Generating and Learning from 3D Models of Objects through Interactions

by
Kiana Alcala, Kathryn Baldauf, Aylish Wrench

Abstract

For our project, we plan to implement code on the robotic arm that will allow it to create a 3D representation of an object. As of right now, the robot can only construct a 2D view of an object which limits the robot's capabilities, such as its grasping. We want the robot to eventually be able to recognize an object from any angle and pick a path that will allow it to have a greater chance of successfully lifting the object. In order to do this, we plan to have the robot repeatedly pick up the objects, each time manipulating its grasp and movement of the object in order to construct a 3D model. We then hope to store data about the object so that the robot will be able to detect similar objects and thus perform a better grasp in a more timely fashion. Ultimately, we will focus on the robot's ability to grasp an object in a manner that will allow for the highest chance of success to allow the robot to better assist humans.

1. Introduction

At a young age, children have no sense of object permanence and little coordination. This can make actions such as grasping difficult. Robots have similar problems. In the same way that children learn to recognize objects and determine the best way to grasp them, we want the robot to learn through interaction. Through their development, children learn how to approach objects by experimenting with toys. This not only improves their knowledge of objects but solidifies their motor development, allowing them to get better at it with more practice. We plan to use machine learning in order to allow the robot to execute actions similar to those used by small children.

There are a series of problems that we need to address. First, the robot needs to be able to determine whether or not a completed grasp was successful. We then need to teach the robot how to best move an object upon each grasping it in order to construct a 3D model. We plan to do this by implementing a volumetric next best view algorithm that will allow for the robot to choose which way to manipulate the object. Once the robot is able to successfully grasp the object and create a 3D representation, we hope to record characteristics about the object and store them. We can later use these stored characteristics when working with other objects to have the robot recognize the best grasp for a particular object as well as whether a current object has previously been explored or is unknown.

2. Related Works

The theory that children learn by interacting with objects is one that has existed for years in the field of psychology. Jean Piaget [1] discusses children's learning through

sensorimotor activities in his book, *The Origins of Intelligence in Children*, originally published in French in 1936 and later published in English in 1952. Piaget divides a child's development from birth to the age of two into six "substages." According to Piaget, a child begins to interact with objects in its environment in the third substage which occurs sometime after the child reaches four months old. In this substage, the child performs an action with their body (usually accidently) which has a visible effect on its environment. The child not only notices this change, but then tries to recreate it.

A child's interest in objects in its environments continues into the next substage, which begins around eight or nine months. However, instead of reacting to changes in its environment, in substage four the child actively begins experimenting with objects. By performing different actions on an object such as moving or shaking, the child gains a better understanding of the object's properties and abilities. Piaget's observations demonstrate the importance of sensorimotor activities in human development. We believe that incorporating both visual and motor functions to learn about an object could prove to be effective in the development of robotic intelligence. By picking up an object, the robot can learn more than it would by solely looking at it. For example, the robot can learn what grasps are successful for picking up an object. In addition, it can see areas of the object that were obscured previously. More importantly, the robot can accomplish this with less assistance from humans.

Our main goal for our project is to get the robot to combine visual and motor actions to create a 3D model of an object. This is similar to work done by Krainin et al. [2]. They created a system which allows a robot to create a 3D surface model by moving an object into its view. In their system, the robot uses a volumetric information driven "next best view algorithm" to determine how it must pick up and move the object to achieve the view that will give it the most information. Once the robot is finished with this view, it places the object back on the table. The robot cleans up the generated image by removing any points located on or near its hand. It then uses the next best view algorithm again to determine how it should re-grasp and move the object. This allows the robot to make the model more complete by cleaning up holes created by its manipulator during the initial grasp or by allowing it to see areas which it could not in the previous view. Due to its effectiveness in creating 3D models, we plan to implement the system created by Krainin et al. on our own robot.

3. Technical Approach

For our project, we will be developing and testing on the Segway base robots located in the Building Wide Intelligence (BWI) laboratory at the University of Texas at Austin. We will be using an xtion camera sensor, placed several feet above the base of the robot, angled slightly down in order to view the attached robot arm. This camera returns a

3D point cloud of the image it sees, a necessary condition for creating color accurate models with depth for our project. Our project will use the Kinova Mico robotic arm mounted in front of the laptop on the base of our robot. This arm has six degrees of freedom and two end effectors, another necessary component in order to grasp and manipulate our desired object.

