Expanding the Depth Cloud: How to Intelligently Improve Grasping
Senior Thesis

Matt Mullen
mjollnir@cs.uw.edu

December 16, 2011

Abstract

Automated grasping is an area of robotics that is of vital importance in robotics. While it is a simple task for humans, it is difficult for robots to the same tasks. Part of the problem with this is the sensors utilized. Many sensors do not accurately register large regions of objects, leaving the robot to work with a small part of the object to give it an incomplete and often misleading picture of what the object actually is. Fortunately, there are many additional sources of information that we can use to supplement the depth clouds provided by the sensors. We take a look at using both RGB images and a pre-touch sensor to intelligently gather additional information needed to determine how to grasp an object.
Contents

1 Introduction 3
2 Grasping 3
3 The Soda Bottle Problem 4
4 Expanding the Point Cloud 5
5 Pretouch Sensors 6
   5.1 Seashell Effect Sensor ....................................... 7
6 Determining Where to Probe 8
   6.1 Oracle ............................................................... 9
7 Solution for the soda bottle problem 9
8 Results 12
9 Conclusion 13
10 Acknowledgments 13
1 Introduction

Automated grasping is an area of vital importance in the realm of robotics, especially in the realm of personal assistance. Tasks such as object retrieval and manipulation require the robot to have a firm idea of how to grasp the related object. This is a task that humans perform almost flawlessly from a young age, and yet robots traditionally have a hard time accomplishing. While much of this can be contributed to the mechanical limitations of robotic graspers, the sensors that are used to detect objects are imperfect as well.

Much of the difficulty in grasping comes from the way in which the robot can gather information on the subject. While humans have highly mobile hands to interact with objects, developed intuition about how to interact with objects both known and foreign, as well as highly sensitive and accurate optics, robots are mostly limited to 2D images, much more highly limited grasping implements, a limited database of ways to grasp objects, as well as imperfect depth sensors. Working within these challenges is difficult when attempting to grasp even simple objects.

One particularly problematic area is when we can only receive partial information about an object we are trying to grasp. Whether it is because we only have one side of the object available, or because out sensors do not detect it properly, the situation becomes worse because we cannot clearly tell when we only have partial information. Additionally, when trying to receive extra information, we may have to interact with the object, which has the possibility of disrupting it in some way, perhaps by moving it, physically altering it, or some other change. The fact that the robot is not sure of the overall geometry of the object further compounds the issue. Clearly, we want a non-disruptive way to both verify if there we have missing information of the object, and to obtain as much, if not all, of the missing information as possible.

More so, we need make sure the information that we obtain is accurate. For instance, when looking down at an object on a table, it is important that we be able to distinguish this object from others on the table.

This paper discusses a situation in which we have partial, but not complete, depth cloud information of an object that we wish to grasp. First, it covers the grasping problem itself, then introduces the "Soda Bottle Problem". Next, it proposes some methods in which we may get more information of the object, using both pretouch sensors and an RGB image of the data in addition to the known point cloud. Lastly, it describes the process in which we determine the extra information of the object, discussing the results of such using this process.

2 Grasping

The grasping problem can be broken up into several discrete steps:

1. Isolate the object to grasp
2. Determine how to grasp the object
3. Maneuver the grasping apparatus to grasp the object
To isolate the object that we wish to grasp, we utilize a sensor to create a giant point cloud of the objects in front of it. From here, we determine the plane that all of the common objects are sitting on, and remove all points in the point cloud that are on that plane. From here, we have several disjoint point clouds. We can assume, for simplicity, that we identify one of these as the object we wish to grasp, then center the vision of the robot’s camera on that object.

Determining how to grasp the object is the most algorithmically complex task. From the point cloud of the object, a 3D model is constructed from the point cloud. This 3D model is then compared to models of objects that the computer knows how to grasp, so that it can utilize how it grasps those objects to attempt to grasp the object it front of it.

Lastly, we maneuver the grasper to attempt to grasp the object. The success of our model here is demonstrated by either a successful or an unsuccessful grasp. While increasing the aptitude of the mechanical grasper itself can increase the likelihood of a successful grasp, increasing the capability of identifying and determining how to grasp an object can substantially increase the success rate of grasping a given item. This is especially useful in the case of the Soda Bottle Problem.

