Notes about Leap Studies

The Leap Motion Controller is a hardware sensor capable of contactless tracking of hand and finger positions. By doing some light editing to a script provided by its SDK, I managed to transmit data coming from the sensor via OSC, to be subsequently used by audio synthesis software such as SuperCollider. The issue then became how to classify a gesture and ascribe different meanings to its variations.

My experience as an orchestral musician made me interested in how hand gestures can contribute to give shape to musical material. Although I’ve come to question the overly hegemonic role played by conductors in symphonic orchestras (and all the political assumptions this entails), I’m still amazed by how variations on gestures can have some impact in the resultant sound. This can be easily accomplished in systems involving human beings as mediators by harnessing our finely tuned communicational skills, as we attach meaning to subtle variations in verbal and non-verbal communication. Even when there’s no clear one to one relation between intention and movement, we tend to ascribe one by relating to it vicariously.

In my past work with motion and gesture tracking I’ve focused on a very straightforward parameter mapping paradigm: one parameter of movement (be it location, speed, rate of change, etc) gets translated into one or multiple parameters for a sound synthesis engine. However, in order to consider gestures as they evolve in time as meaningful elements in our being-in-the-world and not only as positions or lengths in a three-dimensional space, I had to provide the mapping layer in the system with a certain kind of agency. In a way this is analogous to an orchestral performer being constantly on the lookout for the conductors’ gestures. Machine learning techniques such as classification helped me approach this goal, as well some basic gesture recognition natively provided by the Leap Motion.

The first two studies I worked on utilize only a single gesture: circles drawn by the index finger. In the first study involves rhythmic values that can be added or subtracted to a sequencer, their length determined by the diameter of the gesture. The patch reacts to four types of gestures: Clockwise circles adds and counterclockwise subtracts, while the hand used determines whether the rhythm will be added or subtracted from the beginning (left hand) or the end (right hand) of the list of values. An intuitive interactive system is thus created, with very clear rules and a transparent mechanism.

The second study involves the same set of gestures. However, instead of adding and subtracting single rhythmic values, each gesture creates and loops more complex musical phrases using the HenonC Ugen, a chaotic generator. Some parameters are determined randomly, but the range used by the pitched material and average amplitude are determined respectively by the diameter and the number of times the circumference is drawn. The resultant sounds are unpredictable but stay within some constraints determined by the gesture.

The third study consists of a 3-track live looping system. Each track can be selected by raising the corresponding number of fingers in the left hand using the index, middle and ring fingers. Each track can be controlled by three right hand gestures: closing the hand to start recording audio coming from the microphone, opening it to start looping the recorded sound and waving it to stop the current loop. Once stopped it can’t be played back again and a new loop should be recorded from scratch. This creates a very intuitive system, with hand gestures whose meaning tends to be transparent to both the performer and the public.

To train a system with such gesture recognition skills I employed Fiebrink’s open-source software called “Wekinator”. It comes with a series of “precoded” algorithms for machine learning that can be easily employed to build any kind of interactive musical instrument. It receives floating point numbers as input from any device or software capable of sending OSC[[1]](#footnote-1) and outputs the result of any one of three kinds of processes, each one with their own algorithms to choose from. Regression, sending continuous data; classification, labeling individual gestures and therefore discrete; and dynamic time warping, measuring the similarity between two sequences as they unfold in time. For the third study I employed the latter one, as it can be trained to perceive time-based gestures independently of the speed they are performed.

However, the training is not flawless. The system can classify gestures incorrectly depending on a long series of variables. It can be influenced by the length of the training data, complexity of gesture, slight variations in the position of the sensor, unforeseen secondary gestures and the threshold chosen to confidently choose out of multiple options. Successful training can be a time consuming and frustrating endeavor, and even after achieving satisfying results during a training session I often had to retrain it after picking it up the next day. Furthermore, every person using the instrument would have to train it first, which involves some troubleshooting.

This kind of training is useful when the expected result is an instrument for musical performance adapted to the idiosyncrasies of a particular individual and not that much as a general-purpose controller for several users for interactive computer music or as part of an installation. That being said, the gestures I chose for this study are simple enough and can be easily trained, and the low number of them also helps with accuracy. I consciously sent a different floating-point number for each hand as part of the training data, even though it could infer this from the position of my fingers I found that some redundancy in the system helps it make better decisions. This makes the role of both hands clear for everyone involved, including the software.

1. Open Sound Control. A protocol usually employed for musical performance. [↑](#footnote-ref-1)