Box Stacking Bot - Proposal

Martin Iotchev, Panos Skoufalos & Tory Kolbjornsen

I. Intro

Most tasks in the real world revolve around identifying a problem, identifying what tools are needed to solve the problem, determining where the tools should be applied, and then using the tools properly and in the correct order. When focusing on automating intelligence and teaching robots how to apply their problem solving skills to larger systems it is essential for the robots to have a strong foundation in each of these areas. Specifically, robots need to be able to distinguish which objects in their environment are needed for a task. Identifying where those objects are located, distinguishing between those objects and other surrounding objects, executing their tasks so to move those objects to the location they need to be applied, and then interacting with them properly.

These steps readily manifest themselves into the box stacking problem. A robot will be tasked with finding certain boxes (targets) and stacking them in a desired location. Logically, this will require the robots to be able to identify boxes in an environment and gather information regarding these targets. Once these targets have been identified, the robot will move them to a new specified location while calculating a changing destination point relative to the changing environment and how many boxes it has already stacked.

First, the robot must identify what boxes it needs to move and locate them. If the robot cannot execute this protocol, it will have the ability to ask for help or clarification. Once the targets have been identified and validated, coordinates regarding the boxes locations will be calculated. Using image processing the robot will first identify, which objects in its view are boxes and then, which box is the biggest. For the robots we are using, adding arms to lift the targets is not practical nor essential for the logic or scope of the problem we are trying to solve. Instead of stacking vertically our robot will be lining boxes up horizontally in order of size (so that it could be stacked easily with proper lever arm extensions) starting with the largest box as its 'base'. The turtlebot will push the boxes to their desired destination and then reconfigure its position so to find the next box, notably it will have to alter its destination for this next box to line it up next to the previous box. It will repeat this find box, move box, "stack" box process until all of the desired boxes have been placed in the correct order. Below, you will find a detailed technical approach on how we plan to tackle this problem.

II. Technical Approach

In order to complete this project we will be tackling the problem by breaking it down into separate, testable modules. Each module is designed to incrementally add complexity to our systems starting with a very basic setup, testing it vigorously, and scaling up. The first module will be using image processing to identify boxes in a room. We will start with a simple set up of two boxes, with different sizes and using image processing to identify the boxes in a variety of locations in a room. This will test the validity of our image processing, the ability for the turtlebot to scan and move around a room and the ability for the turtlebot to ask for help if it is uncertain of the boxes locations. The second module will concentrate on identifying boxes and moving them to a set destination. We will be testing this module with different sized boxes and making sure that our turtlebots "arms" can handle a variety of sizes. We do plan on adding a motor or two in order to have variability in how big of a box the "arms" can move. Additionally once the boxes have been identified and the robot is positioned in front of the boxes we will be testing the turtlebots ability to move the box to a predefined space in our test room.

A third module will be completed that builds on the previous two. This module will optimize the robot's ability to decipher between different sized boxes. This module is similar to the travelling salesman problem that we completed earlier in the semester. The robot will be placed randomly in the room with a variety of different boxes and it will identify, travel to and quantify each box by size. For example, the turtlebot will spin around the room so to identify the boxes and their size and travel to the boxes from smallest to largest. The reason for this module will be to test and compare different algorithms for identifying different sized boxes and how to most optimally visit them. The fourth module will build directly off of the third module by adding the moving component to our system. This module will focus on actually getting a grip of the boxes and moving them to a specified place in the room. The robot will stack them side by side starting from largest (the base) to smallest (the pinnacle). We will be testing each module extensively so to emulate [3] breakdown of intelligence, building from the simplest actions and going up to a more and more complicated system.

In order to keep the programming relatively feasible we will be making use of Planning Domain Definition Language (PPDL) files and creating a variety of problem files for both testing and for executing specific actions. Additionally, extensive planning algorithms will have to be researched so to find the most fitting and optimal logic plan for moving our boxes. At the moment we are considering implementing an approach that extends the Kd tree-based ANN algorithm [1]and focuses specifically on efficient nearest neighbor searching for motion planning.

III. Evaluation Plan

Being able to determine whether or not our robot is successful is paramount to the development of the project. We will employ both analytical and empirical measurements and tests in order to validate how successful our robot is in achieving its goals.

