 2007.

[22] N. Gorges, S.E. Navarro, and Heinz Worn. Haptic object recognition using statistical point cloud features. In *Advanced Robotics (ICAR), 2011 15th International Conference on*, pages 15–20, 2011.

[23] B. Hannaford. A design framework for teleoperators with kinesthetic feedback. *Robotics and Automation, IEEE Transactions on*, 5(4):426–434, Aug 1989.

[24] P. Hebert, J. W. Burdick, T. Howard, N. Hudson, and J. Ma. Action inference: The next best touch. In *Robotics: Science and Systems Workshop on Mobile Manipulation*, Sydney, Australia, 2012.

[25] Peter F. Hokayem and Mark W. Spong. Bilateral teleoperation: An historical survey. *Automatica*, 42(12):2035 – 2057, 2006.

[26] G.A. Hollinger, B. Englot, F. Hover, U. Mitra, and G.S. Sukhatme. Uncertainty-driven view planning for underwater inspection. In *IEEE International Conference on Robotics and Automation*, 2012.

[27] K. Hsiao, S. Chitta, M. Ciocarlie, and E.G. Jones. Contact-reactive grasping of objects with partial shape information. In *IEEE International Conference on Robotics and Automation*, 2010.

[28] K. Hsiao, M. Ciocarlie, and P. Brook. Bayesian grasp planning. In *IEEE International Conference on Robotics and Automation*, 2011.

[29] K. Hsiao, Paul 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*, pages 2098–2105, May 2009.

[30] Kaijen Hsiao, L. Kaelbling, and T. Lozano-Perez. Grasping pomdps. In *IEEE International Conference on Robotics and Automation*, Roma, Italy, 2007.

[31] E. Jacobsen and P. Kootsookos. Fast, accurate frequency estimators. *IEEE Signal Processing Magazine*, 24(3):123–125, 2007.

[32] Shervin Javdani, Matthew Klingensmith, J. Andrew (Drew) Bagnell, Nancy Pollard, and Siddhartha Srinivasa. Efficient touch based localization through submodularity. Technical Report CMU-RI-TR-12-25, Robotics Institute, Pittsburgh, PA, 2012.

[33] L. Jiang and J.R. Smith. Seashell effect pretouch sensing for robotic grasping. In *IEEE International Conference on Robotics and Automation*, Saint Paul, USA, 2012.

[34] L. Jiang and J.R. Smith. Design and characterization of the seashell effect pretouch sensor integrated into robot grippers. In *SENSORDEVICES 2013, The Fourth International Conference on Sensor Device Technologies and Applications*, pages 34–40, Barcelona, Spain, 2013.

[35] Liang-Ting Jiang and J.R. Smith. A unified framework for grasping and shape acquisition via pretouch sensing. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 999–1005, May 2013.

[36] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner, and K. Goldberg. Cloud-based robot grasping with the google object recognition engine. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4263–4270, May 2013.

[37] L. V. King. On the electrical and acoustic conductivities of cylindrical tubes bounded by infinite flanges. *Journal of the Acoustical Society of America*, 21:128–144, 1936.

[38] L.E. Kinsler, A.R. Frey, A.B. Coppens, and J.V. Sanders. *Fundamentals of Acoustics*. Wiley, 4 edition, 1999.

[39] Kevin Lai and Dieter Fox. Object recognition in 3d point clouds using web data and domain adaptation. *I. J. Robotic Res.*, 29(8):1019–1037, 2010.

[40] Y. Lamdan, Jacob T. Schwartz, and H.J. Wolfson. Affine invariant model-based object recognition. *Robotics and Automation, IEEE Transactions on*, 6(5):578–589, Oct 1990.

[41] Sangyoon Lee, G. Sukhatme, Jounghyun Kim, and Chan-Mo Park. Haptic control of a mobile robot: a user study. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2867–2874 vol.3, 2002.

[42] A. Leeper, K. Hsiao, M. Ciocarlie, Leila Takayama, and D. Gossow. Strategies for human-in-the-loop robotic grasping. In *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on*, pages 1–8, March 2012.

[43] Adam Leeper, Kaijen Hsiao, Matei Ciocarlie, Ioan Sucan, and Kenneth Salisbury. Methods for collision-free arm teleoperation in clutter using constraints from 3d sensor data. In *IEEE Intl. Conf. on Humanoid Robots*, Atlanta, GA, 10/2013 2013.

[44] H. Levine and J. Schwinger. On the radiation of sound from an unflanged circular pipe. *Physical Review*, 73:383–406, 1948.

[45] I. Lysenkov, V. Eruhimov, and G. Bradski. Recognition and pose estimation of rigid transparent objects with a kinect sensor. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, 2012.

[46] A. Maimone and H. Fuchs. Reducing interference between multiple structured light depth sensors using motion. In *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE*, pages 51–54, March 2012.

[47] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2308–2315, May 2010.

[48] A. Maldonado, H. Alvarez, and M. Beetz. Improving robot manipulation through fingertip perception. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2947–2954, Oct 2012.

[49] B. Mayton, L. LeGrand, and J.R. Smith. An electric field pretouch system for grasping and co-manipulation. In *IEEE International Conference on Robotics and Automation*, 2010.

[50] A.T. Miller and P.K. Allen. Graspit! a versatile simulator for robotic grasping. *Robotics Automation Magazine, IEEE*, 11(4):110–122, Dec 2004.

[51] A.T. Miller, S. Knoop, H.I. Christensen, and P.K. Allen. Automatic grasp planning using shape primitives. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 2, pages 1824–1829 vol.2, Sept 2003.

