Abstract

This paper describes a study into the realisation of a new method for capturing 3D sound control data. It expands on our established practice with the radiodrum 3D input device by incorporating a computer vision platform that we have developed using the Xbox Kinect motion sensing input device. It also extends our previous work in systemizing a low-cost, open-source 3D gesture sensor system, as a novel method for musically oriented human computer interaction. We discuss in detail the development process and the system performance in different scenarios and outline the future directions of this development.

1. Introduction

Although instruments of the modern symphony orchestra have reached maturity, musical instruments will continue to evolve. A significant area of development is in electro-acoustic instruments, combining natural acoustics with electronic sound and/or electronic control means, also called hyperinstruments [7]. The evolution of new musical instruments can be described in terms of both the sound generator and the controller. Traditionally, these separate functions are aspects of one physical system; for example, the violin makes sound via vibrations of the violin body, transmitted via the bridge from the strings, which have been excited by the bow or the finger. The artistry of the violin consists of controlling all aspects of the strings vibrations [4]. The piano is a more complex machine in which the player does not directly touch the sound-generating aspect (a hammer hitting the string), but still the piano has a unified construction in which the controller is the keyboard, and is directly linked to the sound-generation. For hyperinstruments these two aspects are decoupled, allowing for the controller to have an effect that is either tightly linked to the sound produced (as any conventional acoustic instrument has to be) or can be mapped to arbitrary sounds.

Modern advancements in consumer based digital technologies are allowing for unprecedented control over sound in ways that make traditional methods seem primitive. What the traditional model provides in terms of performance is a context for proven-to-be-successful socially-binding uses of music. We believe that this information is critical to retaining and evolving essential methods of music production, dissemination and performance.

This research outlines our progress towards the development of a system for the design of digitally extended musical instruments and their utilization in multi-media performance contexts. A method is presented for expanding the 3D control paradigm we previously developed on the radiodrum (an electromagnetic capacitive 3D input device), using the Kinect (a motion sensing input device for the Xbox 360 video game console). We also examine the performance of the Kinect when adapted for use with a vibraphone as an example of effectively embedding new technologies onto traditional instruments.

We are an interdisciplinary group of designers, electrical engineers, computer scientists, musicians and craftsmen who design and build new electro-acoustic instruments, and have chosen specific traditional musical metaphors for our research because as performers we have an understanding of what works musically.

2. Previous Work

Previous areas of research include Randy Jone's Soundplane (a force-sensitive surface for intimate control of electronic music that transmits x, y and pressure data using audio signals generated by piezo input captured at a high audio sampling rate) [3], Adam Tindale's acoustically excited
physical models using his E-drumset (a piezo based drum surface that uses audio signals, created by actual drumming, to drive physical models) [7], and the radiodrum (an instrument that uses audio carrier waves transmitted by the tip of the drum stick to track the stick’s tip’s movement in 3D space through capacitive sensing using an antennae [5]).

Past work regarding control interfacing with the xylophone includes: Simmons Multimallet, Mallet Kat, Xylosynth, Marimba Lumina, The Deagan Electravibe, The Ludwig Electro-Vibe dedicated instruments and Ayotte’s, K and K Audio and Vanderplaas piezo based audio pick-up conversion systems and optional impulse to MIDI conversion kits. We have also developed our own similar prototypes using piezos. Conventional percussion based trigger interfaces typically use piezos but only detect onset and velocity. In this case most of the dynamic information contained in a waveform is lost. Here we explore utilizing the full audio waveform of the vibraphone coupled with extended control functions to also take advantage of the full range of motion involved in order to modulate the source audio.

3. Design

In order to take into account the specifics of musical interaction, one needs to consider the various existing contexts- sometimes called metaphors for musical control [8] where gestural control can be applied to computer music. In order to devise strategies concerning the design of new hyperinstruments for gestural control of sound production/modulation, it is essential to analyze the characteristics of actions produced by expert instrumentalists during performance [2]. The importance of the study of gestures in new hyperinstrument design can be justified by the need to better understand physical actions and reactions that take place during expert performance. The xylophone is marginalized in contemporary music for several reasons: the antiquated, bulky frame makes it hard to move, and its sound is generally difficult to amplify without expensive special equipment; however, it makes for good coupling of technology because of the very constant visceral movement required to produce sound. This is good for two reasons: the sound produced is well suited for conversion into control data and the motion involved in the process is ideal for computer vision recognition. Thus, along with furthering our 3D control practice, we subsequently seek to advance the role of the acoustic xylophone tradition to incorporate advanced HCI capabilities.

