3D Positional Movement Interaction with User-Defined, Virtual Interface for Music Software: MoveMIDI

Timothy Arterbury
Baylor University
Waco, TX, USA
timothy_arterbury@baylor.edu

G. Michael Poor
Baylor University
Waco, TX, USA
michael_poor@baylor.edu

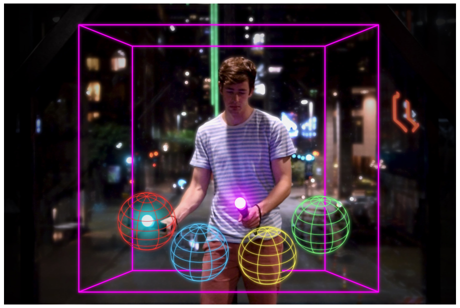

Figure 1: Representation of MoveMIDI's virtual MIDI controller interface. Sphere Icon: ⒸⓘBy Gonzalo Bravo from the Noun Project.

Figure 2: Example of a traditional MIDI controller with piano keys, knobs, and pads. Photo by Cedrik Malabanan from Unsplash.

ABSTRACT

This paper describes progress made in design and development of a new digital musical instrument (MIDI controller), MoveMIDI, and highlights its unique 3D positional movement interaction design differing from recent orientational and gestural approaches. A user constructs and interacts with MoveMIDI's virtual, 3D interface using handheld position-tracked controllers to control music software, as well as non-musical technology such as stage lighting. MoveMIDI's virtual interface contributes to solving problems difficult to solve with hardware MIDI controller interfaces such as customized positioning and instantiation of interface elements, and accurate, simultaneous control of independent parameters. MoveMIDI's positional interaction mirrors interaction with some physical acoustic instruments and provides visualization for an audience. Beta testers of MoveMIDI have created emergent use cases for the instrument.

KEYWORDS

Music; MIDI Controller; Positional Interaction; User Defined Interface; RBI; MR; VR; AR;

CHI'19 Extended Abstracts, May 4–9, 2019, Glasgow, Scotland UK

ACM ISBN 978-1-4503-5971-9/19/05.
https://doi.org/10.1145/3290607.3312954

Figure 3: Example of a MIDI controller with pads, knobs, sliders and buttons. Photo: *CNTRL:R* ©ⓘ Livid Instruments from Flickr

INTRODUCTION

Computers have become a prominent tool for creating and performing music. Music software is used not only to record performances of acoustic instruments, but also to record and play virtually simulated musical instruments. Natural and expressive interaction to control music software is key for an electronic musician to perform proficiently with virtual instruments. This paper explores 3D position-tracked movement interaction for music software with a prototype instrument, MoveMIDI.

To play and control virtual software instruments, musicians interact with hardware instruments called MIDI Controllers which send digital MIDI signals to virtual instrument software to produce sound from a computer. MIDI (Musical Instrument Digital Interface) is a standard digital communication protocol for music. By using a MIDI Controller to send MIDI signals to virtual instrument software, a musician can play musical notes, manipulate timbral characteristics of the sound, control sequences of musical events, etc. The advantage of this MIDI abstraction is that the same signals used to play one virtual instrument sound can be used to play another. As long as the musician knows how to play their MIDI Controller, then they know how to play any virtual instrument sound. MIDI can also be used for some non-musical applications such as controlling stage lighting.

The MIDI Controller is essentially the bridge in interaction between a user and music software. A user may interact with music software using conventional input methods such as the keyboard and mouse, but the goal of a MIDI Controller is to make this interaction more natural and playable as a musical instrument. To bring playability and familiarity to music software, some MIDI Controllers may have a set of piano keys, while others mimic interaction with wind instruments. Other controllers do not mimic traditional musical instruments and may have an array of pads, knobs, and sliders.

In an exploration of user interaction with music software, Arterbury began developing MoveMIDI in December 2016. MoveMIDI uses 3D positional tracking technology to track the location of handheld controllers in space allowing for a user's movements to be interpreted relative to a positional, virtual interface around them. MoveMIDI allows the user to define a virtual, 3D MIDI controller interface layout, map that interface to various functions in music software, and play that interface and mapping combination as a musical instrument. Essentially, users can create and manipulate music with positional arm movements. The initial implementation uses PlayStation Move controllers and the PlayStation Eye camera for tracking 3D position of handheld controllers. Arterbury developed a MoveMIDI desktop GUI application that communicates with the tracking hardware, visualizes the virtual interface, allows the user to specify MIDI mappings, and generates MIDI signals. The software uses the PS Move API by Thomas Perl to communicate with the PlayStation hardware. MoveMIDI for PlayStation Move hardware is now in a beta testing stage with around 19 beta testers. In the future, MoveMIDI intends to support other 3D positional tracking hardware such as VR, AR, and MR systems.