[52] B. Mishra, J.T. Schwartz, and M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica*, 2(1-4):541–558, 1987.

[53] Richard A. Newcombe, Andrew J. Davison, S. Izadi, P. Kohli, Otmar Hilliges, J. Shotton, David Molyneaux, Steve Hodges, David Kim, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pages 127–136, 2011.

[54] G. Niemeyer and J.-J.E. Slotine. Stable adaptive teleoperation. *Oceanic Engineering, IEEE Journal of*, 16(1):152–162, Jan 1991.

[55] A. M. Okamura. Methods for haptic feedback in teleoperated robot-assisted surgery. *Ind Rob*, 31(6):499–508, Dec 2004.

[56] A. Petrovskaya and O. Khatib. Global localization of objects via touch. *IEEE Transactions on Robotics*, 27(3):569–585, 2011.

[57] A. Petrovskaya, O. Khatib, S. Thrun, and A.Y. Ng. Bayesian estimation for autonomous object manipulation based on tactile sensors. In *IEEE International Conference on Robotics and Automation*, 2006.

[58] R. Platt, L. Kaelbling, T. Lozano-Perez, and R Tedrake. Non-gaussian belief space planning: Correctness and complexity. In *IEEE International Conference on Robotics and Automation*, Saint Paul, USA, 2012.

[59] R. Platt, F. Permenter, and J. Pfeiffer. Inferring hand-object configuration directly from tactile data. In *ICRA Workshop on Mobile Manipulation*, Anchorage, USA, 2010.

[60] C. Potthast and G. S. Sukhatme. A probabilistic framework for next best view estimation in a cluttered environment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, USA, 2011.

[61] J.M. Romano, K. Hsiao, G. Niemeyer, S. Chitta, and K.J. Kuchenbecker. Human-inspired robotic grasp control with tactile sensing. *IEEE Transaction on Robotics*, 27(6):1067–1079, 2011.

[62] S. Ross, J. Pineau, S. Paquet, and B. Chaib-draa. Using bayesian filtering to localize flexible materials during manipulation. *The Journal of Machine Learning Research*, 32, 2008.

[63] R.B. Rusu. 3d is here: Point cloud library (pcl). In *IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011.

[64] F. Ryden and H.J. Chizeck. Forbidden-region virtual fixtures from streaming point clouds: Remotely touching and protecting a beating heart. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 3308–3313, Oct 2012.

[65] F. Ryden and H.J. Chizeck. A method for constraint-based six degree-of-freedom haptic interaction with streaming point clouds. In *Robotics and Automation, 2013. ICRA 2013. Proceedings 2013 IEEE International Conference on*, pages 2353–2359, May 2013.

[66] F. Ryden and H.J. Chizeck. A proxy method for real-time 3-dof haptic rendering of streaming point cloud data. *Haptics, IEEE Transactions on*, 6(3):257–267, 2013.

[67] F. Ryden, S.N. Kosari, and H.J. Chizeck. Proxy method for fast haptic rendering from time varying point clouds. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 2614–2619, 2011.

[68] Jee-Hwan Ryu, Dong-Soo Kwon, and B. Hannaford. Stable teleoperation with time-domain passivity control. *Robotics and Automation, IEEE Transactions on*, 20(2):365–373, April 2004.

[69] A. Sahbani, S. El-Khoury, and P. Bidaud. An overview of 3d object grasp synthesis algorithms. *Robotics and Autonomous Systems*, 60(3):326 – 336, 2012. Autonomous Grasping.

[70] Ashutosh Saxena, Justin Driemeyer, Justin Kearns, Chioma Osondu, and AndrewY. Ng. Learning to grasp novel objects using vision. In Oussama Khatib, Vijay Kumar, and Daniela Rus, editors, *Experimental Robotics*, volume 39 of *Springer Tracts in Advanced Robotics*, pages 33–42. Springer Berlin Heidelberg, 2008.

[71] N. Senin, B. M. Colosimo, and M. Pacella. Point set augmentation through fitting for enhanced icp registration of point clouds in multisensor coordinate metrology. *Robot. Comput.-Integr. Manuf.*, 29(1):39–52, February 2013.

[72] T.B. Sheridan. Space teleoperation through time delay: review and prognosis. *Robotics and Automation, IEEE Transactions on*, 9(5):592–606, Oct 1993.

[73] K. B. Shimoga. Robot Grasp Synthesis Algorithms: A Survey. *The International Journal of Robotic Research*, 15(3):230–266, June 1996.

[74] A. R. Da Silva, P. H. Mareze, and A. Lenzi. Approximate expressions for the reflection coefficient of ducts terminated by circular flanges. *Journal of Brazil Society of Mechenical Science and Engineering*, 34(2), 2012.

[75] Joshua R. Smith, Eric Garcia, Ryan Wistort, and Ganesh Krishnamoorthy. Electric field imaging pretouch for robotic graspers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, USA, 2007.

[76] S. Stramigioli, R. Mahony, and P. Corke. A novel approach to haptic tele-operation of aerial robot vehicles. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 5302–5308, May 2010.

[77] J.C. Trinkle. A quantitative test for form closure grasps. In *Intelligent Robots and Systems, 1992., Proceedings of the 1992 lEEE/RSJ International Conference on*, volume 3, pages 1670–1677, Jul 1992.

[78] D. Tsetserukou, K. Sato, N. Kawakami, and S. Tachi. Teleoperation system with haptic feedback for physical interaction with remote environment. In *ICCAS-SICE, 2009*, pages 3353–3358, Aug 2009.

