Lab 6

Robot Competition

Objective

In this experiment, students will continue using the PUMA 260 robot manipulator to achieve the following goals:

1. Develop a C/C++ or Matlab program to successfully throw an object.
2. Design a robust image processing pipeline to integrate with the program designed in 1.
3. Develop a strategy to pick up an object and throw it to the scoring zone of the competitor robot.

Reference Materials

1. Lab 2 handout, reference manual of image processing toolbox and the template C++ functions for controlling the Puma robot.
2. Matlab code from Lab 2.

Lab Procedure

This lab is made of three parts:

A. Throwing Object

1. The PUMA 260 is able to interface with the C++ program and the gripper through the serial port. You are to design a robot motion for throwing an object while integrating all necessary functions in one C++ program.

2. Your goal in this part is to design your own binary that is able to control the robot and the gripper during the path from one starting point to one ending point. You can later use your binary with a system call from within your Matlab program to perform the throw!
B. Image Processing Pipeline

1. Before we proceed to part C, you have to make the Matlab program from lab 2 robust enough to achieve the object detection and object reconstruction using a sequence of image processing algorithms. For example, your program should be able to deal with shade and the edges of the boxes.

   **NOTE:** You may also need to calibrate the cameras again.

2. Integrate your robust image processing pipeline with the C++ binary designed in Part A. You should control the robot under Matlab environment and all the C++ binaries should be called by the Matlab “system” command.

C. Scoring Strategy

1. The scoring contains two steps: pick up the object and throw the object through a long passage across the two robot workspace.

2. Try to make your robot throw the object at its highest acceleration so that the object will be thrown far enough to reach the scoring zone of the competitor robot.

3. Think about implementing a defense strategy as well!

4. You may want to use Leap Motion to improve your game strategy!