A first design goal was communication with the sensors. The use of a radiodrum as a controller was developed for use through an audio interface [5]. The first challenge of this musical interface was communicating with the Kinect controller. We chose to use the OpenNI API to communicate with the Kinect because it offered access to the Kinect’s embedded user tracking functionality. From there, we developed a C++ application that takes the 3D position data provided by the Kinect and sends the positions data as a stream of Open Sound Control (OSC) messages.

Once the position data is in the OSC format, many other music-related programs can receive and manipulate the data.

4. Experiment

In many ways, the radiodrum and Kinect are similar controllers. Both return a set of 3D positions in real time. However, there are some major differences between these two pieces of technology. In this experiment, we present some early experiments that aim to demonstrate some of the major differences between these sensors.

Figure 4 shows the basic layout of the hardware. The Kinect connects to the computer via USB, and the radiodrum via firewire through an audio interface. We’ve also connected a microphone to the audio interface, which we will use as a reference when comparing the reaction of the sensors - similar to [9].

Custom software was developed to record and compare data received from both sensors. The program flow is shown in Figure 2. A C++ program was written that takes the user tracking data made available by OpenNI and sends it as OSC data. A Max/MSP program then receives data from the audio interface and converts it into position data, up-samples the Kinect data, and saves all the data to a file. The transport of data between the two programs is notable as it may lead to increased temporal effects.

Various movements were captured in an attempt to demonstrate some of the observed effects we have come
across when using these sensors in a musical context. This work is not an attempt to show whether one device was superior to the other. Instead, we are more interested in comparing these sensors so that data can be more intelligently fused for better control over the instrument.

We experimented with practical applications using the Kinect in a musical scenario. Using a vibraphone as the sole sound source, we arranged a system for improvisation that would take advantage of the new extended capabilities. We used OpenNI, OpenCV, and Max/MSP for software integration and the RME Fireface 800 and Macbook Pro hosting Ableton Live for audio processing. The Vibraphone audio channel went to the Master bus with a slight bit of equalization and compression, it was also routed to Auxiliary sends 1 and 2. On Aux Return 1 was a harmonic resonance filter plug-in. The X axis in the right hand controlled the tonic of the 7 voice harmonic scale pre-programmed into the filter. Moving the mallet to the left would lower the tonic, as would the inverse-moving the mallet to the right, would raise the pitch. The Y axis controlled the global frequency gain of the filter, allowing a performer to over-accentuate the high frequencies of the filtered audio by raising the mallet and boosting low frequencies as the mallet is lowered. Aux Return 2 hosted a bit reduction plug-in that was used as a type of digital distortion, much the way a guitarist would use an analog fuzz pedal. Except, in this case, the Ghanaian Gyil was a direct reference in that in a traditional context the Gyil’s resonators (made of gourds) would have holes drilled into them with spider egg casing stretched over them resulting in an intentional distorting buzz of the acoustic sound. The X axis in the left hand controlled the ratio of the filter applied to the source signal. Moving the mallet to the left would reduce the effect resulting in a dry source signal, moving the mallet to the right, would increase the ‘wetness’ of the filter effect. The Y axis controlled the rate of reduction. The bit rate decreases when the mallet is raised resulting in more distortion, lowering the mallet results in less distortion. These effects are calibrated so that the default playing position of the vibraphone results in no processing so they react only to extended gestures.