[79] A. Weiss, D. Wurhofer, M. Lankes, and M. Tscheligi. Autonomous vs. tele-operated: How people perceive human-robot collaboration with hrp-2. In *Human-Robot Interaction (HRI), 2009 4th ACM/IEEE International Conference on*, pages 257–258, March 2009.

[80] Peter D. Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *Audio and Electroacoustics, IEEE Transactions on*, 15(2):70–73, Jun 1967.

[81] Ryan Wistort and Joshua R. Smith. Electric field servoing for robotic manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, 2008.

[82] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In *ICRA Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, Anchorage, USA, May 2010.

[83] H. Zhang and E. Kim. Micromachined acoustic resonant mass sensor. *Journal of Microelectromechanical Systems*, 2005.

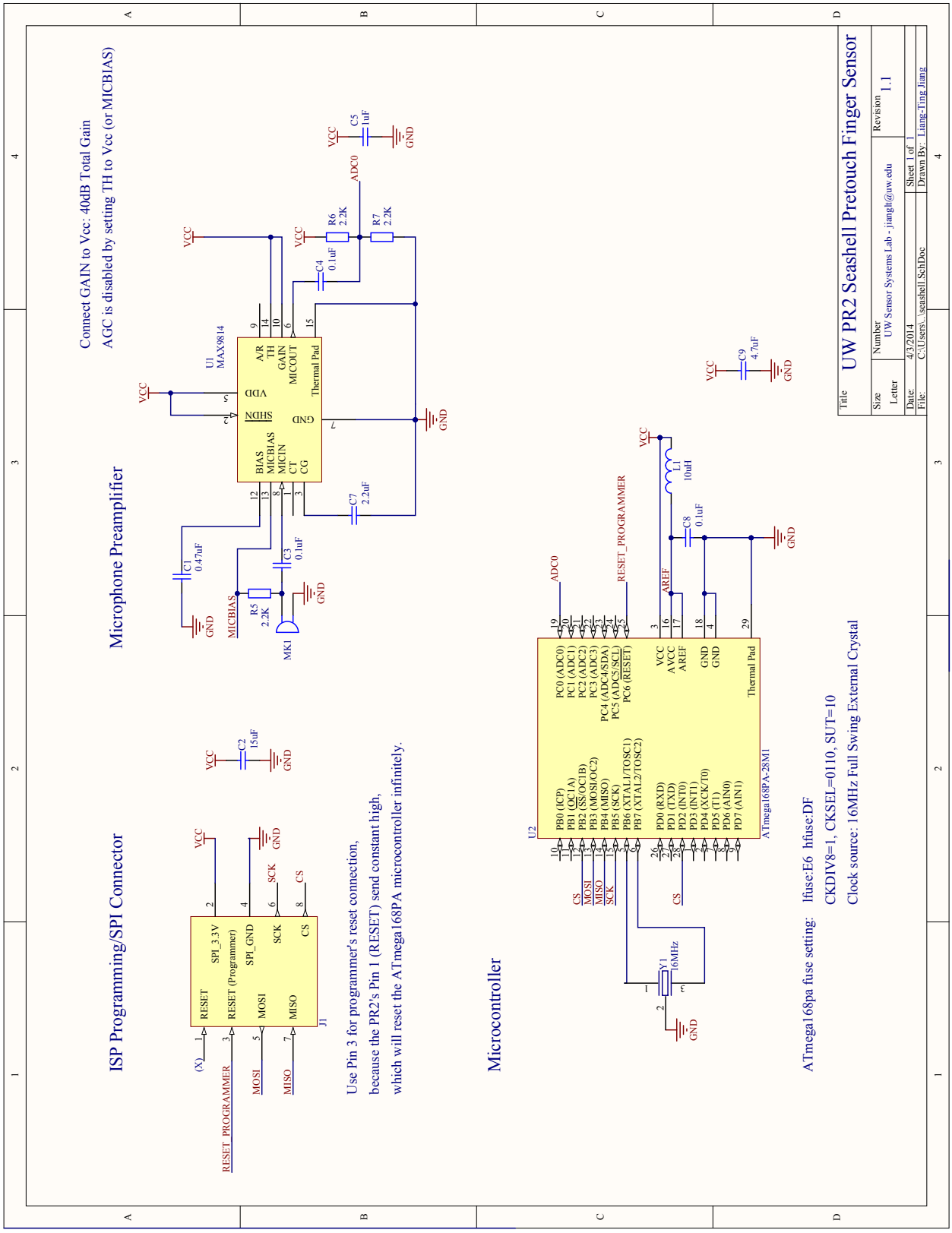
[84] H. Zhang, M. Marma, S.Bahl, E. Kim, and C. McKenna. Sequence specic label-free dna sensing using film-bulk-acoustic-resonators. *IEEE Sensors Journal*, 7(12):1587–1588, 2007.

[85] L. Zhang, M. Ciocalie, and K. Hsiao. Evaluation with graspable feature matching. In *RSS Workshop on Mobile Manipulation: Learning to Manipulate*, Los Angeles, USA, 2011.

