Sonification of a juggling performance using spatial audio

VOJTĚCH LEISCHNER and PAVEL HUSA, Czech Technical University in Prague, Czech Republic

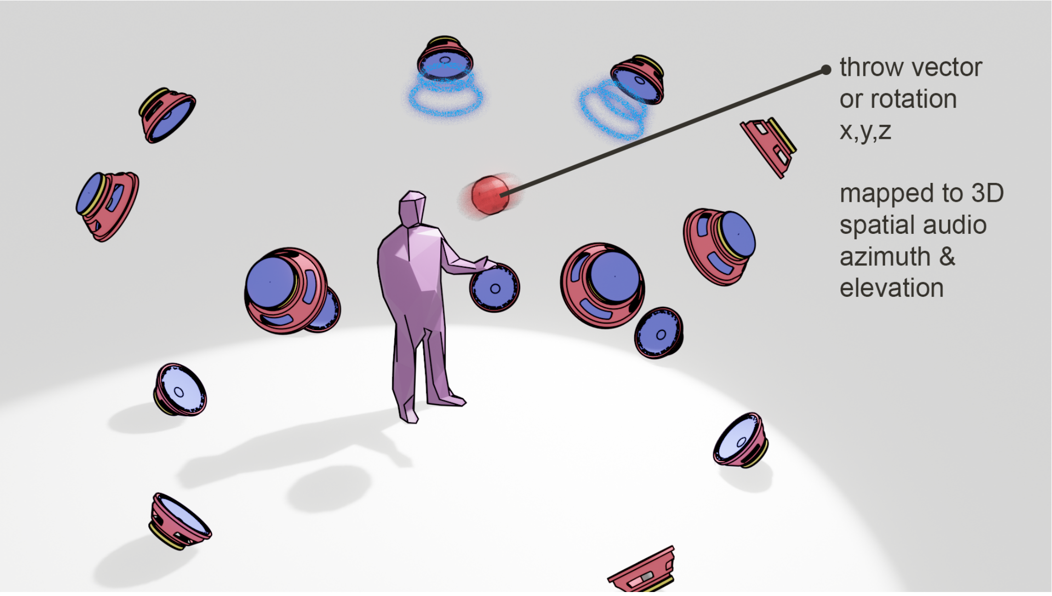

Fig. 1. Throw vector or rotation mapped to spatial audio azimuth and elevation angles. (©2023, Vojtěch Leischner)

We developed a new musical instrument by equipping juggling balls with accelerometers, gyroscopes, and WiFi sensors. The system measures acceleration and rotation, allowing us to distinguish between ball movement and static states. We can determine the direction of each throw based on the acceleration data.

A movement-driven instrument animates the virtual audio source position based on the throw direction. A different synthesizer is assigned to each ball, resulting in a synesthetic experience. The system integrates the juggling patterns simulator with OSC and MIDI API, allowing real-time rhythm generation and composition.

CCS Concepts: • **Human-centered computing → Sound-based input / output**; *Gestural input*.

Additional Key Words and Phrases: juggling, instrument, sonification, spatial audio, synthetizer, music, performance, haptic, interface, movement, postdigital, MIDI, OSC, audio

ACM Reference Format:
Vojtěch Leischner and Pavel Husa. 2023. Sonification of a juggling performance using spatial audio. *Proc. ACM Comput. Graph. Interact. Tech.* 6, 2, Article 20 (August 2023), 6 pages. https://doi.org/10.1145/3597619

Authors' address: Vojtěch Leischner, info@tricktheear.eu; Pavel Husa, husapave@fel.cvut.cz, Czech Technical University in Prague, Karlovo náměstí 13, Prague, Praha 2, Czech Republic, 121-35.