Figure 4: User-defined 2D interface created with TouchOSC based on a drawn sketch.

RELATED WORK

Motion Interaction for Electronic Instruments

The idea of using physical human motion to control electronic musical instruments is not a new idea. In 1920, Leon Theremin invented the Theremin, a musical instrument played by moving one's hand towards and away from two fixed electronic antennae to affect pitch and amplitude of sound. Recently, steps have been taken to use motion interaction to control MIDI. The Wave MIDI ring tracks **orientation** (pan, tilt, and roll) of the user's hand to affect parameters in music software [1]. More than just orientation, the Mi.Mu Gloves, created by Imogen Heap, and the Satta MIDI Glove, based on research by Elena Jessop, detect changes in **gestural** hand postures to manipulate music software [6, 7]. Josh Gledhill's Kinect2OSC uses recognition of body gestures to manipulate music software [4]. A natural extension of these motion interaction practices with music software is **positional interaction**, allowing a user to interact with music using the location of a their hands in 3D space. Positional interaction is the area of motion interaction MoveMIDI explores.

User-Defined Interfaces

An interesting group of MIDI Controllers are those whose interfaces are customizable, modular, or user-defined. These controllers also allow users to customize the layout of interface elements. For example, ROLI's Blocks are a set of modular hardware pieces which can be put together into a custom MIDI controller interface. Another example is TouchOSC, an iOS app which allows the user to define and interact with their own 2D controller interfaces on a touchscreen. This is quite similar to MoveMIDI's user-defined interface, yet MoveMIDI brings this concept to the 3D domain.

MOVEMIDI'S UNIQUE APPROACHES

Dynamic, User-Defined Virtual Instrument

MoveMIDI allows the user to define a virtual instrument construct with which they will interact. This instrument definition consists of three parts: the **virtual interface layout**, the **MIDI mapping** of that interface, and the **MIDI target** which receives the MIDI messages. First, the user defines the 3D positions and sizes of virtual interface elements with which they will interact. Second, the user creates a mapping from actions done on the virtual interface to MIDI signals which should be sent to the MIDI target. Third, the user chooses the target to receive the MIDI messages. The target is most often a piece of music software which can generate sound from the MIDI messages it receives.

To construct the virtual interface layout, the user positions multiple instances of the two types of virtual interface elements. **The first interface element is a Hit Zone**. Hit Zones are spherical volumes in 3D space. When a user hits a Hit Zone with their handheld controller, the zone will generate an associated MIDI Note message. MIDI Note messages are often mapped to trigger musical notes

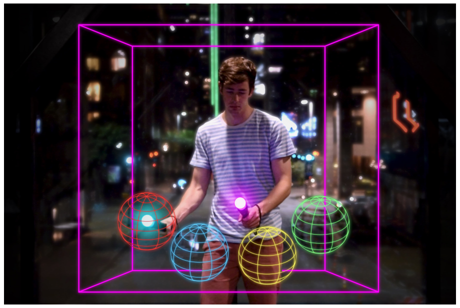

Figure 5: Representation of MoveMIDI's Hit Zones as 3D spheres and Morph Zone as a cuboid volume. Sphere Icon: ©ⓘ by Gonzalo Bravo from the Noun Project.

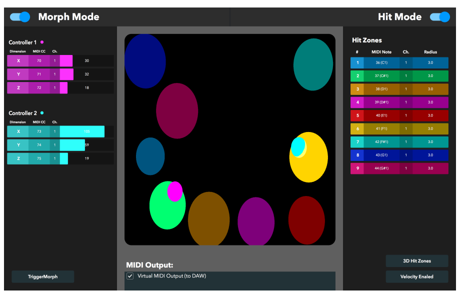

Figure 6: MoveMIDI's application GUI with visualization of 3D Hit Zones and handheld controller positions and MIDI mapping settings.

or audio samples. The harder a user hits a Hit Zone, the louder or more harsh the sound becomes. Hit Zones are great for "air-drumming". **The second interface element is a Morph Zone**. Morph Zones are cuboid volumes in 3D space which a user may move their hands through to send MIDI Control Change messages which are often mapped to manipulate timbral parameters of sound. A user can map each of the X, Y, and Z components of their hand's location in a Morph Zone to independent parameters. Morph Zones are great for manipulating effects and are advantageous for manipulating many parameters simultaneously, yet independently, and accurately. On traditional MIDI controllers, users are limited to the number of knobs or sliders they can manipulate accurately at once. With MoveMIDI, 3 independent parameters can be manipulated simultaneously per handheld controller as accurately as the user is able to move their hands through space.

