Written by Lior
Wednesday, 21 April 2004 00:00 -

Abstract

This project will implement a system for controlling a robot arm based on biologically inspired techniques. The goal of the project will be to have a robot arm reach for an object. This will be accomplished using a camera that will fovate on an object using its 2 degree-of-freedom (pan and tilt) mechanism which then will use a 6 degree-of-freedom arm to maneuver to the object, either touching the object or pointing to it. The solution for this problem resets in the mapping between vision coordinates to angle coordinates on the arm (assuming that the coordinates are already known either using motion detection or markers). Previous attempts to solve the mapping problem were to determine mathematically the formulas required for this mapping. Some of the problems with this method are that it is often very difficult to find the formulas to perform the mapping. Moreover, if one of the joins of the arm fails (due to a motor failure or arm link replacement) the formulas have to be manually recomputed. The solution propose in this project will be to use Self-Organizing-Feature-Maps (SOFM) to learn the mapping. Such network will enable the robot to learn the mapping required to map vision coordinates to arm coordinates. Since the learning will be completely self-supervised, the arm will be able to adapt to changes as well as generalize the mapping (which means that not all coordinates mapping will have to be learned, saving on learning time). This project draws its inspiration from COG, a robotic humanoid developed at MIT. **The system proposed here would be similar to the one proposed for COG in visually guided pointing. However, instead of using the cascade and ballistic maps, a SOFM will be used.**

Implementation

To achieve the task, the system will be broken down into three components. The first will be to locate the object that is of interest and determine the end of the arm (the fingers) in the vision field. The second component will be to map between pixel coordinates of the object to 2 dimension values, which will give the pan and tilt position for camera to fovate on the object. The third component will map between the pan and tilt position of the camera to an arm position. Each component will then be broken down into the following sub systems.

1. First Component: Vision recognition

This will be accomplished using one of two techniques. The first will be to use markers on the object and the end on the arm. The markers will be colors chosen arbitrarily, but will be unique to the background. For instance a red color will be chosen for the object of interest and a green color for the end of the arm. As long as red and green do not appear in the background a color recognizing software will be able to determine the x and y position on the markers within the visual field. The second method of determining the coordinates of the arm and object will be to use motion detection. This method will use the absolute values between the differences of two successive frames acquired from the camera. The values will then be threshold to find the coordinates of the object that moved. However, using this method will require the object to move first, followed by the arm so that the system will not be confused as to what to track. Both methods will use the center of the object as the coordinates.

2. Second Component: Fovate on the object

A SOFM will be used to map between the pixel coordinates and the pan a tilt coordinates in order for the camera to place the object in its center. The SOFM will be used as a self-learning lookup table in order to achieve the mapping. After getting the pixel coordinates from the first component an input vector of 2 dimensions consisting of where the image is in the vision field will be inputted to the neural network. The winner node of the map will be selected which will contain the pan and tilt coordinates. Research will still need to be done for the best method in training the network.

3. Third Component: Mapping between pan tilt coordinates to arm coordinates

In order to limit the dimension of the arm position vector, a system similar to the one obtained from the associations of the movement in frogs will be implemented. It was found that the legs of frogs moved to a given fixed position under a stimulus, and that there were only 4 of these positions. The conclusion was made that these postures were primitives and that the combination of these primitives result in the desired movement. Using this technique in the robot arm will greatly reduce the input vector to the network. The SOFM network will then be processed similar to the second component, mapping the coordinates between the pan tilt and the percentage of the primitive positions.

Systems and Software

1. A 6 degree-of-freedom arm (L6) design by Lynxmotion.
2. Pan and tilt mechanism, which will be custom made to house a Quickcam 4000 camera.
3. An IsoPod DSP microcontroller that is running a version of Forth from NewMicros will be used to communicate with the servos.
4. An 800mhz computer running a version of Linux, which will perform the vision and mapping computations.

The software will be written mainly in C++ using Gtk+ as the graphical interface. Forth will be used in the microcontroller to perform the servo calculations and outputting the PWM waves to the servos.

Potential Difficulties

Some of the potential difficulties, which will be encountered in this project, will be with the SOFM. The map may not learn the correct mapping or will not be able to generalize correctly, which in turn will cause a wrong movement of the arm. The methods in training the SOFM will probably be a key in determining if the map will be correct or not. Other difficulties will be in getting a consistent position from the center of the object from the vision field across multiple trials, as well as other inconsistencies in the servos used in this project. These inconsistencies might cause the map to misbehave; however the ability of the map to generalize will, hopefully, overcome some of the small inconsistencies.