2577-6193/2023/8-ART20
https://doi.org/10.1145/3597619

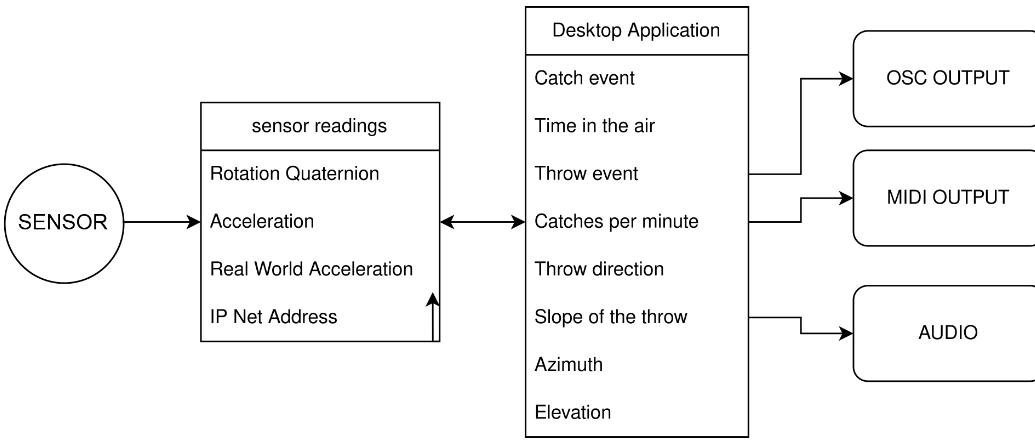

Fig. 2. Dataflow between sensor and the application (©2023, Vojtěch Leischner)

1 STATE OF THE ART

Till Bovermann et.al. created a similar setup for juggling sonification using multiple cameras and markers stuck to the juggling clubs [Bovermann et al. 2007]. Our proposed system is based on relative motion measurement rather than tracking absolute position in space. The advantage of cameras is that we can clearly define the relationships of props to each other. The limitation is mainly the required hardware and calibration setup that hinders the system's usability. Using the inertial measurement unit and WiFi microcontroller directly inside the balls, we can use the system independently of the current environment in all lighting conditions without the camera setup.

Another juggling tracking system was developed by Jesse Checkla et al. in 2015 at Cornell University [Checkla 2015]. The project aimed to detect sequences of juggling patterns or siteswaps. Their proposed system uses microcontrollers and accelerometers inside balls and sends data over 2.4 GHz radio. Their system focuses on visual LED feedback instead of audio. They use Matlab for analysis, and sequences are detected retrospectively. Our system differs in offering OSC and MIDI API, a juggling simulator, and mapping the data to musical compositions in real time.

Multi-instrumentalist Juzzie Smith [Smith 2023] uses "Chuka chucks" stainless steel bearings filled balls to act as rumba balls to sonify ball catches. Juggler Peter Aberg used hollow tubes in Tube Act performance [Aberg 2008]. Tubes produce sound due to air rushing through. Lastly, we should name Michael Moschen and his performance Triangle, where he juggles bouncing balls inside a triangle structure to produce various rhythms [Moschen 2007].

Many new musical instruments are created by adding electronic sensors. For example Extended Actuated Digital Shaker [Williams and Overholt 2017] is extending classical instrument using IMU and microcontroller inside. Spatial audio is getting more widespread thanks to recently developed tools [Leischner and Mikovec 2021] and plugins [Giller and Schörkhuber 2019] that made it easy to integrate.

2 DATA FLOW

The sensors are connected to the microcontroller with integrated WiFi, see Figure 2. We send UDP packets over WiFi containing the acceleration vectors and rotation quaternions. We subtract the gravitation vector from the acceleration and rotate it around the quaternion from the gyroscopes. The result is a direction vector relative to the observer reference frame. We receive the packets on a computer or mobile device that analyzes the data and sends out events such as catch and throw with time stamps over OSC and MIDI. Users receive events in their preferred software, such as Ableton, Reaper, and other DAWs. For testing, we have also included the MIDI sound engine directly in the application to showcase the possibilities without additional DAW software. Users can record the juggling session as CSV or MIDI files and instantly replay them. The replay can be used for archiving and as a simple loop station during live performance. We used the replay mode to compare different musical mappings on the same data set.

2.1 MIDI API

Users can select the MIDI input device to receive events from the GUI list. Throw and catch events are represented as NoteOn and NoteOff messages. Azimuth, altitude, and catches-per-minute are sent as the Controller Changes. Users can alter the id of the controller used to carry messages in the application settings.json file inside the data folder. All values are remapped from the 0 to 127 range.