MoveMIDI's entire instrument definition (consisting of interface layouts, mappings, and targets) is dynamic and recallable. Users can save and swap definitions to vary instrumentation during performance. This is advantageous over static physical hardware interfaces of traditional MIDI Controllers.

Positional Interaction

The primary difference between MoveMIDI and other motion based MIDI controllers such as Mi.Mu Gloves is MoveMIDI's use of positional interaction. MoveMIDI interprets users' movements with handheld controllers relative to a positional, virtual interface around them. Other motion based MIDI controllers such as Mi.Mu Gloves make use of gestural and orientational interaction allowing the user to manipulate music software using memorized hand gestures or orientational hand changes. The difference between gestural and positional is easily seen in the use case of "air-drumming". In a positional approach, the user places a set of positional triggers in various disperse locations, mapping each zone to a unique drum sound. The user then strikes the zones as if they were physical drums to play the sounds. With a purely gestural controller it is not possible to vary location of gestural interactions to result in different sounds. A user could map a strike gesture of each hand to different drum sounds, but to add additional drum sounds they would need to define variations of the gesture by modifying orientation or direction of the hand. The user could not simply translate the same gesture to a different location in a room to result in a different drum sound or mapping.

While gestural and orientational approaches are useful, there are advantages to a positional approach. Positional interaction is a reality-based approach which may be more natural to learn. We all interact with the physical world in a positional way. MoveMIDI's positional approach aligns with themes of the HCI framework, RBI (Reality-Based Interaction). The positional interaction style may be natural for users due to their pre-existing knowledge of the physical, non-digital world, and their own body awareness relative to environmental objects [5]. Positional interaction eliminates some interface-specific skills the user would need to learn to interact with MoveMIDI. A gestural approach requires learning and memorization of gestures for interaction.

In a user survey about a gestural music controller by Josh Gledhill, participants were asked how they would map various gestures to music software functions. 32 of 41 participants chose to map a strike gesture to the triggering of an audio sample. This shows that these users naturally want to trigger sounds with a strike gesture. The positional approach extends this strike movement to be used for interaction with positional triggers, similar to physical acoustic instruments such as drums. "A strike gesture could represent the action of a performer striking a percussive instrument" [4].

A user's prior knowledge of physical acoustic instruments strengthens the positional interaction approach. Many acoustic instruments require positional interaction. On a piano, low pitched notes are on the left of the keyboard and high pitched notes are on the right. Each drum in a drum-set is a positional element. Positional movement of a bow relative to the strings of a cello creates sound.

This relation to acoustic instruments can be a metaphor which a performer employs to contribute to the audience's understanding of a performance. MoveMIDI's strike and wave motions can provide a familiar visual metaphor with acoustic instruments for the audience, as shown below in Christo's example. "Metaphor enables device designers, players, and audience to refer to elements that are 'common knowledge' ... through metaphor, transparency increases, making the device more expressive" [3]. "The creation of meaningful and perceivable connections between human action and sound has been identified as a key point for making a performance convincing for the audience" [2] [8].

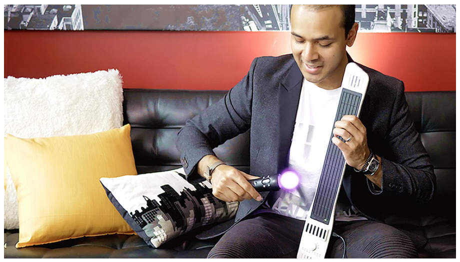

Figure 7: Sanjay Christo playing MoveMIDI and the Antiphon Instrument 1 as a virtual cello. Photo by Sanjay Christo from YouTube video thumbnail.

USER EXPERIENCES AND FEEDBACK

Sanjay Christo - Virtual Cello Playing & Movie Sound Effects

Sanjay Christo is a musician and YouTuber who reviews music tech. While beta testing MoveMIDI, Christo created a video demonstration using MoveMIDI's Morph Mode as an expressive extension to the Artiphon Instrument 1. Christo used the two instruments to mimic the interaction of playing a cello, using the Artiphon Instrument 1 as the metaphorical neck of the cello to select musical notes and using MoveMIDI as a metaphorical bow for the cello to affect timbral characteristics.

