Dropping Disks on Pegs

Adam Campbell

Cpr E 585X Project Proposal

Dr. Alexander Stoytchev

10 March 2011

Introduction

My project involves getting the robot to learn to place disks on pegs. Some household activities involve placing things on pegs, such as … But the most practical application of the ability to place disks on pegs is the Towers of Hanoi. The Towers of Hanoi is a game invented by Edouard Lucas in 1883. The goal of the game is to move a stack of disks from one peg to another, using an auxillary peg as needed. Only one peg may be moved at a time, and smaller pegs can only go on top of larger pegs (see Figure 2).

Figure 1: This toy is designed to help children learn how to put disks on pegs. (image source: http://vtonlineshop.com)

Human children learn to place disks on pegs through toys and activities like the one in Figure 1. The children will try to place the disks on the peg through random motor babbling, and every time they are successful, they improve at their success rate.

I plan to teach a robot how to place disks on pegs by having it drop the disks from above the peg. The robot will see if the disk landed on the peg, and if it did, it will register that location as a success point. In the end, there will be a map of locations that have different probabilities of success, and the robot will be able to choose the location with the highest chance of success.

Figure 2: The Towers of Hanoi. (image source: http://www.mrpearse.com)

Related Work and Other Attempts

At Carnagie Mellon University in 2007, Chang et al. created a robot that would solve the Towers of Hanoi, perhaps the most practical application of the ability to drop disks on pegs. The robot would rotate a platform with disks and pegs, and lift the disks using precise and carefully measured controls. All the robots movements were preprogrammed, and it was told explicitly what to do, and when. (Chang et al., 2007).

In a similar project at Berkley in 2005, Tesch and Hsu created a robot capable of solving the Towers of Hanoi. Like Chang et al., this robot relied heavily on preprogramming to know where to move and how to handle the disks.

The problem with both of these approaches is that it involves no learning. Because of this, there is no room for adaptation. The robots have no way of verifying the success of the operations, and so would be clueless without humans there to make sure they put the disk on the peg correctly. If one of the disks were to miss the peg, the robots would just go on their merry way without a care in the world, blissfully unaware of their mistakes.

By having a robot *learn* how to put the disk on the peg, instead of being *told* where to put it, the process can be much more flexible. If the robot has learned to pick the most probable spot for dropping the disk, then it can account for changes such as moving the peg during runtime.

Me

My name is Adam Campbell, and I am a freshman in Software Engineering at Iowa State University. I have about 1 year of experience in programming in C and C++, and I am eager to learn new things about computers. I decided to take this class because I am interested in robotics and I want to see the kind of amazing things that they can do. Even though robots right now are in the stage of toddlers, I believe that the future holds great advances in intelligence for robots.

Setup and Equipment

For this project, I will be using the upper torso humanoid robot in Figure 3. The robot uses two 7-degree-of-freedom Barrett Whole Arm Manipulators (WAMs) as arms, and each arm has a 3-finger Barrett Hand attached to it.

I have already made some pegs out of a long wooden rod that I cut into sections. For disks, I have obtained several hard Styrofoam disks and cut holes in the center so that they can be placed over the pegs. The Styrofoam disks, however, are somewhat brittle, so I will probably end up replacing them with more durable wooden disks.

For the camera, I will use a simple webcam to capture color images. I will keep the background stationary so the only things moving will be the disk and the peg.

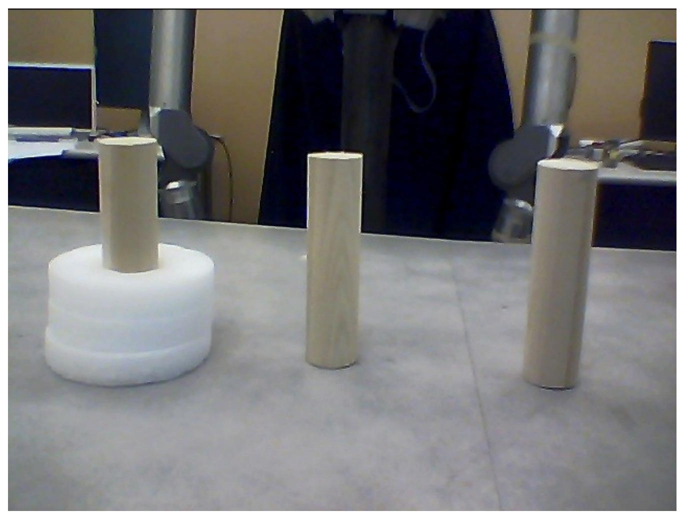


Figure 4: The setup. The robot will drop the disk on top of the peg.

Figure 3: The robot

Figure 4 shows the setup for the experiment. I will only end up using one of the three pegs, but I have the other ones in case something happens to the main one. I will attach a wooden base to the peg, and that will allow the robot to push it more easily, as well as prevent the peg from tipping over.

I will paint the peg and disk distinct colors, so the object detection will be much easier. I am currently planning on painting the peg solid black and the disk solid red.