2.2 OSC API

Users can define IP addresses and ports to be used by the OSC server. All values are sent in their original range. Below is the list of variable types and value ranges of the data:

- timestamp [int milliseconds from the first throw]
- event type, "throw" or "catch" [String]
- time in the air[int milliseconds]
- rotation Quaternion [float w, float x, float y, float z]
- throw direction vector [float x, float y, float z]
- azimuth [float 0-1 representing 0-360 degrees]
- altitude [float 0-1 representing -90/90 degrees]

2.3 Recording and Replay

Users can save the juggling sessions into MIDI or CSV files to replay later. MIDI files feature the same events as the live output. CSV files contain raw data from which other variables can be calculated during replay. You can use the CSV file for further analysis.

3 APPLICATION DEVELOPMENT

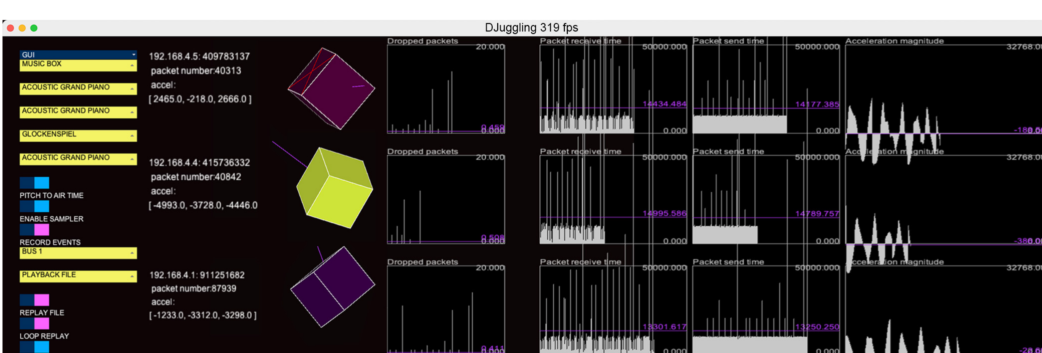

Fig. 3. Application GUI running on a computer (©2023, Vojtěch Leischner)

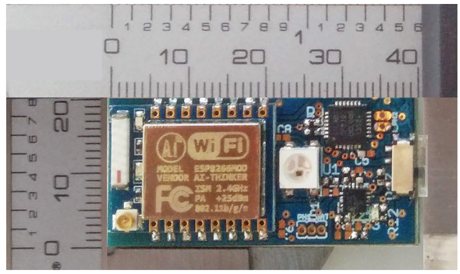

Fig. 4. Hardware prototype - battery powered WiFi microcontroller with IMU (©2023, Vojtěch Leischner)

3.1 Graphical User Interface

We have implemented a 3D visualization of rotation on cubes, and we visualize the direction of the throw by the cube's color. Furthermore, we showcase real-time graphs showing acceleration magnitude in time and individual UDP packet receiving times, see Figure 3. On the left side of the window, we feature interactive control over the integrated audio engine. Users can choose to disable it when using other sound engines. We added toggle controls to individual parameters, such as mapping pitch to the time spent in the air or choosing a musical instrument for each ball / sensor.

3.2 Hardware prototype

We created the custom hardware for the prototype, see Figure 4. The suitable alternative modules are LILIGO® TTGO TAudio V1.6 or M5stick with needed sensors on board. We use a widely available ESP8266 microcontroller and MPU6050 IMU unit on a custom PCB. We attached a 102535 Li-Po 800mAh battery to power the unit with a step-up and charging module TP4056. We also add LEDs for signaling the status of the device and absolute tracking with cameras if the user chooses to. Firmware was developed in Arduino IDE.

4 MAPPING AUDIO TO MOVEMENT

4.1 Events

We detect throw and catch per sensor and send a catches-per-minute event once a second. In the throw event, we include information about the throw direction. Direction is defined as azimuth and altitude rotations. In the catch event, we also send the time spent in the air in milliseconds.

4.2 Mapping Events to Music Parameters

We utilize the MIDI On and MIDI Off events to represent the throwing and catching of a ball, respectively. These events trigger and shape the sound of the instrument. We assigned a separate MIDI channel to each ball to create a custom sound design. We used Ableton 11 DAW [Hein 2021] to create generative musical compositions.

