Requirements
3D object pose tracking for Robotics Grasping
Lux Vision
CS461 Fall 2017

Connor Campbell, Chase McWhirt, Jiawei Mo

Abstract

As robots and artificial intelligence become increasingly important in the modern age, they will need to be able to interact with their environment on their own, and computer vision is an integral part of a robot's understanding of its surroundings. Oregon State University's robotics department has an ongoing project attempting to improve computer vision, and this Capstone project is aimed at furthering that goal. Specifically, this Capstone project will be the culmination of implementing various dynamic masking techniques that can be executed in real time.

Contents

1 PURPOSE

This document will define the goals given by the client. The goals will include specific tasks that can be accomplished by a single person. These tasks will be defined in such a way that they are easily distributed among the group members. Tasks will also effectively be verifiable.

2 OVERVIEW

2.1 Product Perspective

The project is only a piece of a much larger project. The larger project is a robotic grasping problem. Oregon State University has a robotic arm that should be able to grasp and manipulate an object within a specific environment. The environment will be a box containing the desired object for manipulation, the robotic arm, a single light source, and two cameras placed on adjacent corners of the ceiling. This particular robotic arm has no tactile sense. Thus, it must be able to understand the entire environment with only optical input. The project discussed in this document is geared to solving the problem of handling optical input. The current code behind this functionality is able to understand its environment by finding cross sections within its two camera inputs. This effectively gives the robot stereo vision. However, its sense of objects is extremely limited. The code behind the robotic arm attempts to create a mask over important objects in the scene. This functionality expects uniform color though. Since the environment only contains a single light source, it is essentially impossible for an important object to be uniformly colored. Light diffusion will make an object appear covered in a wide variety of shades. If there were multiple light sources, this effect could be decreased. However, since an object is going to be manipulated, creating perfect lighting conditions is highly unrealistic.

It should be noted that these functions will primarily be used by the University's robotics research team. They won't be robust enough for complex, real world applications, but are rather meant to be a step forward for future research.

2.2 Product Functions

In order to solve the problem of inconsistent light sources, this project will have to create and execute a more dynamic method of image masking to accurately identify the object. The client has discussed multiple possible approaches to do this, but is not sure which method will work best. There will be three different implementations attempted and these will be tested against a straw man implementation. The best implementation measured against the straw man will become the final product. This project will only attempt to find the best masking function for identifying the robotic arm itself. As stretch goals, the best method found in the main project will be implemented to mask the object the robotic arm will grasp. As a final stretch goal, both masking methods will be verified as compatible with each other.

2.3 Constraints

This phase is a part of a big project. Our group is to undertake only the lighting changes during object grasping. As such, it will need to interact appropriately with the rest of the project. Since this program is meant to replace an older identification system, it will be ideal for the inputs and outputs to be similar, in order to minimize the need for adjustments to the rest of the project.

3 System Requirements

3.1 Functional requirements

To begin, this project will be based on images and pose transforms that our client has gathered. These images will be of the robotic arm in various poses, and will be used both for training and testing the algorithms. The initial straw man will be determined by simply running a k-means clustering over all of the data to determine a color range to look for. Next, the images will be grouped based on the arms pose. Each group will have its own version of each of the three methods. The three methods include another k-means clustering (though over a single group as opposed to all of the data), neural networks, and support vector machines. The method that

3.2 Performance requirements

The goal is for the best of the three methods to replace part of the larger projects current system. As such, it should label at least 80% of the pixels correctly as either part of the arm or not. Ideally, it should have at least 90% accuracy, but 80% will at least be an improvement.

4 Verification

Implementation effectiveness can be measured in two distinct ways. First, it can be measured against a designated "truth" defined when images were manually masked. Each pixel will fall into four groups. True positives will be a particular implementation masking a pixel as part of the robot arm being correct. True negatives will be masking a pixel as not part of the arm and being correct. False positive will be masking a pixel as part of the arm and being incorrect. Finally, false negative will be masking a pixel as part of the arm and being incorrect. The client has established that false negatives are worse than false positives. This is because false positives could exist due to environmental conditions such as background color. False negatives mean that the algorithm has failed to correctly identify the arm through flaws in implementation or even the algorithm itself. The client has also stated a desired accuracy of 80% or more, though, she has also said this is not a hard requirement. Rather, 80% is an arbitrary amount she expects will be practically effective, though the percentage could be lower or higher.

The other way to verify each implementation is whether or not it is more effective than the straw man. If it is less effective, the implementation fails. If it is more effective, it passes. It is possible to measure how much more effective the implementation is over the straw man. For example, if the straw man has 40% of all true positives and implementation x has 80% of all true positives, implementation x is twice as effective at defining true positives than the straw man. This concept will be important for defining how successful any implementation is.

5 Definitions

- K-means Clustering - An algorithm that involves dividing a group of data points into smaller groups such that each point is in the group with the group average most similar to it.
- Mask - An overlay defining a set of pixels as a unique object. Applying a mask over an object is the main goal of the project.
- Neural Network - A form of machine learning designed to mimic the activity of organic brains by simulating neurons firing in order. This can be simplified as a multiplying a series of matrices to the input.
- Stereo Vision - Also known as binocular vision, this is the mechanic of taking two images taken from different angles and defining a perceived depth.
- Straw Man - In this instance, a straw man is an implementation that is expected to under perform. An extreme general approach is tested and measured. It then acts as a baseline metric to compare to other implementations.

6 Gantt Chart