Soundscape and sound space in the SoundThimble real-time gesture sonification framework

Grigore Burloiu, Ștefan Damian, Bogdan Golumbeanu
CINETic UNATC
Bucharest, Romania
gburloiu@gmail.com

Valentin Mihai
University "Politehnica" Bucharest
Faculty of Electronics, Telecommunications and IT
Bucharest, Romania

ABSTRACT

We introduce *SoundThimble*, a platform for sonic interaction based on the relationship between human motion and virtual objects in 3D space.

A Vicon motion capture system and custom software are used to track, interpret and sonify the movement and gestures of a performer relative to an object.

We define three possible interaction dynamics, centred around object searching, manipulation and arrangement. We illustrate the resulting possibilities for layered structures and extended perception and expression.

The software developed is open source and portable to similar hardware systems, leaving room for further extension of the interaction mechanics.

CCS CONCEPTS

• **Applied computing** → **Sound and music computing**; • **Computing methodologies** → **Motion capture**; • **Human-centered computing** → *Gestural input*; *Auditory feedback*;

KEYWORDS

Sonification, motion capture, gesture spotting, interactive installation, synthesis

ACM Reference format:
Grigore Burloiu, Ștefan Damian, Bogdan Golumbeanu and Valentin Mihai. 2017. Soundscape and sound space in the SoundThimble real-time gesture sonification framework. In *Proceedings of AudioMostly conference, London, UK, August 2017 (AM'17)*, 5 pages.
DOI: 10.475/123_4

DOI: 10.475/123_4

1 INTRODUCTION

High resolution three-dimensional motion tracking is traditionally used for animation in film and games, as well as for life sciences research and engineering applications [18]. This technology has long been mined by the audio computing community, although in many early cases, technical limitations meant that the motion data transmission and the sound generation processes were not simultaneous [3, 10].

The *SoundThimble* project harnesses current motion capture technology and gesture detection algorithms to enable new modes of real-time sound exploration. Our aim is to push beyond the standard paradigms of isolated body motion audification [3, 10] or sound control interfaces [4, 12], towards deeper narrative structures coupled with layered arrangement of music patterns.

Our implementation uses a state-of-the-art Vicon motion capture system[1] containing eight Vantage 5-megapixel infrared cameras and two Bonita video cameras. Since the open-source software developed in this project[2] is built around Vicon's Datastream SDK,[3] the platform can be ported to both older and future Vicon-based systems.

In the remainder of the paper, we review relevant literature and technology (section 2), we describe the *SoundThimble* concept and implementation (sections 3, 4), and finish with a survey of future challenges and perspectives (section 5).

2 STATE OF THE ART

Sonification, as the auditory representation of a datastream, is a rich tool for interpreting human movement [7]. From the musical perspective, infrared motion capture systems have been revealed as a technically superior means for expressive interaction in a controlled environment, with many features being translatable to more portable technologies [15, 17].

In particular, Vicon motion capture systems have been used for over a decade for music applications [3, 4, 10, 17]. A software bridge for streaming OSC data from Vicon

[1] See https://www.vicon.com.
[2] Available at https://github.com/RVirmoors/viconOSC.
[3] See https://www.vicon.com/products/software/datastream-sdk.

exists[4] [4] as part of a concluded project, which proved to be incompatible with our current setup.

This decade has seen qualitative advances in the interaction between human gesture and sound behaviour, made possible by real-time gesture recognition and following tools [1, 2, 6]. These allow for more complex scenarios, where movement is used both for direct sonification and for multi-level control of system behaviour—features which our project channels into a coherent framework.

3 CONCEPT

The "sound-thimble", as the basic building block of our framework, is based on the concept of *sound object* in the Schaefferian sense, as a clearly delimited sounding unit, open to manipulation, arrangement and composition [13].

Such an entity, once instanced, can retain an ambiguous nature (spatially and acousmatically) or can switch to a more material state (positioned in space and tied to a causal source) [9]. The duality between the latent positioning of the object (which can be inferred from phenomena other than spatial sound reproduction), and the active sound spatialisation and transformation, becomes an innovative tool for the sonic arts, by enabling sound sketching, auditory games and other real-time interaction setups.