The audio from the acoustic vibraphone was captured with an Akai 414 microphone and input into an RME Fireface 800 connected to a Macbook Pro hosting Ableton Live. The Vibraphone audio channel went to the Master bus with a slight bit of equalization and compression, it was also routed to Auxiliary sends 1 and 2. On Aux Return 1 was a harmonic resonance filter plug-in. The X axis in the right hand controlled the tonic of the 7 voice harmonic scale pre-programmed into the filter. Moving the mallet to the left would lower the tonic, as would the inverse-moving the mallet to the right, would raise the pitch. The Y axis controlled the global frequency gain of the filter, allowing a performer to over-accentuate the high frequencies of the filtered audio by raising the mallet and boosting low frequencies as the mallet is lowered. Aux Return 2 hosted a bit reduction plug-in that was used as a type of digital distortion, much the way a guitarist would use an analog fuzz pedal. Except, in this case, the Ghanaian Gyil was a direct reference in that in a traditional context the Gyil’s resonators (made of gourds) would have holes drilled into them with spider egg casing stretched over them resulting in an intentional distorting buzz of the acoustic sound. The X axis in the left hand controlled the ratio of the filter applied to the source signal. Moving the mallet to the left would reduce the effect resulting in a dry source signal, moving the mallet to the right, would increase the ‘wetness’ of the filter effect. The Y axis controlled the rate of reduction. The bit rate decreases when the mallet is raised resulting in more distortion, lowering the mallet results in less distortion. These effects are calibrated so that the default playing position of the vibraphone results in no processing so they react only to extended gestures.

The specific mappings and filter parameters chosen were not arbitrary, but rather specific to the researcher’s artistic practice. Being both a sound designer and computer musician, the researcher is also a vibraphonist and thus chose intuitive mappings based on specific vibraphone techniques within a given original composition, and subsequently chose the extended digital sound design parameters
based on the familiarity of both the music and instrument’s natural characteristics, having a background with the devices, their workings and characteristics as well.

4.1. Latency

Humans can control transient events with a relatively high speed. This is demonstrated in the percussive technique known as the flam, where trained musicians can play this gesture with a 1ms temporal precision [8]. It is also important to look at the accuracy of events. Delays experienced by the performer will change the perceived responsiveness of the musical instrument, a major consideration for musicians.

A basic difference between these two sensors is the vast difference in the frame rate of the captured data. The radiodrum sends continuous signals to an audio interface, and the sampling rate of the data is determined by the audio interface. For this experiment, we used a frequency of 48000 Hz, but higher rates are possible. Conversely, the Kinect outputs position data at approximately 30Hz, the most stark difference in the capabilities of the sensors.

Figure 5. Demonstration of Latency for the Radiodrum and Kinect

We begin by demonstrating this latency by holding the tip of the radiodrum stick, and hitting the surface of a frame drum that has been placed on the surface of the radiodrum. We now have the output of three sensors to compare. The microphone has very little delay, and the auditory event of the frame drum being hit will occur before these events are seen by the gesture capturing devices. For our purposes, the audio response is considered a benchmark. The radiodrum will capture the position of each drumstick during this motion, and the Kinect will capture the position of the user’s hands. We performed simple piece-wise constant upsampling to the Kinect data, so the low framerate is evident.

As seen in Figure 5, the Kinect displays a significant amount of latency. A small amount of this could be attributed to the data transport, but the slow frame rate makes it nearly impossible to detect sudden events like the whack of a mallet.

Figure 6. Captured Motion of Four Drum Strikes

Although small temporal changes will not be detected by the Kinect, capturing slower movements is still possible. The following plot shows four discrete hits of the drum. Although the Kinect would not be able to use this information to produce a responsive sound immediately, we could still perform beat detection to determine the tempo a performer is playing at. However, it is already a sufficient alternative to the traditional “fader” style mapping parameter, as we have shown in our use here for this example.

4.2. Range

The choice of mapping for gestures onto or into audio data has also been a source of significant attention. How much apparent change should a movement produce?

Figure 7. Radiodrum Viewable Area

First, we examine the range of motion for both sensors. The radiodrum will only return consistent position data while the drum sticks are in an area close above the
surface of the sensor. It will also tend to bend the values towards the center as the sticks move farther above the surface.

Perspective viewing gives the Kinect a much larger viewable area. The Kinect's depth sensor's field of view is 57 degrees in the horizontal direction and 43 degrees in the vertical direction. This means that at closer ranges, the Kinect cannot detect objects far to the sides of the camera whereas when depth is increased, objects far from the center of view may be detected.

