

Computer Science Tripos Part II Project Proposal

Gesture-controlled Robotics using the Vicon Motion Capture System

C. J. Hung, King's College

Originator: C. J. Hung

24 October 2008

Special Resources Required:

Vicon Motion Capture System

Two iRobot Creates and OLPC XO laptops

Project Supervisor: Cecily Morrison, Laurel Riek

Signatures:

Director of Studies: Simone Teufel

Signature:

Project Overseers: Graham Titmuss & Markus Kuhn

Signatures:

Introduction and Description of the Work

The objective of this project is to provide real-time gesture-based control of robotics. Gestures can be characterized as temporally evolving data, so by exploiting the same techniques used in speech recognition, a stream of coordinates from the Vicon motion capture system can be used to drive a small robot.¹

Since the motion capture system has the benefit of capturing full body data, I intend to explore recognizing gestures whilst moving around in 3 dimensional space. This could allow the user to control two more parameters by their location.

The project will implement two different machine learning techniques for gesture recognition. The two approaches can be summed up as the 'Connectionist Approach', ie. Artificial Neural Networks, and the 'Stochastic Approach', ie. Hidden Markov Models. Both are well researched and have many applications in pattern recognition.

Direct evaluation will be based upon the success rate, using a common test set of recorded motion capture samples, and speed/efficiency, since the robot must be driven in real time. The evaluation will also take the cost of training into account, using the same training data for ANN of different number of hidden nodes and HMM with different number of states.

In addition, the system should support multiple users, each controlling a single robot (limited to two for the physical robots). Interaction between robots is supported by the touch sensors for a simple collision based game.

Resources Required

The Vicon motion capture system will be used to collect data from gestures. Since most of the work can be done offline, initial work will include collecting and archiving multiple sets of data for training and testing purposes.

My current intention is to use a OLPC XO mounted upon an iRobot Create. This choice was based on several contributing factors, including ease of programming, ease of assembly and cost. In particular, the XO laptop handles wireless connectivity and includes a webcam.² I have secured all

¹However, this is not a hardware project; existing interfaces to the Vicon system and the robot will be used.

²This setup is based on <http://www.instructables.com/id/OLPC-Telepresence/>, which

necessary funding from the Women@CL Outreach programme and college for buying two robots. The itinerary for each robot:

- XO laptop: approx £120 second hand, second hand from Ebay
- UK - US adapter: £5
- iRobot Create: \$130 = £75
- USB to serial adapter: £10

Total: £210 per robot³

I intend to write a virtual robot which uses the same interface for initial testing. In case that the hardware fails or is unavailable, the project will be continued using the virtual robots only.

Starting Point

I have previously used Python for a small summer project. The PyRobot library interfaces with the Create's motors and sensors as well as the OLPC's webcam, which means that integration with the robots should be fairly hassle free. The Vicon motion capture system has a Java interface.

Substance and Structure of the Project

The initial setting up will include writing a virtual robot and recording training and test data from the Vicon system.

The bulk of the project will be implementing the two gesture recognition schemes, Hidden Markov Models and Artificial Neural Networks. My intention is to write my own implementation of these using Python, a high level scripting language, rather than using a pre-existing package such as the MATLAB Neural Network Toolbox or Hidden Markov Model Toolkit (HTK). This will ensure that external influences will be minimized when it comes to comparing the two approaches.

The recognized gesture will be mapped to the robot controls, which will be sent wirelessly to the XO laptop. The gestures will most likely include:

includes a Python library for driving the robot and laptop.

³One Wi-Fi adapter for the main computer (if needed): £10

- Start/stop (for the Create and webcam)
- Accelerate
- Decelerate/Reverse
- Turn left
- Turn right

Hidden Markov Models

The gestures can be modelled as a Markov Chain, a discrete-time process where future states are only dependant on the present state. However, since the underlying state is not visible, a learning algorithm together with a training set must be used to find the most probable state and parameter probability distributions. The most commonly used technique is the Baum-Welch Procedure, a generalized expectation-maximization algorithm which uses the forward-backward algorithm.

Artificial Neural Networks

An alternative machine learning technique is the neural network, based on a set of interconnected nodes or neurons. In this paradigm the feed-forward network, specifically the multilayer perceptron, typically consists of three layers (input, hidden and output) where each neuron employs a non-linear activation function. The well known backpropagation algorithm will be used for supervised learning.

By using the same training set and test data as the Hidden Markov Model, I will be able to compare and contrast the different approaches in terms of success rates and efficiency.

In addition, I also intend to investigate gestures moving through 3 dimensional space, to allow users to walk around while performing a gesture. An initial approach would be to calculate arm gestures relative to something that represents the user's location eg. a belt, then to subtract the effect of the user moving around.

Once the models are working with the virtual robot, I will implement an extension for multiple users, each controlling a robot in a simple collision-based game. Finally the switch to using real robots will move to using the touch sensors and streaming video from the XO laptop's webcam.

Criterion for Success

By the end of the project, the following goals should be completed:

1. Implement the Baum-Welch algorithm for training the Hidden Markov Model.
2. Implement a feedforward neural network and backpropagation algorithm.
3. Train with a set of approximately five gestures.
4. Demonstrate that a virtual robot can be driven using these gestures in real time.

Timetable and Milestones

7 November, two weeks

One week each of studying Hidden Markov Models and Neural Networks. Decide on gestures. Investigate Jython for Python-Java interoperability.

Milestone: Write Introduction.

21 November, two weeks

Record training and test data from the Vicon motion capture system. Write a virtual robot and test.

Milestone: Write Preparation.

19 December, four weeks

Implement feedforward neural networks and backpropagation algorithm. Test using archived data and virtual robot.

Milestone: Write first half of Implementation.

23 January, five weeks

Implement the Baum-Welch algorithm for HMM. Test using archived data and virtual robot.

Milestone: Write second half of Implementation.

30 January, one week

Milestone: Write progress report.

13 February, two weeks

Integration with real time data. Switch to physical robots.

27 February, two weeks

3D gestures using filtering. Add multiuser facilities.

13 March, two weeks

Write a simple collision game using the Create touch sensors. Stream OLPC webcam to projector.

20 March, one week

Final online integration and testing.

10 April, three weeks

Milestone: Complete Implementation. Write Evaluation.

24 April, two weeks

Milestone: Completed Dissertation.

8 May, two weeks

Proof read, Latex, bind.

Milestone: Submit Dissertation by 15 May.