

SimPooch

Canine Medical Acupuncture Training & Simulation System

Project Continuation Report

Ben Cordova
Brendan Dalh
Eric Hall
Jeff Barlett

7 December, 2007

SimPooch's ever persistent goal is to eventually fully integrate the Physical and Virtual Systems. As we have come to the close of another semester, we have seen both systems realize a huge amount of growth and development. The base structure for the Virtual System has been finalized. There is now an architecture in place, upon which the team can add new features and abilities. Finishing the Base Unit allows the smallest of connection and coordination between the two systems. Future SimPooch teams will only expand upon this growing connection to create fully fledged integration.

The best thing about this project is that it is mature enough to keep its team members from working without direction. Currently, there are five issues that compose next semester's team's primary foci. The first step will be to create stronger cohesion between the dog haptic model's angle value and the base unit's rotational output. At the moment, the two systems communicate through a single entry in a MySQL Database. The Base Unit constantly writes this variable and the dog model constantly reads it. This solution is not ideal as it involves a lot of overhead. MySQL is more processor heavy than anticipated. Also, there is no governance of the Base Unit's write rate and the Dog.py script's read rate. Based on observation, the Base Unit writes a new variable to the database at a much higher rate than the Dog.py script reads it; this difference in rate is inefficient. A better solution would be to integrate the Base Unit Device commands into the Dog.py script. Communication can be done through the use of the python "pySerial" module. This module allows python scripts to directly interface with a computer's COM ports. Our A/D converter interfaces to the computer via USB. The computer sees the USB device as port COM5. An ideal solution would have the Dog.py script query the Base Unit for a rotational value once for each scene refresh.

The next item to be developed by next semester's team also deals with the interface between the Virtual and Physical Systems. Currently, there is no way to tell the computer where the physical dog mannequin is in real space. Thus, the only way to line up the unit is by manually adjusting the initial

conditions of the haptic program scripts. This is a tedious process and its results do not persist as the entire system (haptics device plus Base Unit) is not yet stationary. To resolve this issue, the first step would be to mount the haptics device onto the base unit in a semi-permanent manner. It should be removable but unmoving. It should also, with the mannequin's rotation, be able to touch every part of the dog head. The second step to resolve the calibration issue would be to write a calibration script that prompts the user to touch certain known features on the dog head. The program would then adjust the variables in the three-dimensional environment to fully encase the physical mannequin in the virtual 3D model.

The following two objectives involve improvement in the Virtual System only. The team should write python code to fully utilize the Voronoi Algorithm when grading student points. This involves creating a three-dimensional map of distances to the closest teacher point. This will create a set of polyhedrons (one for each teacher point) that will define that teacher point's region. Then, any user touch in a given region will be assumed to be aiming at the teacher point that owns that regions. Multiple touches in a particular region define multiple attempts at a single teacher point. The absence of a user touch in a given region correlates to a miss. This functionality not only requires a Voronoi Algorithm to be applied to the teacher point set, but also a steepest descent algorithm which would direct a user point to its intended teacher point. This semester's team has already started coding this functionality. The best place to start would be to get the code working for two dimensions and then expanding it to three. The script that generates the Voronoi Map should be coded with an adjustable step size. This adjustment can then be used to adjust how long the script takes to generate the map; in general, a smaller step size results in longer processing time and more memory space needed to store the map. A smaller step size will also cause the steepest descent algorithm to take longer.

The other improvement to the Virtual System is general improvement to the User Interface. The best way to do this would be to sit down with Dr. Robinson and have her help to define a particular program flow. Program the functionality she is looking for as she is the primary customer. Ask her what she wants to see when she sits down at the system. There is nothing more wasteful than writing a

program that appears to be functional to the developer, showing it to them, and having them be completely unsatisfied and unimpressed.

The final step for next semester's team, after they get the above things working, should be to begin research into force feedback and force grading. One of the team and Dr. Robinson's main reasons for selecting a haptics device for data acquisition is the haptic device's ability to provide its user with force feedback. This idea has two facets. The first is providing the user a realistic sense of feedback when touching the mannequin with the stylus. The second use of force feedback is capturing the user's pushing force as a graded variable. Then, the user would be graded on whether or not they applied the correct amount of pressure to the patient in addition to hitting the correct mark.