Figure 8. Kinect Viewable Area

To demonstrate the constriction on possible movements recorded by the radiodrum, we recorded the captured output of a user moving their hand back and forth while holding the radiodrum stick. As you can see, the Kinect is able to capture a much larger range of gestures.

Figure 9. Horizontal Range of both controllers

5. Future Work

We have shown that there is significant latency and temporal jitter from the Kinect data relative to the signals from the radiodrum. This makes direct fusion of the data difficult except for slow movements. One potential way to help resolve this issue is to extend the body model to include more of the physics of motion. The current body model is largely based on just the geometry of segments (a kinematic description) whereas a full biomechanical model would include inertial (kinetic) parameters of limb segments as well as local limb acceleration constraints. Once the biomechanical model is initialized with some slow moving data, it can be used to predict (feed-forward) the short term future motion and then the delayed motion data from the Kinect can be used to make internal corrections (feedback). Also, because internally the biomechanics body model will have estimates of limb segment accelerations, it would be relatively easy to incorporate data from 3D accelerometers placed on important limb segments (such as the wrist) to enhance motion tracking.

General free motion (motion with minimal physical contact) is usually imprecise in absolute terms as it relies on our proprioception (our internal sense of relative position of neighboring parts of the body) combined with vision to provide feedback on the motion. The exception to this is highly trained free motion such as gymnastics or acrobatics. Physical contact with a target reduces the spatial degrees of freedom and provides hard boundaries or contact feedback points. In the absence of this hard feedback, gestures performed in free space are going to be difficult to automatically recognize and react too, unless it is highly structured and trained. This requirement could significantly distract from the expressive nature of a performance.

One of our goals is to go beyond simple static one-to-one mappings and incorporate Machine Learning for gesture recognition. While one-to-one mappings are by far the most commonly used, other mapping strategies can be implemented. It has been shown that for the same gestural controller and synthesis algorithm, the choice of mapping strategy became the determinant factor concerning the expressivity of the instrument [6]. We have been actively developing a systemized mapping strategy for many years and will use this as an apparatus to inform and evaluate our advancements with the Kinect. Because the proposed future system will also include a biomechanical body model of the motion, it should be possible to predict potential inertial trajectories (gestures) in real-time as the trajectory evolves. This would allow the system to continuously predict the likely candidate match to the gesture and output an appropriate response. The trajectory mapping will need to be robust and relatively invariant to absolute positions and rotations.

Figure 10. Mapped Centroid (black circles) of Mallet Heads

Another scenario that is in development but has yet to be implemented involves extending the predicted movements from the kinect to include accurate tracking of the tips of a mallet. Verifying the accuracy and precision of mallet
tip position data acquired from both the radio drum and the Kinect would also be vital when effectively modeling of an interactive virtual marimba. As previously established, the radiodrum is accurate near the surface of the instrument, but position data veers towards the center of the rectangular surface as the tip is moved away. Preliminary measurements ranging from half a meter to a full meter away from the plane containing the Kinect’s cameras demonstrated that depth values increase linearly \( (R^2 = 1) \) along a straight line perpendicular to said plane. However, the slope of the line was not 1. Rather, it varied from 1.01 to 1.05. In cases where only the relative change in position matters, such a value would not strongly affect results. When the absolute position is desired, as in our application, it means the deviation from the actual value is as much as 5 cm at a depth of 1 meter. Another potential area of exploration involves comparing the three-dimensional coordinate measurements from both the radiodrum and the Kinect with a ground truth and attempting to compensate for the disparity.

6. Conclusion

Our Kinect instrument will be compatible with virtually all MIDI hardware/software platforms so that any user could develop their own custom hardware/software interface with relative ease. The externals we will develop in Max/MSP for use in Max4Live will enable anyone to plug in a Kinect and adapt our language of 3D sound control to their own practice with minimal configuration. This has many implications, putting technology that previously was restricted to research purposes into the common computer musician’s hands. Our shared belief is that these new instruments have a legitimate place [4] with potential to become part of an embedded conventional musical practice, not just a research curiosity. While hyperinstruments might seem niche or esoteric at this point [1], historic innovations such as the leap from monophonic to polyphonic music, electrical amplification of the guitar, and computers in the recording studio all brought skepticism, eventually becoming mainstay practices.

References