3 The Soda Bottle Problem

While the sensors used in creating depth clouds are highly accurate in many cases, in some cases they do not create accurate readings at all. For instance, in the laser scanner used by the Kinect sensor, it cannot accurately calculate the distance from the scanner to transparent or translucent surfaces. This is especially evident in soda bottles that have large portions of their surface area covered by translucent plastic. As seen in figure 2, the only portion of the bottle that is accurately scanned by the Kinect are the plastic wrapper and the bottle cap. In fact, they are separate enough that even though they are part of the same object, they are registered as two different objects.

These type of problem can occur in more situations than with soda bottles and
The Soda Bottle Problem is any situation in which we have incomplete information of the point cloud of the object that leads to misrepresentation of the object.

Note that this problem is different from having no point cloud information for the object at all.

The Soda Bottle Problem is a problem that can show up in many different ways, depending on the sensor. This problem can be difficult to solve, and can lead to improper grasps on the object, as the partial information of the object may be misleading. For instance, if only half of an object is covered by a point cloud, then it will prove impossible to find a proper center of gravity of the object, and may lead to trying to grasp an object in an impossible manner. In order to be certain of grasping the object properly, we need to add the missing information of the object to its point cloud, so that we may more accurately model the object.

4 Expanding the Point Cloud

There are a large number of ways to increase the amount of point cloud information available for a given object. One of the simplest ways to add extra information is to merge together two point clouds of the same object. This involves physically moving the sensor to a different location, then getting new point cloud information. As we constrained the grasping problem to looking directly at the object that we desire to be grasped, we can be sure that the object we are getting point cloud information from is the same object. This proves especially useful, as we can receive information about the back end of the object that would otherwise be unutilized. It also allows us to correct our grasps in the case of unusual angels of the object.

Unfortunately, this process does little to help out with the Soda Bottle Problem. While the additional information is beneficial, it will not help in gaining information.
about the surfaces that the sensor cannot already detect, which is the information we are most interested in.

Another potential solution is to expand the point cloud probabilistically. Using the known point cloud, we can extend the surface of the object slightly in order to gain more potential information. The main problem with this (and many other methods of extrapolating the point cloud) is that there is large room for error, as well as no implicit way to verify the correctness of the new information. While in many cases of smoothly-shaped objects, we are able to gain more information, often times with the Soda Bottle Problem, we have a very small portion of information on the object itself. In figure 2, the object of interest in the center of the image has very little of its surface represented by a point cloud, and it doesn’t hint at the sudden change in shape directly above and below that region. Without a way to verify the information, this does not gain us very much useful information.

One other method is using a pretouch sensor to physically probe the object, in order to determine where the object physically is. This method has a lower chance for error, as we are physically verifying that an object is where we are probing. This is the basis of the method that we are using to solve the Soda Bottle Problem.

5 Pretouch Sensors

Pretouch sensors are a group of sensors that are designed to determine when it is about to touch an object, as opposed to using a sensor to detect contact or using some sort of vision process. There are a large number of motivations to use a pretouch sensor, ranging from additional sensor data to robot-to-human handoff scenarios[4][5][6]. One of the more common pretouch sensors is the e-field sensor[4], which is used not only in pretouch scenarios, but is useful in handoff situations, and the electric field of the object changes when a human grasps or stops grasping the object of interest.

Pretouch sensors can prove especially useful when dealing with objects that grasping is either unfeasible or impossible. For instance, when grabbing an unknown object, many graspers will tighten until the object provides some level of force back on the grasper. While this works well for many items that are grasped, for food items especially this proves problematic, as the item becomes rendered usable from the force applied against it.

In addition, the pretouch sensors do not work so well when interacting with unknown items, as it is necessary to know how to be able to interact with the object. When an object is obscured in some manner, it becomes more difficult to fit a known grasping schema to interact with the object, and can lead to problematic interactions. When using pretouch, all that is necessary is knowing where part of the object is located. This advantage allows us to utilize it in situations where we want to learn how to grasp an object, but it potentially has areas that would be problematic to grasp, such as a wine glass. With a normal grasp, we may shatter the more fragile part of the glass, whereas we want to actually grasp it on the much more sturdy stem.