Interaction scenario

We present the initial application of our framework, in the form of an interactive installation comprising three phases: search, manipulation, arrangement.

The narrative begins as an immersive game, with a human player attempting to find a sound-thimble (a stationary virtual object, randomly positioned in 3D space), by analysing cues that are constantly shifting in the sonic fabric based on the hand's movement relative to the object. Analogously to the traditional game of *Hunt the Thimble* (a.k.a *Hot or Cold*), the space between the human and the virtual object is correlated to sound synthesis and modulation parameters. Briefly, the closer one comes to the object, the more coherent the sound and vice-versa.

Once the object is found and attached to the hand, its sonic manifestation gains a richer causal relationship: the player becomes a performer, and is now able to explore the object's sonic palette, and record a number of gestures that can be re-performed later, re-called, and used to trigger or manipulate sonic shifts and events.

Finally, the performer can drop the virtual object to the floor, or discard it by "pushing" it outside of the installation boundaries. This triggers a new object to be randomly generated, while the player retains a degree of control over the initial object via the recorded gestures. Both objects are now in a latent state, with the new one guiding the player's search, and the previous one responding to the learned set of gestures.

[4] See http://sonenvir.at/downloads/qvicon2osc/.

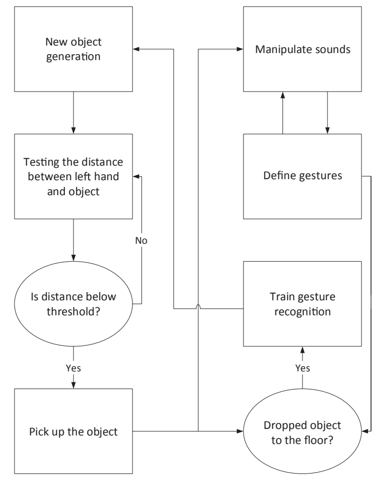

Figure 1: *SoundThimble* interaction workflow.

This repeating scenario is outlined in Figure 1: objects are randomly generated, the performer finds them, defines gestures and interacts sonically, before arranging them in a pleasing configuration. With each spawning of an object or assignment of a gesture, the game becomes more challenging and complex, but also more flexible and rewarding.

Performance aesthetic

The human-object dynamic at the core of our framework results in certain interaction features which circumscribe the aesthetics of any application of the platform.

Our approach is informed by Worrall's study [19], which reveals a necessity for the mapping of minute gestural inflections to alter sonic material with a view to certain modes of listening. The aim of *SoundThimble* is to fluctuate between: reflexive, kinaesthetic, connotative, empathetic, reduced.

The cross-modality between different kinds of sense perception guides the performer's attention to the various sonic responses to physical actions. The multi-modal information is processed in real time, continuously redefining the affordances enabled by the system.

Moreover, since the system's responsiveness is reliant on marker visibility, the performer becomes, to a degree, existentially dependent on the camera eye. This, coupled with the coexistence of virtual, responsive objects in the same scene, can lead to novel mechanics at the limits between presence and absence, real and virtual.

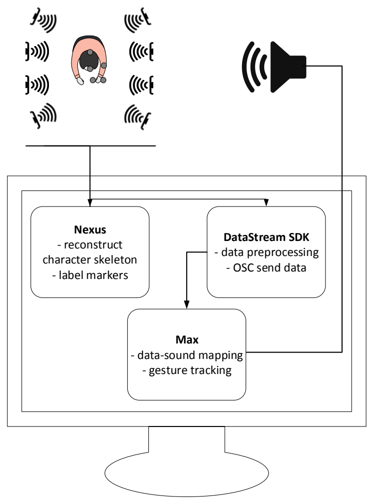

Figure 2: *SoundThimble* framework architecture.

4 IMPLEMENTATION

The framework architecture diagram is laid out in Figure 2. Three-dimensional sensor data is streamed into the Vicon Nexus software, which is able to reconstruct and label the underlying character skeleton. The gesture recognition and sonification algorithms are programmed in Max[5], which receives control data via the OSC[6] protocol. Since Vicon systems do not support OSC out of the box, we used the *oscpack*[7] library to extend the DataStream C++ SDK and send OSC bundles to Max.

The following description is tailored to our installation application, but any *SoundThimble*-based project involves similar conditions.