As mentioned, the first module serves the purpose of identifying boxes. As such, our image processing algorithm needs to be able to reliably determine what is a box and what isn't. A good algorithm needs to have high accuracy so that the next modules are capable of carrying out their operation.

The second module is going to be responsible for moving the boxes to their actual places. A good planning algorithm will find a plan to solve a particular problem without getting stuck and producing a result that satisfies the ordering condition for the boxes.

The third module is mostly centered in optimizing the time it takes to solve a particular problem. Our robot might be able to solve the box stacking problem, but if the robot requires a substantial amount of time to complete the task, it is inefficient and does not correctly serve its purpose.

The metrics we employ to judge and evaluate the success of our robot should be quantitative in nature [6]. The first module deals with the robot's ability to identify boxes and the metrics related to that are the number of objects (boxes) detected and the amount of mislabeled objects (false positives or false negatives).

Our second module handles the actual planning and sorting of the boxes. For planning, we will measure whether the robot was able to create a successful plan and the number of steps required to achieve the goal [2]. As for the actual stacking, we will check whether the boxes are in the desired order and how many of them are out of order. Other data to gather at this step include the type of boxes that the robot struggles the most with. For example, whether the robot is more easily fooled by larger boxes or by irregularly shaped boxes.

Our third proposed module deals with optimizing the plan the robot uses for reaching its goal, so we will measure the time different algorithms take to get to a fully stacked box state.

For the fourth module, we will test the robot's ability to "pick-up" different boxes and see which boxes it has a harder time managing. For example, boxes that are wider than the robot might be picked up, but the robot might not be able to navigate properly given its new dimensions.

IV. Related Work

A large scale implementation of the basic box stacking problem is being implemented by amazon robotics. They credit their low prices and ability to provide two day shipping to the phenomenal automation that they have accomplished where robots move objects around the floor and robotic arms transport items from bins, to conveyor belts, and back again. Using these robots also allows amazon to stack their shelves far higher than a human can reach and closer together since the robots can maneuver in much tighter spaces. The key difference between our stacking robots and amazons current set up is that our theoretical robotic arms are mounted on the moving robots which will maneuver to a desired location/shelf to stack. Whereas the amazon facilities have the shelves themselves on movable bases and leverage robotic arms and their human workers to remove and replace items from this shelf. So in there systems, the shelves move to the arms instead of the other way around. [5]

Although the concept of grasping is outside the scope of our project and frankly outside of the scope of the turtlebot capabilities, the algorithm that is used in Saxena's supervised learning systems [4]. Here they are altering the complexities of 3D visual comprehension by presenting an algorithm, "*that neither requires nor tries to build a 3-d model of the object. Given two (or more) images of an object, our algorithm attempts to identify a few points in each image corresponding to good locations at which to grasp the object."* Again, although we will not be using the grasping component in our project this algorithm for understanding an environment and identifying targets could be very useful. Additionally there is an alternate algorithm introduced by Atramentov and LaValle circa 2002 which makes use of nearest neighbor nodes [1]. This could be very useful in identifying boxes and the most optimal way of visiting the boxes.

Sources:

[1] Atramentov, A., and S.m. Lavalle. "Efficient Nearest Neighbor Searching for Motion Planning." *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, 2 Aug. 2002, doi:10.1109/robot.2002.1013429.

[2] Chang, K.-H., and W.g. Wee. "A Planning Model with Problem Analysis and Operator Hierarchy." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 5, 1988, pp. 672–675., doi:10.1109/34.6775.

[3] R. Brooks (1991). "Intelligence Without Representation", Artificial Intelligence, Volume 47, Issue 1-3.

[4] Saxena, Ashutosh, et al. "Robotic Grasping of Novel Objects Using Vision." *The International Journal of Robotics Research*, vol. 27, no. 2, 2008, pp. 157–173., doi:10.1177/0278364907087172.

[5] Wingfield, Nick. "As Amazon Pushes Forward With Robots, Workers Find New Roles." *The New York Times*, The New York Times, 10 Sept. 2017, www.nytimes.com/2017/09/10/technology/amazon-robots-workers.html.

[6] Godil, Afzal, et al. "Performance Metrics for Evaluating Object and Human Detection and Tracking Systems." *National Institute of Standards and Technology*, 2014, doi:10.6028/NIST.IT.7972