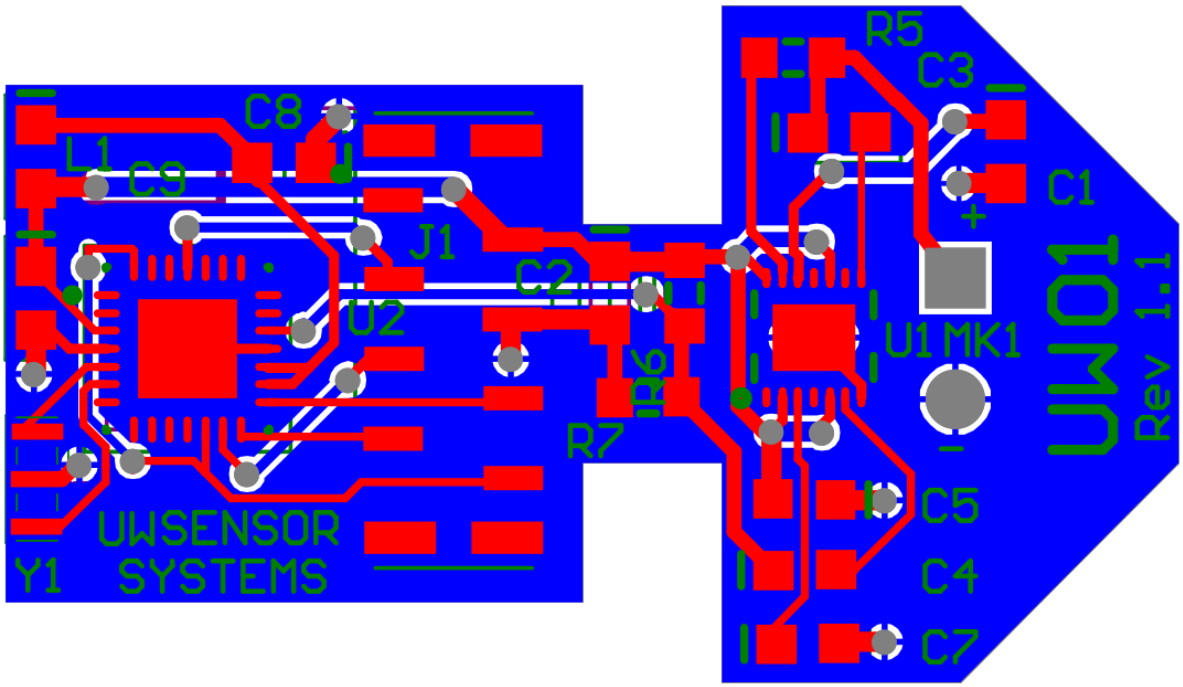
Appendix A

PRETOUCH SENSOR HARDWARE DESIGN DOCUMENT AND FIRMWARE

A.1 Seashell Effect Pretouch Sensor Board – Schematics



A.2 Seashell Effect Pretouch Sensor Board – Layout and Bill of Materials



Comment	Description	Designator	Footprint	LibRef	Quantity
0.47uF	0603 SMD Capacitor	C1	CAPC1608N	Cap	1
15uF	0603 SMD Capacitor	C2	CAPC1608N	Cap	1
0.1uF	0603 SMD Capacitor	C3, C4, C8	CAPC1608N	Cap	3
1uF	0603 SMD Capacitor	C5	CAPC1608N	Cap	1
2.2uF	0603 SMD Capacitor	C7	CAPC1608N	Cap	1
4.7uF	0603 SMD Capacitor	C9	CAPC1608N	Cap	1
FFC 8-PIN Connector (1mm)	FFC 8-PIN Connector (1mm) (SMT)	J1	Hirose FFC connector 8-Pin	FFC Connector-8Pin-SMT-1mm	1
10uH	0603 SMD Inductor	L1	INDC1608N	Inductor	1
Panasonic WM-61A	Electret Microphone	MK1	PIN2	Mic1	1
2.2K	0603 SMD Resistor	R5, R6, R7	RESC1608N	Res2	3
MAX9814	Microphone Amplifier with AGC and Low-Noise Microphone Bias	U1	QFN-14-Minnimum-Solder-Mask-Relief	MAX9814	1
ATmega168PA-28M1	8-Bit AVR Microcontroller with 8K Bytes of In-System Programmable Flash Memory	U2	28M1	ATmega168-28M1	1
AVX PRQC16.00SR5010X	16MHz Ceramic Chip Res	Y1	AVX-PRQC Ceramic Reso	Ceramic Resonator SMT	1

A.3 Seashell Effect Pretouch Sensor – AVR Microcontroller Firmware

```
/*********************************************************************
  Firmware for the PR2 seashell effect pretouch sensor board
  The onboard ADC is used to sample the amplified sound signal
  Serve as SPI Slave to send sampled data to SPI Master (PicoBlaze)
  microcontroller: Atmel AVR ATMega168(PA)/328(P)
  @Liang-Ting Jiang 01-16-2014
*********************************************************************/
#include <avr/io.h>
#include <avr/interrupt.h>

char SPI_SlaveReceive(void) {
  // Wait for reception complete (until SPIF is set)
  while(!(SPSR & (1<<SPIF)));
  return SPDR;           //return the value received from Master
}

int main (void) {
  /********************** EXTERNAL INTERRUPT ***********************/
  DDRD &= ~(1<<DDD2);     //setup External interrupt 0 as Input (PD2)
  PORTD |= (1<<PD2);      //set PD2(INT0) to high (impedance)
  EIMSK |= (1<<INT0);     //enable INT0
  EICRA |= (1<<ISC01);    //trigger INT0 on falling edge
  /***************************** ADC *******************************/
  //using ADC0 -- No MUX values change required (default)
  //ADC prescaler=4 -- ADC clock rate  = 4MHz @ 16MHz clock rate
  ADCSRA |= (1<<ADPS1);
  ADMUX  |= (1<<REFS0);  //set ADC reference to AVCC
  ADMUX  |= (1<<ADLAR);  //left adjust results to allow 8 bit reading
  ADCSRA |= (1<<ADEN);   //enable ADC
  ADCSRA |= (1<<ADIE);   //enable ADC Interrupt
  sei();                 //enable Global Interrupts
  /***************************** SPI *******************************/
  DDRB = (1<<DDB4);      //MISO output
  SPCR = (1<<SPE);       //enable SPI
```

```
  /************************** MAIN LOOP ***************************/
  for (;;) {}             //loop forever
}

ISR(INT0_vect) {
  // This vector is triggerd when PicoBlaze sends a ChipSel signal
  ADCSRA |= (1<<ADSC);   //start a single ADC conversion
}

ISR(ADC_vect) {
  SPDR = ADCH; //move the left-adjusted sampled value to SPI buffer
  SPI_SlaveReceive(); //wait for SPI and get the new idx
}
```