The current e-field pretouch sensors fail for objects that are not similar to objects they are preprogrammed to grasp, for instance in the case of a difference in the dimensions of the object, and when presented with multiple objects. Other pretouch sensors,
such as the optical-based pretouch sensor described in Hsiao et al. [1], fail when for large categories of objects (e.g., the vision pretouch sensor fails for highly specular metallic objects).

5.1 Seashell Effect Sensor

One of the more recent pretouch sensors developed is a sensor that utilizes the so-called "seashell effect" [2]. The sensor utilizes two microphones, one on the surface of the sensor, and one inside a resonant cavity in the sensor. As the sensor approaches the surface of an object, there will be a difference in the sound on the surface and inside the resonant cavity of the sensor. Using this, one can use the sensor to detect a wide range of objects without displacing them or breaking them. For instance, Jiang cites that he was able to detect contact for a folded paper box without crushing it, while the pressure sensor on the PR2 that he was using was unable to [2].

The advantage of this is that the sensor works with a wide variety of object, beyond that which most other pretouch sensors are able to do. This is ideal for the Soda Bottle Problem, as we want to be able to expand the point cloud of arbitrary objects.

In an upcoming paper, Jiang discusses a method of using the seashell sensor to expand a point cloud for an ordinary bottle. Given a partial point cloud for the bottle, one can assume that it’s shape is more or less cylindrical, and extrapolate what the point cloud should look like from the side not facing the camera. Then, the seashell sensor approaches the bottle along the calculated normal for that point, until the sensor detects that it’s hit something (presumably the bottle). Then, that point and the region around it is in the point cloud. The end result is that you have a much more accurate representation of the object, so that modeling the object for grasping is a lot more accurate. Using this method, the PR2 used can use a more accurate and robust grasp on the object.

Of course, this method only works on one particular shape of object. While it works for the titular object of the Soda Bottle Problem, it does not work for a large number of objects. The main problem with using a pretouch sensor is that, given only a point cloud, it is difficult to suppose where next to probe with the sensor. While it may be possible to probe at locations surrounding the existing point cloud, this is both slow and prone to inaccuracies, as there is little way to immediately differentiate it from the surrounding objects. The information gained is also not always the most useful information. For instance, with the soda bottle in Figure 2, we only have depth cloud information on the label, so when we assume the object is cylindrical, we do not gain.

<table>
<thead>
<tr>
<th>Objects</th>
<th>Pressure</th>
<th>Seashell Pretouch</th>
</tr>
</thead>
<tbody>
<tr>
<td>cookie box</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>disposable cup</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>folded paper box</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>folded aluminum foil box</td>
<td>x</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 1: Results of Experiments for Seashell Pretouch Sensor [2]
information about the translucent parts above and below the label.

6 Determining Where to Probe

Without a way to predict where the object should be located, or if there even are parts of the object that are not covered by the point cloud, then it becomes difficult to accurately expand the point cloud. Fortunately, there are a number of sources that are not currently used that can help provide guidance for where to probe.

One source of information that is underutilized in grasping is the RGB image of the object that we wish to grasp. While depth cloud information may not be directly gleaned from the image, we can directly correspond pixels of an RGB image with known depth information from a sensor. This makes corresponding points on an RGB image with points on the surface of an object a rather simple task.

One source of information that can be gleaned is from color segmentation on the image. When broken down with a color segmentation algorithm, we break down the image into regions of relatively similar color. Within these regions, it is highly likely that it is the same object throughout the region. That is not to say that the entire object is contained in a single region.

If a point in the depth cloud is contained in a color segment, then it is highly likely that the rest of the color segment is part of the object. Using this information, from a single point, we have a sizable amount of area to probe for depth cloud points of the object.

Color Segmentation only gains us useful information in the case that a point already happens to fall inside it. We can use edge detection on the image in order to determine likely edges of the the object. If we detect an edge of the image that is close to one of the edges of the depth cloud, then it is likely that it is a strong edge in the object. From those edges, we can find neighboring color segments, either contained inside the edges or adjacent to them, in order to intelligently expand the depth cloud.