Character design

Figure 3 shows a skeletal reconstruction in the Nexus environment. We pursued a minimal amount of markers, for ease of setup and prototyping. The resulting configuration—sufficient for tracking hand gestures, while ensuring redundancy in case a marker is obscured from the cameras—consists of 5 markers: two positioned on the head, one on the forearm, and two on the hand (thumb and index finger).

Each OSC bundle sent through the SDK consists of the following data:

- 3D coordinates for the head (averaged from the two head markers);
- 3D coordinates for the hand (averaged from the two hand markers);
- distance between thumb and index finger.

Each of the three items is sent only if non-zero, i.e. for the head and hand coordinates at least one of the respective markers is active, while for the distance computation, both markers need to be visible and correctly labelled. The forearm marker only serves for skeleton reconstruction, and is not sent via OSC.

To maximise responsiveness and minimise data loss, we raised the frame rate to 500Hz, taking into account the maximum movement speed and the minimum spacing between markers [16]. This configuration produces highly stable and responsive inputs into the Max system, with a spatial resolution of 1mm and a latency of around 5ms.

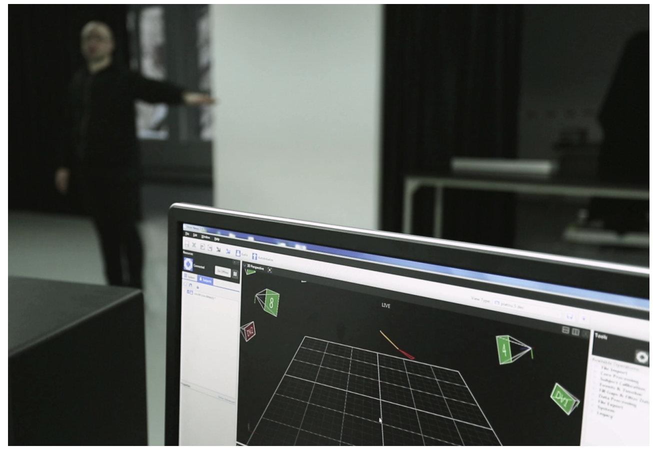

Figure 3: A performer being tracked in Vicon Nexus. Two visible segments: head-forearm, forearm-hand.

Object generation & interaction mechanics

The following object-related mechanics are implemented as basic algorithms in Max: generation, detection, release.

Objects are generated at random positions within the boundaries of the motion capture field. Detection of the sound-thimble occurs when the distance between hand and object falls below a set threshold. By default this threshold is set at 200mm radial distance; lowering it can make the game considerably more difficult. Once the object is detected, it becomes mobile, its coordinates tracking those of the hand’s.

Finally, a simple thresholding of the z-axis (height) value of the hand position serves to release the object. If the velocity computed on the x or y axes (horizontal plane) is high enough, then the object is pushed in the respective direction, and is able to leave the area of the installation, essentially being removed from the game. At this point, a new object is

[5]Max is a state-of-the-art programming environment for real-time multimedia: http://cycling74.com/.
[6]OpenSoundControl is a multimedia communication protocol: http://opensoundcontrol.org/.
[7]See http://www.rossbencina.com/code/oscpack.

generated. The system keeps track of all object coordinates, as they appear and disappear over time.

Gesture recognition

We use the thumb-index finger distance value to enable gesture recording while the two fingers are kept close together. The input features captured into *MuBu* multi-buffer containers [14] consist of cylindrical triplets:

- $\Delta\theta/\Delta t$, with $\theta = \tan^{-1}(\frac{y}{x})$;
- $r = \sqrt{x^2 + y^2}$;
- z,

where x, y, z are the respective differences between head and hand Cartesian coordinates, as received via OSC. This feature preprocessing serves two purposes:

Firstly, gestures are recorded based on the position of the hand relative to the head, thus becoming invariable to the performer's absolute *position* within the space. Secondly, by considering the variation of angle θ over time (as opposed to its absolute value), gestures become invariable to the performer's *orientation* on the horizontal plane. Thus, gestures can be recorded and recalled anywhere within the space, irrespective of the direction the performer is facing.