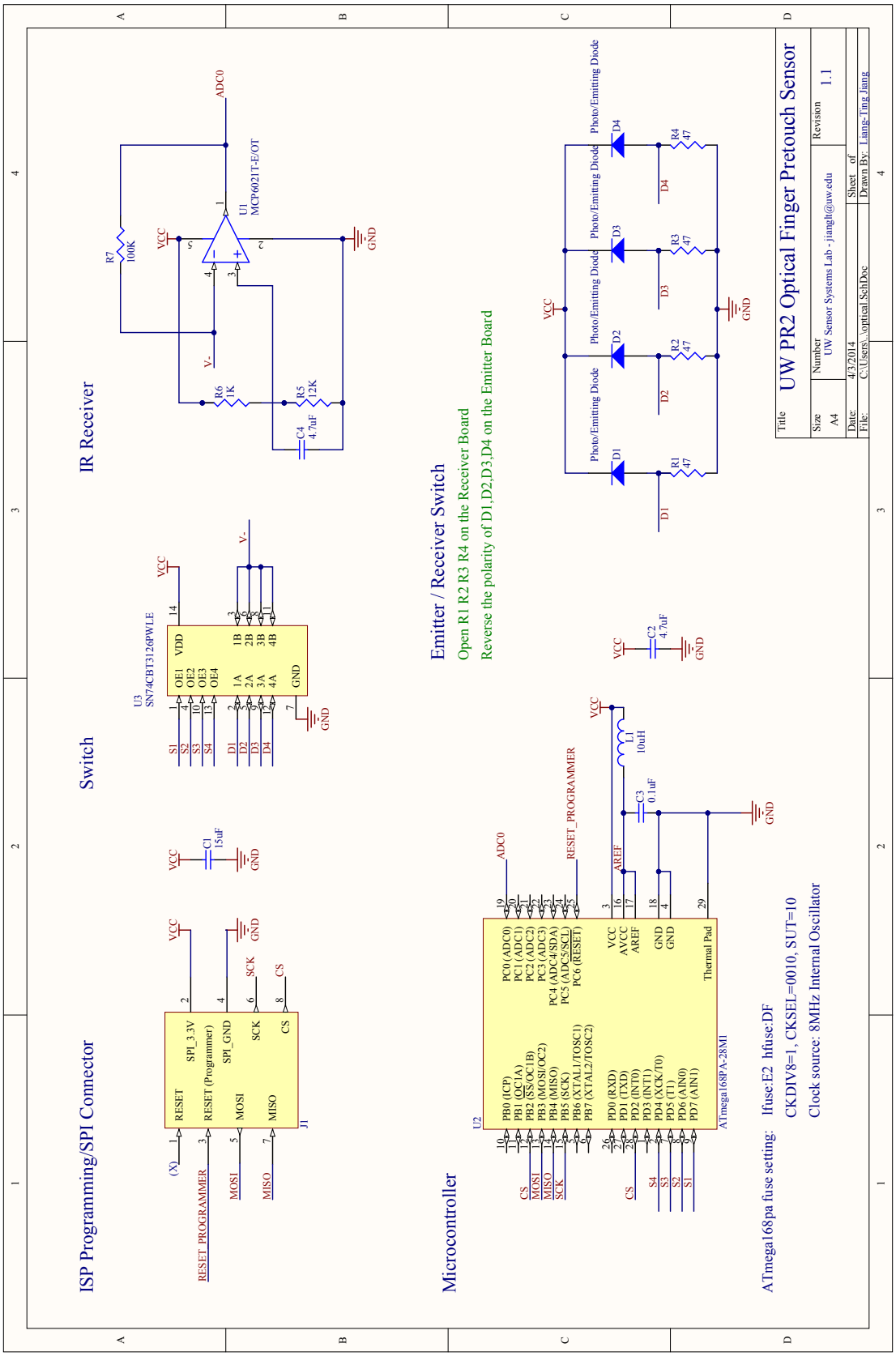
A.4 Seashell Effect Pretouch Sensor – PicoBlaze Firmware

```
; PicoBlaze firmware for the PR2 WG006 gripper to communicate with
; the PR2 seashell effect pretouch sensor. Serve as the SPI Master to
; receive data from the sensor's microcontroller, and also serve as the
; EtherCAT Slave for the ROS PR2 EtherCAT Master.
; @Liang-Ting Jiang 01-16-2014

CONSTANT SPI_CLOCK_DIVIDER, 3  ;0x3=3 -> 3,125 kHz SPI clock
CONSTANT DELAY_3US_COUNT, 12

; INITIALIZATION
init:
    load s1, SPI_CLOCK_DIVIDER
    output s1, SPI_CLOCK_REG
    ; Setup timerB to expire every 28us (35,715Hz)
    load s0, 1c ; 0.001ms * 28 = 0.028ms = 0x1c
    output s0, TIMER_B_COMPARE_REG
    load s0, 0
    output s0, TIMER_B_COUNTER_REG   ; reset Timer B counter to 0
    load s0, TIMER_OVERFLOW_FLAG
    output s0, TIMER_B_OVERFLOW_REG  ; clear overflow flag

; MAIN LOOP
main_loop:
    load s0, 0 ; reset index pointer to 0
    output s0, PRESSURE_INDEX_LOW_REG
    output s0, PRESSURE_INDEX_HIGH_REG
    ; The first value in pressure data is 32bit timestamp
    load   s0, PRESSURE_TIMESTAMP_LATCH_FLAG
    output s0, PRESSURE_CTRL_REG
    load   s0, 0
    ; copy timestamp to begining of pressure data buffer
    input  s0, PRESSURE_TIMESTAMP_REG_0
    output s0, PRESSURE_DATA_REG
    input  s0, PRESSURE_TIMESTAMP_REG_1
```