We can map the pitch to the airtime of the ball if we play the note when the ball is caught instead of thrown. The note length corresponded directly to the ball airtime. We implemented tempo-synced effects such as echo, sync pitch shift, and sidechain compression to enhance the rhythmical aspect of the music. Sidechain compression means modulating the volume using another signal e. g. bass drum. We did not quantize the incoming MIDI notes to keep rhythm variations resembling actual music performances whenever possible.

Another musical principle we used is arpeggio, meaning the sequence of predefined notes played while the ball is in the air instead of playing a single note. Using pentatonic scales yields a

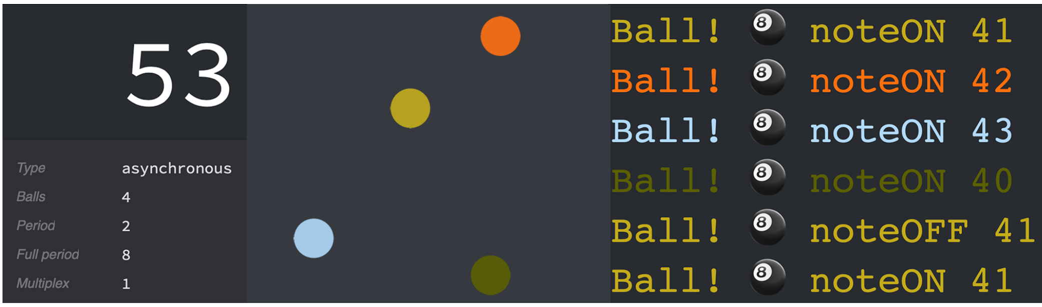

Fig. 5. Juggling simulator and rhythm MIDI sequence generator (2023, public domain)

more consonant output. The scale and dynamics can also be controlled by other means, such as a pre-prepared timeline in the Ableton session view.

This approach transforms juggling into more of an instrument, allowing the juggler to focus on overall dynamics and explore the relationship between patterns and sound.

4.3 Spatial Audio

Each ball in Ableton [Hein 2021] is connected to a distinct instrument and can modify its root note independently. Azimuth, and altitude data of the thrown ball are mapped to a spatial audio VST plugin Envelop for Live [Willits 2023], resulting in the 360° panning of the corresponding instrument. Other VST plugins offering azimuth and altitude rotations, such as IEM suite [Giller and Schörkhuber 2019] can also be used. Airtime, known after the ball is caught, is mapped to different aftereffects, such as pitched echo. We also map catches per minute averages to predict throwing cadence and patterns applied to overall dynamics. Users can map 3D rotation to the Ambisonics sound sphere as well.

4.4 Juggling simulator & sequencer

As a counterpart to sonifying physical juggling, we have developed a juggling simulator for testing different patterns and corresponding rhythms, see Figure 5. We can describe juggling with mathematical formulas [Shannon 1993] and verify if the input pattern is feasible. In our simulator, we propose a novel approach for creating rhythmic patterns using siteswap notation, a popular juggling notation [Sethares 2007]. We adapted an open-source web library for siteswap animation, sani.js, and added a MIDI functionality using the WEBMIDI.js library [Côté 2022; neunato 2022]. With this tool, we can investigate the relationship between siteswap notation and polyrhythms generated.

We have developed this tool as a standalone drum sequencer, bringing a physical aspect to the rhythm-creation process. This approach can enhance state-of-the-art sequencers, such as linear or euclidian [Toussaint 2005], by introducing a new and unique way of creating rhythms.

5 FUTURE RESEARCH

Complex juggling patterns can emerge in multi-user collaboration, such as passing balls between multiple jugglers. In a physical audio simulation, we propose using juggling throws as an audio particle emitter, where virtual particles produce sound on reflections in a virtual room.

To verify spatial audio performance on a 20-channel loudspeaker array and develop a public performance for the Ottosonic Festival [Mitterhuber 2023], we are collaborating with the Tangible Music Lab in Linz [Kaltenbrunner 2023]. Additionally, user studies will be conducted to validate the design.

