Robot Control through Gesture Recognition Proposal

Ben Yett

Charlie Hartsell

Overview

We intend to implement a robot control system using gesture recognition as the primary user input method. Gesture recognition will be performed using the Microsoft Kinect sensor, which provides both an RGB video stream and a depth map of the environment. Our project will utilize a robotic ground vehicle such as the iRobot Create, which will initially be simulated in the open source robot simulator Gazebo. While our project will utilize a ground vehicle, the gesture recognition and control software could be configured as a high-level controller for other types of robots as well. Additionally, we plan to model our system as a Finite State Machine (FSM) and perform formal analysis to verify correct operation. Finally, we plan to transition from using a simulated robot to a physical robot, if time permits.

Approach

We plan on using a Kinect motion sensor to accomplish the gesture recognition aspect of the project. Similar to a webcam, the Kinect was originally designed by Microsoft for their XBOX 360 gaming system as a an alternative controller for specific games. The device was later revised for use on Microsoft Windows and the XBOX ONE. Following the Windows release, Microsoft also released a Kinect SDK to allow developers to create apps or otherwise apply the sensor for their projects. There is already a sizeable amount of work that has been accomplished with Kinect that we will be able to draw from. We will have to perform certain non-trivial modifications as well.

Our original plan also involves using a robot to respond to these gestures in some way. We plan to begin with a simulation approach using the open source Gazebo simulator, and as a stretch goal we would move on to a real robot depending on time and availability. The Gazebo simulator has been chosen for the following reasons:

* Team’s prior experience
* Ease of interfacing with external programs
* Library of pre-existing robot models including the iRobot Create
* No cost to acquire and open source

Objectives

One main objective for our project is to enable the Kinect to properly recognize various gestures. These gestures will symbolize certain things for the robot - real or simulated - that we will use in tandem with the sensor. The Kinect SDK or another similar open source tool will be used to program the gestures that should be identified. For example, waving could be used to indicate turning in the appropriate direction to the robot, and can be implemented by examining the position of the hand relative to the elbow at a few moments in time. If the hand is consistently above the elbow and is also alternating between being to the left of and the right of the elbow, the Kinect will recognize that the person is waving. Gesture recognition poses a soft real-time problem since the Kinect sensor must be sampled and processed at a high enough rate to provide acceptable accuracy and responsiveness.

The next step is to decide on the logic for the robot controller. We plan on modeling this using an FSM, with transitions between states occurring as the result of the gestures recognized by Kinect. The eventual idea is to have incoming gestures causing these transitions in one state machine, which is composed with another state machine that receives outputs from the first FSM. These outputs in turn determine the behavior of the robot. This is where the formal analysis will come in - while our system is not safety critical for these examples, it is still important to ensure that no deadlocks will occur. We need to have a system that is complex enough to handle a variety of gestures and combinations of results. It should also run quickly enough to clearly show that the robot is performing the correct action before new inputs are added.

As previously discussed, we will be using Gazebo for our initial simulation setup. Ideally, we would like to expand on this by applying our controller to a real robot. While we do not currently have such a robot available to us, we will attempt to obtain one and implement the final model with it in the final weeks of the project. The simulation is a good tool but we would still like to have a physical example assuming there is some time left to work on it. Finally, we have the general objectives of planning, practicing, and executing our presentation and demonstration, as well as completing the final paper for the project.

Major Deliverables

1. Software Package including:
   1. Kinect gesture recognition software
   2. Robot simulation environment
   3. Robot controller
2. Project Report
3. Presentation and Demonstration

Constraints

We have concerns about the amount of time available to complete this project, both in terms of the time remaining in the semester and in terms of our ability to dedicate large amounts of time to this while still keeping up with our other responsibilities. In addition, our budget and available resources have led us to implement the following constraints:

1. Use of Kinect sensor module
2. Free and/or open-source Software
3. Availability of physical robot to meet stretch goal
4. Gazebo simulator due to team experience

Risk and Feasibility

Primary Risks:

1. Inexperience with Kinect sensor and SDK
2. Difficulty of accurate gesture recognition
3. Difficulty of creating a suitable simulation environment and robot model

Mitigation Strategies:

1. Utilize pre-existing research on gesture recognition with kinect sensors
2. Utilize pre-existing Gazebo model libraries and environment creator

Project Milestones and Schedule

|  |  |
| --- | --- |
| DATE (Week of) | MILESTONE |
| March 10 | * Research on Gesture Recognition, Kinect SDK, and Gazebo Simulator |
| March 17 | * Successfully interface with Kinect * Create Gazebo environment and robot models * Create initial FSM model of system |
| March 24 | * Basic Gesture recognition * Interface between controller software and Gazebo |
| March 31 | * Integrate Gesture recognition with controller software * Additional Gesture recognition functionality * Refine FSM model |
| April 7 | * Formal analysis of FSM model * Testing and Simulation * Work towards stretch goals (Time Permitting) |
| April 14 | * Troubleshooting * Preparing Presentation * Begin Project Report * Work towards stretch goals (Time Permitting) |
| April 20 | * Presentation and Demonstration * Continue work on project report |
| April 27 | * Project Report Due |