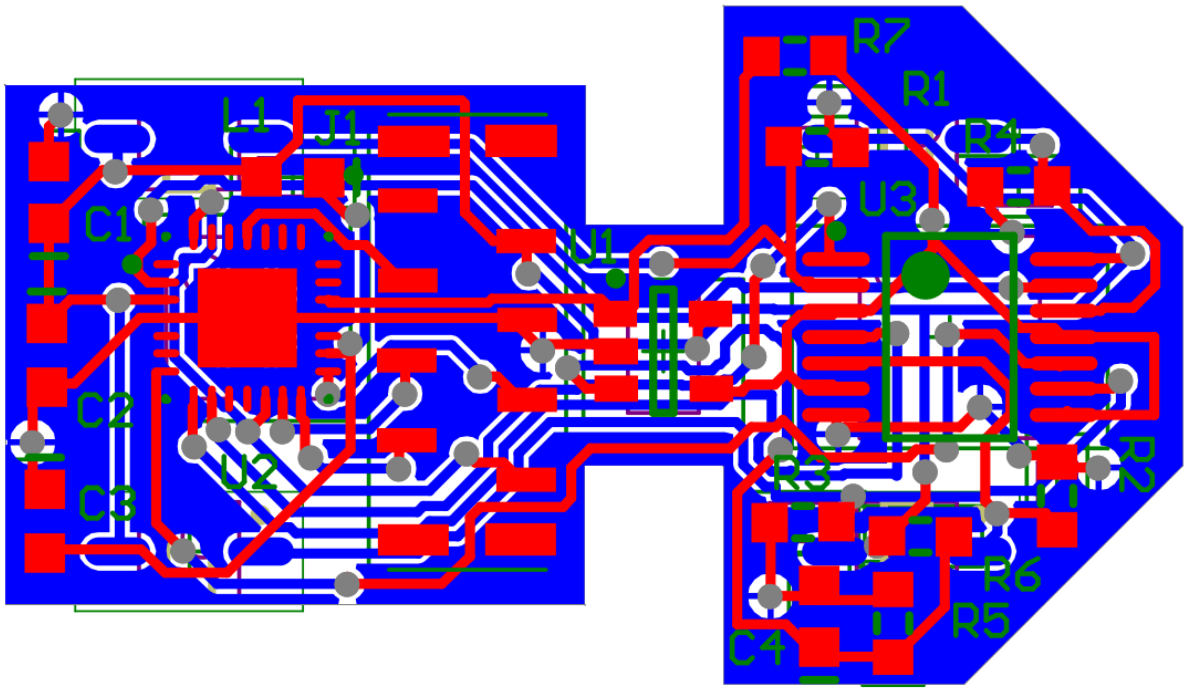
```
output s0, PRESSURE_DATA_REG
input  s0, PRESSURE_TIMESTAMP_REG_2
output s0, PRESSURE_DATA_REG
input  s0, PRESSURE_TIMESTAMP_REG_3
output s0, PRESSURE_DATA_REG
; Data buffer size: 512 bytes ; Flush every 254 iterations
; 4 (timestamps) + 254*2 (2 sensors) = 512 (bytes)
load s1, fe ; 0xfe=254
read_sensor_multiple_times:
    ; Wait for timerB to expire
    wait_for_timerB_overflow_loop:
        input s0, TIMER_B_OVERFLOW_REG
        test s0, TIMER_OVERFLOW_FLAG
        jump Z, wait_for_timerB_overflow_loop
    load s0, TIMER_OVERFLOW_FLAG ; Clear overflow flag
    output s0, TIMER_B_OVERFLOW_REG
    ; assert nChipSel1 and read from the first sensor (reference)
    load s0, SPI_ASSERT_CHIPSEL_1_FLAG
    output s0, SPI_CTRL_REG
    call read_sensor
    ; assert nChipSel2 and read from the second sensor (sensing)
    load s0, SPI_ASSERT_CHIPSEL_2_FLAG
    output s0, SPI_CTRL_REG
    call read_sensor
    sub s1, 1
    jump NZ, read_sensor_multiple_times
; New sensor data is ready, all we need to do is flip the buffer
; Make sure old buffer was copied out, before flipping data buffer
pressure_buf_ready_wait_loop:
    input s0, PRESSURE_CTRL_REG
    test s0, PRESSURE_BUSY_FLAG
    jump NZ, pressure_buf_ready_wait_loop
; Flag pressure data as being ready
load s0, PRESSURE_READY_FLAG
output s0, PRESSURE_CTRL_REG
jump main_loop
```

```
; Reads data from sensor into current location of output buffer
read_sensor:
    ; wait a bit for chipselect signal to propogate and ADC
    load s0, DELAY_3US_COUNT
    call delay_us
    ; read 1bytes of data from pressure sensor into buffer
    call spi_xfer_8
    ; put recieved data (s0) into pressure buffer
    output s0, PRESSURE_DATA_REG
    ; deassert chipselect and wait for bit for de-select to propogate
    load s0, SPI_DEASSERT_CHIPSEL_FLAG
    output s0, SPI_CTRL_REG
    load s0, DELAY_40US_COUNT
    call delay_us
    return

; Transfers 8bit over SPI, waits for transfer to complete,
; and returns recieved data
spi_xfer_8:
    ; put the current s2 value (idx value) in SPI bufferr before start
    output s2, SPI_DATA_REG
    ; wait for transfer to complete
    spi_xfer_busy_wait:
        input s0, SPI_CTRL_REG
        test s0, SPI_BUSY_FLAG
        jump NZ, spi_xfer_busy_wait
    ; put result of transfer into s0
    input s0, SPI_DATA_REG
    return

; Delays for a given number of milliseconds by running an idle loop
delay_us:
    sub s0, 1
    jump NZ, delay_us
    return
```