A better hardware unit with an ESP32 and MPU9250 can improve performance and precision. Furthermore, we aim to port the application onto a Raspberry Pi (embedded mini PC) for increased accessibility. To facilitate sensor data parsing, we are writing a MaxMSP plugin for the Ableton Live DAW, a widely used software that can attract more users. Ultimately, we aim to integrate juggling sonification into the music composing pipeline, expanding composers' toolkits.

6 CONCLUSION

We have created a new expressive musical instrument that maps movement to sound. The prototype follows the post-digital approach inspired by reacTable synthesizer [Jordà et al. 2007]. Movement-based music controller offers a more intuitive way to create music[Enrique et al. 2021]. It combines the physical and tangible with the digital. In this way, we achieved an instrument that is intuitive to use and gives enough challenge to master. By the nature of the instrument, it can be modified and adjusted to fit any musical genre. The MIDI interface makes it easy for musicians to integrate it without coding. We offer a new rhythm generator algorithm to explore. More importantly, non-musicians like dancers, jugglers, and generally, performers can become musicians and collaborate with composers more closely.

ACKNOWLEDGMENTS

This research has been supported by the project funded by a grant SGS22/172/OHK3/3T/13 and by TACR RCI (CZ.02.1.01/0.0/0.0/16 019/0000765).

REFERENCES

Peter Aberg. 2008. Tube act. https://www.youtube.com/watch?v=I0nl3244VQ4

Till Bovermann, Jonas Groten, Alberto deCampo, and Gerhard Eckel. 2007. Juggling sounds. In *Proceedings of the 2nd International Workshop on Interactive Sonification*.

Jesse Haydn Checkla. 2015. *LED Juggling Balls with Pattern Detection*. Ph. D. Dissertation. Cornell University.

Jean-Philippe Côté. 2022. User-Friendly MIDI in the Web Browser. In *NIME 2022* (The University of Auckland, New Zealand). PubPub. https://doi.org/10.21428/92fbeb44.388e4764

Tomás Enrique, Thomas Gorbach, Hilda Tellioğlu, and Martin Kaltenbrunner. 2021. Embodied Gestures: Sculpting Energy-Motion Models into Musical Interfaces. In *NIME 2021*. PubPub. https://doi.org/10.21428/92fbeb44.ce8139a8

Peter Maximilian Giller and Christian Schörkhuber. 2019. A Super-Resolution Ambisonics-to-Binaural Rendering Plug-In. *Fortschritte der Akustik-DAGA, Rostock* (2019).

Ethan Hein. 2021. Ableton Live 11. *Journal of the American Musicological Society* 74, 1 (2021), 214–225.

Sergi Jordà, Günter Geiger, Marcos Alonso, and Martin Kaltenbrunner. 2007. The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces. In *Proceedings of the 1st international conference on Tangible and embedded interaction*. 139–146.

Martin Kaltenbrunner. 2023. Tangible music lab. https://tamlab.kunstuni-linz.at/

Vojtech Leischner and Zdenek Mikovec. 2021. Spatial Audio Music Player for Web.. In *RoCHI - International Conference on Human-Computer Interaction*. MATRIX ROM, 125–129. https://doi.org/10.37789/rochi.2021.1.1.19

Manu Mitterhuber. 2023. Festival. https://www.alterbauhof.at/ottosonics/festival/

Michael Moschen. 2007. Michael Moschen performs the Triangle. https://www.youtube.com/watch?v=qjHoedoSUXY

neunato. 2022. *sani.js*. https://github.com/neunato/sani.js original-date: 2017-10-09T14:37:30Z.

William A. Sethares. 2007. Visualizing and Conceptualizing Rhythm. In *Rhythm and Transforms*. Springer, 23–51. https://doi.org/10.1007/978-1-84628-640-7_2

Claude E Shannon. 1993. Scientific aspects of juggling. , 924 pages.

Juzzie Smith. 2023. Juzzie Smith. https://juzziesmith.com/

Godfried Toussaint. 2005. The Euclidean algorithm generates traditional musical rhythms. In *Renaissance Banff: Mathematics, Music, Art, Culture*. 47–56.

Peter Williams and Daniel Overholt. 2017. Beads Extended Actuated Digital Shaker. (2017). https://doi.org/10.5281/ZENODO.1176153 Publisher: Zenodo.

Christopher Willits. 2023. Listen together. https://envelop.us/