3.1 Lifting the object in view of the camera

In order to create a 3D model of an object, we must first successfully lift the object in front of the camera sensor. To lift the object, we will be taking advantage of pre-existing code in the Building Wide Intelligence codebase, specifically the grasp and lift verification actions located in the segbot_arm_manipulation package. The agile grasp action will allow us to grasp the object in a legal position based on the calculation of Darboux's Box, a combination of the normal and two principle curvatures for a given point, for two points parallel to each other. From there the lift action will be called, given a predetermined location to lift the desired object to. In our project, we will first use this action to move the arm and object out of the view of the camera. This will allow the lifting action to compute various heuristics to determine if the correct object was lifted. If the object was successfully grasped and lifted, we will then move the arm and object together in front of the camera. Given this end location in view of the sensor, we will compute the point cloud associated with the arm and end effectors. This will be used later for construction of the model.

3.2 Construction of the 3D model

For the actual construction of the 3D model, we intend to roughly follow the algorithm outlined in Krainin et. al's work [2]. In their paper, they outline a statistical approach to finding the most efficient way to manipulate an object using a volumetric and information driven next best view algorithm given the current point cloud of the object as well as the position and joint angles of the robotic arm. This algorithm reasons about which regions of the current model of the object are determined to be the most uncertain. The "unknown" areas of the model are of particular interest in this paper, as these areas could correspond to an

Algorithm 1 Next best view of a grasped object

procedure SELECTVIEW($\mathcal{V}, T_{hand}, \theta$)
  $\mathcal{S}_{obj}$ = ExtractMesh($\mathcal{V}$)
  $\mathcal{S}_{hand}$ = GetHandModel(T_{hand}, θ)
  x_{obj} = GetObjectCenter($\mathcal{S}_{obj}$)
  $dirs$ = GetPossibleViewingDirections()
  for dir **in** $dirs$ **do**
    $range, roll, cost$ = SelectFreeDOFs(dir,x_{obj})
    if $range, roll \neq \emptyset$ **then**
      $x_{cam} = x_{obj} - range * dir$
      $T_{cam} = (x_{cam}, dir, roll)$
      q = GetPoseQuality(T_{cam}, $\mathcal{S}_{obj}$, $\mathcal{S}_{hand}$)
      $score = q - \alpha_{cost} * cost$
      if $q \geq t_q$ and $score > score^*$ **then**
        $T^*_{cam}, q^*, score^* = T_{cam}, q, score$
  return T^*_{cam}, q^*

Figure 1. Next Best View algorithm from Krainin et. al's paper

undiscovered, visible part of the object. As the object is reconstructed, a mesh is created with vertices corresponding to the certainty of each point in the model. This is useful for both the calculation of the unknown or empty spaces as well as in determining how to later fill in potential holes. The algorithm outlined in their article is pictured in Figure 1. Using the previously computed position of the robotic arm and fingers, the algorithm subtracts any points within some distance of arm, minimizing error in the constructed model. We will make necessary adjustments for our project set up as well as make use of pre-existing functions in the point cloud library for subtracting points near the hand from the model. Figure 2 depicts an example image of a 3D model constructed using the described algorithm with red sections corresponding to uncertain areas of the model. As the points from the hand are subtracted, holes are created in the model. In order to fill in these gaps, the paper

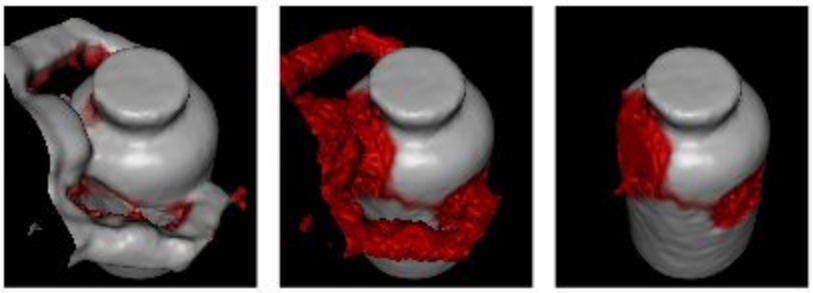

Figure 2. Example creation of a 3D model

proposes a second grasp may be necessary. The object is placed back in the starting location, in our case, a table. For the purposes of our project, we will always assume the desired object is the only object on the table. The grasping and lifting procedure outlined in section 3.1 is then repeated. A novel grasp for the object will be guaranteed as the object will not be placed in the exact same position and orientation as it began in. The algorithm outlined previously in this section will then be repeated. When the points in the reconstructed model are relatively certain, the reconstruction process is completed. From here, we will store this model for later use in our project with machine learning and recognition. The reconstruction process will be implemented as a ROS action to allow interruption and frequent feedback throughout the construction process. This will be useful in real time use of this code in order to perhaps only retrieve a partial view of the object or interrupt the action in the case of an error.