One way to approach this would be detect edges on the RGB image and then compare them to the depth cloud information. A more robust method would be to use the RGBd (RGB image plus depth information) edge detection found in the Oracle Project.
6.1 Oracle

The Oracle project is an old Intel Labs project that contains a variety of algorithms that work on RGBd images. Recently, it has been used for the purpose of object recognition, of either categories of objects of specific objects[3]. One feature that is used in order to categorize this is RGBd edge detection, specifically from a top-down view with a Kinect-style 3D camera.

Here, instead of running edge detection and comparing contours to where the point cloud of the original object is cutoff, we instead have an algorithm that implicitly relates the two. This allows us to more accurately determine areas where we wish to expand the point cloud. First, any area inside the detected edge is highly likely to be part of the object of interest. Secondly, any part of a color segment that is adjacent to the detected edge is likely to be part of the object. With this information, we can intelligently determine areas in which we want to use the pretouch sensor to test for points on the object.

It is worth noting here that the Oracle project originally assumed that the sensor used to detect the object would be able to have depth information about the entirety of the object, that is, there is accurate depth information about the entire object. While this is unfortunate, it is a symptom of the fact that we are trying to gain additional information about the object. Due to the nature of the Soda Bottle Problem, we cannot accurately calculate the outline from this. However, as we do have depth information of part of the object, we can relate the edges of the depth cloud with edges in the image, which was the goal to begin with.

7 Solution for the soda bottle problem

Consider again the originally stated Sode Bottle Problem.

The Soda Bottle Problem is any situation is which we have
Figure 5: Here we have an object we would potentially like to grasp. While RGB Edge detection outlines the lip of the bowl, it is not the only edge detected. The object mask provided by Oracle’s RGBd edge detection, however, we have selected the entire bowl at the most useful edge.

One of the primary difficulties in solving this problem is not only determining how to expand to the point cloud, but more importantly how to correct for potential errors. Without a sound way to error check, there is a large danger of adding false positives to the depth cloud.

By using a pretouch sensor alone, we are both not certain where to probe the object, and not certain if what we probed is part of the actual object. Using image analysis, we can determine regions where we believe the object should reside, but there is no way to be certain.

By combining the two methods, we arrive at a relatively robust solution. We can use the image analysis to determine where we believe the object should reside, and then we can use the pretouch sensor to determine if the object is actually there, and where it resides if it finds it. By using oracle to detect the edge of the point cloud and the object, and using a robust seashell effect pretouch sensor, we can do so with even more accuracy.
There are still some problems with this method. For one, if the image analysis proves faulty, and we do contact an object, there is the potential that we either accidently probe another object, or the surface on which the object resides. Correcting for the error of probing the surface is relatively simple, as we know the plane that the surface lies on, from when we first determine the depth cloud of the object of interest. Correcting for intersecting with other object is much less immediate.

Fortunately, we can use the depth cloud information of the original object. If we probe at an unknown region that is relatively close to known depth cloud information, then the three-space coordinates of the object should be relatively close to the three-space coordinates of a nearby point. We can compare the x,y,z coordinates of the point to all the points in the depth cloud, and only if it is within a tolerance of another point in the existing point cloud, then we can confidently add it to the point cloud.

Another problem is that we are not entirely certain of what angle to approach the object at. Ideally, we approach the potential region of the object along its normal. While the seashell sensor was able to do so when expanding the point cloud of a bottle, it was only able to do so by presupposing the general shape of the object; this method would not work in the general case.

As our solution to false positives is to only check for points that are close to existing points in the object’s point cloud, we can use this to extrapolate likely normals as well. From the coordinates in the point cloud, we can calculate the normal of the object at any given point. Once we have done so, we can extrapolate the likely normal for the point that we are probing.