A.5 Optical Pretouch Sensor Board – Schematics



A.6 Optical Pretouch Sensor Board – Layout and Bill of Materials



Comment	Description	Designator	Footprint	LibRef	Quantity
15uF	0603 SMD Capacitor	C1	CAPC1608N	Cap Semi	1
4.7uF	0603 SMD Capacitor	C2, C4	CAPC1608N	Cap Semi	2
0.1uF	0603 SMD Capacitor	C3	CAPC1608N	Cap Semi	1
Photo/Emitting Diode	Emitting/Photo Diode 940nm Gullwing	D1, D2, D3, D4	Vishay VEMD/VSMB footprint	Diode	4
FFC 8-PIN Connector (1mm)	FFC 8-PIN Connector (1mm) (SMT)	J1	Hirose FFC connector 8-Pin	FFC Connector-8Pin-SMT-1mm	1
10uH	0603 SMD Inductor	L1	INDC1608N	Inductor	1
47	0603 SMD Resistor	R1, R2, R3, R4	RESC1608N	Res3	4
12K	0603 SMD Resistor	R5	RESC1608N	Res3	1
1K	0603 SMD Resistor	R6	RESC1608N	Res3	1
100K	0603 SMD Resistor	R7	RESC1608N	Res3	1
MCP6021T-E/OT	Rail-to-Rail Input/Output, 10 MHz Operational Amplifier, 5-Pin SOT-23, Extended Temperature, Tape and Reel	U1	SOT-23-OT5_N	MCP6021T-E/OT	1
ATmega168PA-28M1	8-Bit AVR Microcontroller with 8K Bytes of In-System Programmable Flash Memory	U2		ATmega168-28M1	1
SN74CBT3126PWLE	Quadruple FET Bus Swit	U3	PW014_N	SN74CBT3126PWLE	1

A.7 Optical Pretouch Sensor – AVR Microcontroller Firmware

```
/*********************************************************************
  Firmware for the PR2 optical sensor receiver board
  The onboard ADC is used to sample the amplified photodiode signal
  Serve as SPI Slave to send sampled data to SPI Master (PicoBlaze)
  microcontroller: Atmel AVR ATMega168(PA)/328(P)
  @Liang-Ting Jiang 01-16-2014
*********************************************************************/
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>

volatile char ch;
volatile char idx;
int idx_to_ch[] = {2,1,4,3};

char SPI_SlaveReceive(void) {
  // Wait for reception complete (until SPIF is set)
  while(!(SPSR & (1<<SPIF)));
  return SPDR;           //return the value received from Master
}

int main (void) {
  /****************** SWITCH SELECT OUTPUT PINS ******************/
  DDRD |= (1<<DDD4) | (1<<DDD5) | (1<<DDD6) | (1<<DDD7);
  /********************* EXTERNAL INTERRUPT **********************/
  DDRD &= ~(1<<DDD2);     //setup External interrupt 0 as Input (PD2)
  PORTD |= (1<<PD2);      //set PD2(INT0) to high (impedance)
  EIMSK |= (1<<INT0);     //enable INT0
  EICRA |= (1<<ISC01);    //trigger INT0 on falling edge
  /***************************** ADC *****************************/
  //using ADC0 -- No MUX values change required (default)
  //ADC prescaler=128 -- ADC clock rate  = 62.5KHz @ 8MHz clock rate
  ADCSRA |= (1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0);
  ADMUX  |= (1<<REFS0);  //set ADC reference to AVCC
```

```
  ADMUX  |= (1<<ADLAR);  //left adjust results to allow 8 bit reading
  ADCSRA |= (1<<ADEN);   //enable ADC
  ADCSRA |= (1<<ADIE);   //enable ADC Interrupt
  sei();                 //enable Global Interrupts
  /***************************** SPI ******************************/
  DDRB = (1<<DDB4);      //MISO output
  SPCR = (1<<SPE);       //enable SPI
  /********************** INITIAL VALUES ************************/
  ch = 1;
  idx = 2;
  /************************** MAIN LOOP ***************************/
  for (;;) {}             //loop forever
}

ISR(INT0_vect) {
  // This vector is triggerd when PicoBlaze sends a ChipSel signal
  // Before ADC, we send a signal to select the desired channel
  // The desired channel (ch) order: 1-2-3-4 (PD7->PD6->PD5->PD4)
  // The idx value receive the from SPI master: 4-3-2-1
  // However, new idx values are received during SPI after ADC
  // Therfore, ch should lag behind the idx by 1
  //   ch : 1 2 3 4 1 2 3 4   -->   ch : 1 2 3 4 1 2 3 4
  //   idx: 4 3 2 1 4 3 2 1   -->   idx:   4 3 2 1 4 3 2 1
  ch = idx_to_ch[idx-1];    //the desired channel to sample
  PORTD |= _BV(8-ch);       //send HIGH to select the switch channel
  _delay_us(1);             //wait for switching
  ADCSRA |= (1 << ADSC);    //single ADC conversion
  _delay_us(3);             //wait a bit for ADC to finish
  PORTD &= ~_BV(8-ch);      //de-select current channel
  idx = SPI_SlaveReceive(); //wait for SPI and get the new idx
}

ISR(ADC_vect) {
  SPDR = ADCH; //move the left-adjusted sampled value to SPI buffer
}
```