Method

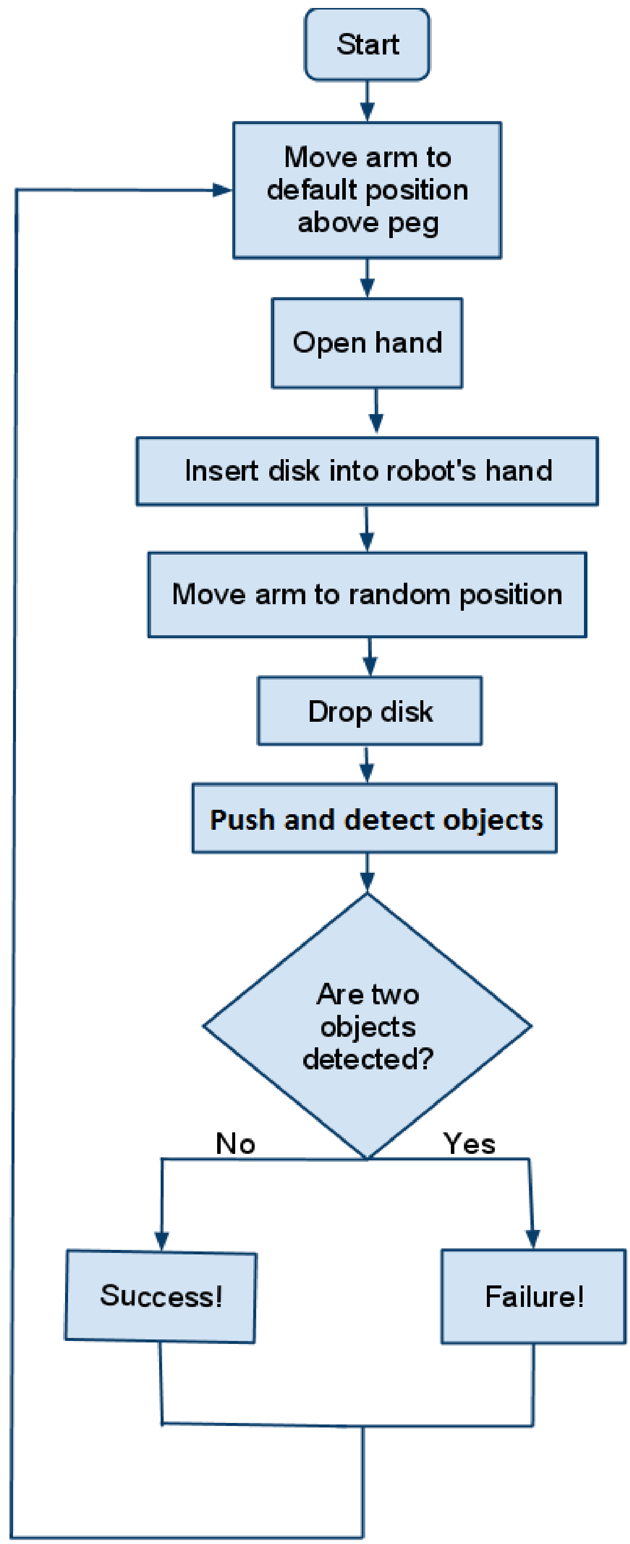
Overview

The general approach to the problem is to drop the disk on a random position near the peg, push the peg from the base, and see how many objects are detected.

If there are two objects detected, then the disk is not on the peg and the robot will register that location as a failure.

If only one object is detected, then the disk is on the peg and the robot will register than location as a success.

The robot will learn the best spot from which to drop the disk, based on the probability of the disk landing on the peg. The probability of a disk landing on the peg can be determined by the number of successful dropping attempts at that particular spot in the past. So a location far from the peg would never have any successes, and would therefore have a very low probability of success. On the other hand, a location right above the peg would produce many successful attempts, and have a very high probability of success.



Specifics

I am planning to use vision to detect the number of objects after each drop. I will try to implement the algorithm used by Shane Griffith and Alexander Stoytchev in their container experiment, which is as follows:
When the robot pushes the peg, it can use the visual co-movement patterns of the disk and the peg to count the number of objects. That is, the robot will register a movement when either the disk or the peg moves by a certain threshold. This produces four possible events: 1) Neither object moved; 2) The disk moved; 3) The peg moved; or 4) Both objects moved. (Griffith and Stoytchev 2009).

I plan to first do 200 trials, dropping the disks randomly. After the data from the random trials has been collected, I will do 200 more trials, but this time trying to aim for the most probable region of success. Using the data from all these trials, I will do 200 final trials, this time aiming for the regions where there is the least certainty (on the edge of the certain region). This way, the robot can explore the boundaries of the unknown and increase its certainty.

When all the trials are completed, I will measure the robot's learning by having it drop the disk on the location that it thinks has the highest probability of success. The Project's success will be defined as the robot having a 90% success rate in dropping a disk onto the peg. If the robot's success rate is lower than 90%, then I will either: 1) have it do more trials until the rate improves; or 2) improve my method if the rate seems to be stagnant. If the poor success rate is a result of physical limitations (e.g., the disk doesn't fit correctly on the peg), I can just resize the holes.

For software, I plan to use OpenCV for image processing, and I will implement the object detection algorithm as previously mentioned. I haven't used image processing software before, so it will probably be a challenge to learn how to use it. I think that what I'm doing, however, can be done in the time allotted for the project.

Timeline

March 21th: Have all the supplies for the experiment ready, including painted disks and peg, and the base attached to the peg.

March 22th: Begin writing the code for the robot's behaviors.

April 7th: Begin doing trials with the objects.

April 14th: Finish testing, begin final report.

April 21st: Finish final report, complete project

Works Cited

1. Griffith, S., and Stoytchev, A., "Interactive Categorization of Containers and Non-Containers by Unifying Categorizations Derived From Multiple Exploratory Behaviors," In Proceedings of the 24-th National Conference on Artificial Intelligence (AAAI), Atlanta, GA, July 11-15, pp. 1931-1932, 2010.

2. Chang, J., Rubi, N., and Hassavayukul, P. "Towers of Hanoi Final Report". 2007. <http://www.ece.cmu.edu/~ece578/teams/H/>.

3. Tesch, J., and Hsu, J. "Towers of Hanoi Robotic Solver". 2005. <http://www.me.berkeley.edu/ME135_S05/projects/gracejonathon.pdf>.