With all of the above tools, we now have an intelligent method to help solve the Soda Bottle Problem.
Algorithm 1 Method for Intelligently Extending the Depth Cloud

Get Depth Cloud Information and RGBd image of Object
Calculate the plane of the surface it is sitting on, and remove those points
Run Oracle Edge Detection on Image
Run Watershed Color Segmentation on Image
Choose a color segment adjacent to the depth cloud to explore

while There are color segments to explore do
  Choose a color segment next to a validated region
  while There chosen segment has not been fully explored do
    Choose a point in the color segment that is sufficiently distant from other explored points
    Extrapolate from the existing point cloud what it’s x,y,z coordinates and normal should be
    Using the extrapolated information, probe the point along the extrapolated normal
    if The probe in the color segment is outside of a tolerance in predicted x,y,z coordinates OR the point is on the plane of the surface the object is sitting on
      Mark the region as invalid
    else
      Add the probed region to the depth cloud
    end if
  end while
  if The entire region was valid then
    Mark the adjacent color segments as explorable.
  end if
end while

Going back to the Soda Bottle Problem, this algorithm provides a robust method for increasing the amount of depth cloud coverage we have of the object. A seashell effect pre-touch sensor is capable to detecting most objects that could be grasped, and the image analysis allows for us to intelligently determine where to use the sensor.

8 Results

As mentioned earlier, the RGBd edge detection implemented in the Oracle project provides a very powerful form of edge detection. Unlike traditional edge detection, in which we have several contours that outline all of the potential edges of an object, by utilizing depth information, we have a much strong initial picture of what an object looks like. Additionally, the oracle edge detection is not as sensitive to sudden changes in texture. The bowl in Figure 5 is an excellent example of this. The texture on the inside of the bowl proves confusing for a Canny edge detection algorithm, which reports multiple edges on the inside of the bowl, whereas Oracle provides an accurate report of the actual edge of the bowl.
The seashell effect pretouch sensor proves able to detect multiple kinds of objects, as seen in Table 1. It is able to interact with objects that are both fragile and difficult for other sensors to detect. In the case of translucent plastic, the pretouch sensor is able to detect when it is adjacent to the material, where both a Kinect sensor is unable to give accurate depth readings and an e-field pretouch sensor is unable to reliably detect when it is adjacent to it.

A Watershed color segmentation algorithm is able to sort of color segments of an object, and in such a manner as we can fill out such a region with probes from the seashell sensor.

Using this, we are able to accurately detect a much larger amount of objects than with the Kinect sensor currently utilized for grasping problems by the PR2. This allows us to increase the amount of information for most objects.

Currently, the types of objects that this method proves difficult to gain large amount of information are objects with sharp edges in the objects surface. For instance, consider a regular octahedron, where we only have depth cloud information of the upper-four faces. Due to the information we have available, the algorithm would predict that the shape would continue in a pyramidal-shape, but then the sensor would potentially not detect the object, or if it did, the coordinates would be off, so the point added would be discarded. This is an unfortunate consequence of the algorithm utilized, and cannot be accurately compensated for.

9 Conclusion

While large amounts of information can be gained from many instances of the Soda Bottle Problem, as the results section shows, there are still cases where we cannot gain information about the entire object. That said, the algorithm presented gains large amounts of useful information that can be utilized in grasping.

The information gained this way utilizes large amounts on information that has historically been underutilized, such as the RGB images, in addition to utilizing pretouch sensors that can accurately determine the location of objects without disrupting them. As pretouch sensors grow in capability, there may be less of a need to utilize image processing in order to intelligently determine where we may want information, although such information is undoubtedly useful, at least for initial guesses as to where more information about the object may lie.

One of the more important aspects of the proposed algorithm is that it does not depend on missing information existing, rather it uses intelligent error-checking to determine if any proposed new information is falicious. While it does lead to false-negatives, it is currently necessary to be useful in most situations.

Overall, this approach leads to more robust grasps, especially on the titular bottles of soda, where previously only small amounts of information were available, due to the nature of the material.
10 Acknowledgments

Liang-Ting Jiang, for his work in the pretouch sensor, and for helping me utilize it in the project.

Peter Brook, for his overall help on the project, and specifically with utilizing many of the visual algorithms used.

Kevin Lai and Jinna Lei, for their work on the Oracle edge detection algorithm.

Joshua Smith, for advising me in my senior year on the overall project.
References