3.3 Machine Learning appropriate grasps

For our project, we will focus on supervised machine learning of the ideal grasp of a 3D modeled object. In order to do this, we will again make use of the grasping and lifting actions code in the Building Wide Intelligence codebase. We will modify the ROS action goal

for the method of grasping to “random” to allow the robot to explore various grasps each time. The robot will then attempt to grasp and lift the object enough times to explore each possible grasp at least twice. Since the lifting action returns a verification indicator, we will use this to determine which grasps were actually successful and which failed. We will then construct a training set given the previously saved constructed model of the object and the information about the completed grasps, including failure cases. We intend to use seventy percent of our overall collected data for training and thirty percent for testing, determined randomly. We will then apply machine learning to the training set; however, we have not decided on a specific algorithm as we will need to experiment to decide the most accurate and appropriate method to use once we have collected our data. We hypothesize that the SVM algorithm could be useful as it is a supervised clustering model. This appears appropriate as we don’t need to predict, for example, if a grasp is *almost* successful. We will adjust the specifics of this section as we progress throughout the semester.

4. Evaluation

Before we transition to each step outlined in section 3, we plan to evaluate our progress. Since we will make use of the pre-existing grasp and lift verification actions available, we will not test the accuracy of these actions. However, we will need to test the accuracy of the computed point cloud corresponding to the arm and end effectors. To do this, we intend to first publish this point cloud using rviz. Our rviz configuration will include a 3D model of the robot, available on the laboratory’s codebase, as well as the image returned from the camera sensor. This will allow us to visualize both the computed point cloud that we publish as well as the actual location of the entire arm. If there is a major discrepancy, we will adjust our code accordingly, otherwise small differences will be regarded as sensor error.

Next, we will evaluate the accuracy of our constructed model. Similar to the process for evaluating the computed arm point cloud, we will publish the object’s model to rviz as the procedure carries out. This will allow us to visualize if any addition to the model is incorrectly placed or if there is any major discrepancies.

Finally, we will test the result from our machine learning training. As previously mentioned, thirty percent of our original collected data will be reserved for this purpose. Using our trained algorithm, we will attempt to determine an appropriate grasp for each model in our test set. If the result returns an unfavorable grasp when another, more appropriate one exists, we will adjust our algorithm accordingly.

5. Expected Contribution

We hope that by the end of the semester, we will have implemented a system on the robot that allows it to create 3D models of objects. We believe that the ability to create and store 3D models of objects could be prove to be useful in improving a robot's ability to interact with its environment. In our own project, we hope to be able to use the models to train a robot to find the "best" grasp for an object. This could likely improve the robot's grasping accuracy. As the picking up and carrying of an object is a basic but useful skill for an autonomous robot, it is important that the grasping be as accurate as it can be.

6. Future Work

Due to time constraints in this project, we will be focusing only on how 3D models affect the grasping of objects in a limited scene. This could be extended in future work to include learning about other attributes of the object, such as the behavior it follows after certain actions are performed on it. Furthermore, the overall algorithm put forth in this paper can be implemented to work reasonably well when the object is present in clutter. This could be done by creating a planning algorithm where the robot explores objects in a scene one by one, beginning with the object it has the least information about. In combination with knowledge found from performing other actions, the robot could learn substantial information about objects in its every day exploration for later use in completing desired tasks.

Moreover, 3D models are useful in teaching a robot to recognize objects in its environment. Theoretically, the robot would store a library of basic 3D models. Whenever the robot encounters an object in its environment it wishes to identify, it could use a recognition algorithm where the unknown object is compared to ones stored in the robot's memory. If the new object reasonably matches a stored model, the robot could label and interact with the object in its representation of the current environment. However, if no match is found, the robot could create a new 3D model from scratch and store it for later identification.

References:

[1] Piaget, Jean. *The Origins of Intelligence in Children*. Vol. 8. No. 5. New York: International Universities Press, 1952.

[2] Krainin, Michael, Brian Curless, and Dieter Fox. "Autonomous generation of complete 3D object models using next best view manipulation planning."*Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011.