Christo created another demonstration video where he used MoveMIDI's Morph Mode to manipulate intensity and tonal composition of a cinematic score. He mapped MoveMIDI's X and Y positional components to parameters in Native Instruments Thrill music software to record changes to the music using his movement as he watched a movie scene for reference. Techniques such as this are used by post-production engineers for some scoring and sound effect automation for movies.

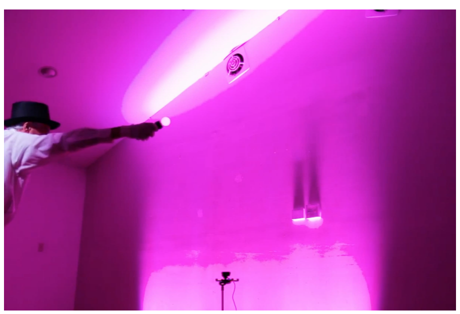

Figure 8: Jeff Firman controlling stage lighting with MoveMIDI. Still image by Jeff Firman from YouTube video.

Jeff Firman - Controlling Stage Lighting

Jeff Firman designs stage lighting for performance. While beta testing MoveMIDI, he created a video demonstration using MoveMIDI to manipulate and trigger stage lighting sequences. With MoveMIDI, he was able to trigger light and sound changes simultaneously using his movements, revealing an interesting use case for live musical performance. He connected his stage lighting software as a MIDI target of MoveMIDI to control the lighting since this tech also supports MIDI.

As feedback, Firman said that visualization of MoveMIDI's virtual interface with the computer GUI application was difficult to interpret along the Z-Axis (toward and away from the tracking camera). The beta hardware he was using at the time did not support an HMD. The software GUI provided a fixed-view 3D visualization of the virtual interface and the positions of the controller's being held by the user. This request validates the continued development of versions of MoveMIDI which supports various VR/AR/MR HMD hardware. He also requested the ability to label virtual interface elements.

CONCLUSION AND FUTURE WORK

Although MoveMIDI and its concepts are a work in progress, past research and user usage/feedback suggest that 3D positional interaction with a user-defined, virtual interface may benefit human interaction with digital musical instruments. MoveMIDI's positional interaction differs from recent gestural and orientational interaction approaches. MoveMIDI's abstraction of a user-defined virtual instrument definition leads to emergent use-cases. To improve MoveMIDI, Arterbury plans to combine interaction styles, provide better visualization with VR/AR/MR, and explore better haptic feedback solutions. This work prompts an empirical study of interaction styles for music software. Arterbury plans to explore future use-cases of MoveMIDI in music education, accessibility, dance, and multi-person instruments. To learn more about MoveMIDI, visit https://movemidi.com.

REFERENCES

[1] 2018. Wave by Genki Instruments. (7 March 2018). https://youtu.be/Fl4s_IsOguQ?t=84

[2] Jerônimo Barbosa, Filipe Calegario, Veronica Teichrieb, Geber Ramalho, and Patrick McGlynn. 2012. Considering Audience's View Towards an Evaluation Methodology for Digital Musical Instruments.. In *NIME*.

[3] Sidney Fels, Ashley Gadd, and Axel Mulder. 2002. Mapping transparency through metaphor: towards more expressive musical instruments. *Organised Sound* 7, 2 (2002), 109–126.

[4] Josh Gledhill. 2018. *An Investigation into Developing a Standardised Mapping Framework for a Digital Musical Instrument.* Ph.D. Dissertation. Staffordshire University. https://drive.google.com/file/d/1uwh_OOaOh6ukSHAyZKyri79XFDzCLmee/view?usp=drive_open&usp=embed_facebook

[5] Robert JK Jacob, Audrey Girouard, Leanne M Hirshfield, Michael S Horn, Orit Shaer, Erin Treacy Solovey, and Jamie Zigelbaum. 2008. Reality-based interaction: a framework for post-WIMP interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 201–210.

[6] Elena Naomi Jessop. 2010. *A gestural media framework: Tools for expressive gesture recognition and mapping in rehearsal and performance.* Ph.D. Dissertation. Massachusetts Institute of Technology.

[7] Thomas J Mitchell, Sebastian Madgwick, and Imogen Heap. 2012. Musical interaction with hand posture and orientation: A toolbox of gestural control mechanisms. (2012).

[8] S. O'Modhrain. 2011. A Framework for the Evaluation of Digital Musical Instruments. *Computer Music Journal* 35, 1 (March 2011), 28–42. https://doi.org/10.1162/COMJ_a_00038