A.8 Optical Pretouch Sensor – PicoBlaze Firmware

```
; PicoBlaze firmware for the PR2 WG006 gripper to communicate with
; the PR2 optical finger sensor. Serve as the SPI Master to receive data
; from the sensor's microcontroller, and also serve as the EtherCAT
; Slave for the ROS PR2 EtherCAT Master.
; @Liang-Ting Jiang 01-16-2014

; CONSTANTS
CONSTANT SPI_CLOCK_DIVIDER, 1E  ;0x1E=30 -> 403.2kHz SPI clock
CONSTANT DELAY_40US_COUNT, F9

; INITIALIZATION
init:
    load s1, SPI_CLOCK_DIVIDER
    output s1, SPI_CLOCK_REG
    ; Setup timerA to expire every 0.5ms (2,000Hz) (V) (ADC prescalar=4)
    ; 4 ch for 1 reading, so the actual reading rate is 2000/4 = 500 Hz
    load s0, 5 ; 0.1ms * 5 = 0.5ms    5 = 0x05
    output s0, TIMER_A_COMPARE_REG
    load s0, 0
    output s0, TIMER_A_COUNTER_REG   ; reset Timer B counter to 0
    load s0, TIMER_OVERFLOW_FLAG
    output s0, TIMER_A_OVERFLOW_REG  ; clear overflow flag

; MAIN LOOP
main_loop:
    load s0, 0 ; reset index pointer to 0
    output s0, PRESSURE_INDEX_LOW_REG
    output s0, PRESSURE_INDEX_HIGH_REG
    ; The first value in pressure data is 32bit timestamp
    load   s0, PRESSURE_TIMESTAMP_LATCH_FLAG
    output s0, PRESSURE_CTRL_REG
    load   s0, 0
    ; copy timestamp to begining of pressure data buffer
    input  s0, PRESSURE_TIMESTAMP_REG_0
```

```
output s0, PRESSURE_DATA_REG
input  s0, PRESSURE_TIMESTAMP_REG_1
output s0, PRESSURE_DATA_REG
input  s0, PRESSURE_TIMESTAMP_REG_2
output s0, PRESSURE_DATA_REG
input  s0, PRESSURE_TIMESTAMP_REG_3
output s0, PRESSURE_DATA_REG
; Flush the data buffer after every 4 sensor reads (4 channels)
load s2, 4;  4 cycles
read_sensor_multiple_times:
    ; Wait for timerA to expire
    wait_for_timerA_overflow_loop:
        input s0, TIMER_A_OVERFLOW_REG
        test s0, TIMER_OVERFLOW_FLAG
        jump Z, wait_for_timerA_overflow_loop
    load s0, TIMER_OVERFLOW_FLAG ; Clear overflow flag
    output s0, TIMER_A_OVERFLOW_REG
    ; Read from the first sensor (IR receiver)
    ; assert nChipSel1
    load s0, SPI_ASSERT_CHIPSEL_1_FLAG
    output s0, SPI_CTRL_REG
    ; use sub-routine to preform actual read of data
    ; also de-assert chipselect before returning
    call read_sensor
    sub s2, 1
    jump NZ, read_sensor_multiple_times
; New sensor data is ready, all we need to do is flip the buffer
; Make sure old buffer was copied out, before flipping data buffer
pressure_buf_ready_wait_loop:
    input s0, PRESSURE_CTRL_REG
    test s0, PRESSURE_BUSY_FLAG
    jump NZ, pressure_buf_ready_wait_loop
; Flag pressure data as being ready
load s0, PRESSURE_READY_FLAG
output s0, PRESSURE_CTRL_REG
jump main_loop
```

```
; Reads data from sensor into current location of output buffer
read_sensor:
    ; wait a bit for chipselect signal to propogate and ADC
    load s0, DELAY_40US_COUNT
    call delay_us
    ; read 1bytes of data from pressure sensor into buffer
    call spi_xfer_8
    ; put recieved data (s0) into pressure buffer
    output s0, PRESSURE_DATA_REG
    ; deassert chipselect and wait for bit for de-select to propogate
    load s0, SPI_DEASSERT_CHIPSEL_FLAG
    output s0, SPI_CTRL_REG
    load s0, DELAY_40US_COUNT
    call delay_us
    return

; Transfers 8bit over SPI, waits for transfer to complete,
; and returns recieved data
spi_xfer_8:
    ; put the current s2 value (idx value) in SPI bufferr before start
    output s2, SPI_DATA_REG
    ; wait for transfer to complete
    spi_xfer_busy_wait:
        input s0, SPI_CTRL_REG
        test s0, SPI_BUSY_FLAG
        jump NZ, spi_xfer_busy_wait
    ; put result of transfer into s0
    input s0, SPI_DATA_REG
    return

; Delays for a given number of milliseconds by running an idle loop
delay_us:
    sub s0, 1
    jump NZ, delay_us
    return
```