The input features are fed to the *Gesture Follower* `gf` Max object, part of the *MuBu* package. The algorithm, based on a Sequential Monte Carlo inference engine [2], allows for *gesture spotting*, i.e. it constantly produces likelihood values of a certain gesture being active, together with an approximation of its completion rate. If these exceed a certain threshold, the respective gesture is triggered. Once detected, the gesture can be followed at a variable rate or scale, even backwards.

In the game, each generated object has a number of gestures associated to it. When an object is released to the floor, the classifier is (re)trained with the new data, and consequently gestures can act as on/off switches for a particular sonic behaviour (if they are performed/spotted once at a time), or as continuous controllers (if they are repeated). When several objects exist on the floor, a specific movement might act on one or more objects, depending on which detection likelihoods exceed the threshold.

Sound design

Each sound-thimble has a corresponding sound design patch, differing in (a) the source sound material used, (b) the synthesis techniques applied, and/or (c) the control mapping schema to the object search and manipulation variables. The various combinations of (a), (b) and (c) give rise to a growing library of objects, each with its own character. By designing various interaction rules for each object, segments are linked to different synthesis parameters or groups of parameters resulting in a continuously evolving, organic soundscape.

We differentiate between the three phases of the installation in terms of mapping technique and level of sonic interactivity. The search mode employs straightforward parameter mapping, where human-object distance measures are linked to synthesis parameters, while the manipulation phase relies on a model-based mapping approach where different gestures and actions reach deeper levels of control [7]. Finally, variations on both these techniques occur in the arrangement phase.

The synthesis patches used for search are built around a process of decorrelation [11]: the farther the human's hand from the object the more decorrelation occurs, up to the point where each instance of the signal becomes a distinct sonic entity. This is done by continuously modulating each of the copies in terms of pitch (FM) and amplitude (AM) with low frequency oscillators (LFO's). Changes in frequency, amplitude and wave shape of the LFO's lead to complex sonorities ranging from coupled streams of sound to distinct iterations with a high degree of randomness. This mechanism is subtly mixed with a granular engine in a latent state, which gains more prominence in the next mode.

In the manipulation phase, the 3D space is split into chunks, each one acting as a zone with its own mappings and interaction laws. The synthesis patches are based on the segmentation of a source sound into short grains. We built a granular synth with these controllable parameters: grain size, grain position, envelope shape, level of scattering, pitch, timbre and stereo width. Another patch implements a concatenative synth that traverses the grains guided by movement velocity and trajectory. Certain gestures trigger sonic events while interpolating between sets of parameter values via a convergent-mapping schema [8].

In the search phase, all sound design is based on two channels that can either be routed to multiple pairs of speakers or downmixed to mono and diffused on an arbitrary number of speakers. In the manipulation phase, the soundscape is spatialised to track the position of the performer. In the arrangement phase, the dropped object retains a "root" source location, which can relate dynamically to the performer position when a gesture is executed: for instance, the granular patch sends spatialised grains back and forth between the dropped object and the performer.

In developing the sound design algorithms, we used two input data sources. The first one is motion capture data recorded in Nexus and played back through the SDK. For more flexibility and immediate control, we also developed a basic visual interface to monitor the input data and to manipulate it in virtual 3D space using the mouse, for instant auditory feedback. This feature comes in useful when specific motion data is not available and needs to be roughly simulated.

5 CONCLUSIONS AND FUTURE WORK

This paper introduced *SoundThimble*, a multi-layered platform for real-time motion-music interaction. All software developed for the project (including the C++ code for data preprocessing and transmission, and the Max patches for gesture tracking and sound design) is open source and publicly available.

The team is currently pursuing several directions for improvement and extension. We are working to support more than one participant at once, implementing mechanics for sharing control of the virtual object. Meanwhile, work continues on enriching the interaction and sound design.

Finally, we are commencing the outreach to composers, artists and creative programmers, to apply our platform to new innovative projects and engage in practice-led research.

REFERENCES

[1] Baptiste Caramiaux, Jules Françoise, Norbert Schnell, and Frédéric Bevilacqua. 2014. Mapping through listening. *Computer Music Journal* 38, 3 (2014), 34–48.

[2] Baptiste Caramiaux, Nicola Montecchio, Atau Tanaka, and Frédéric Bevilacqua. 2015. Adaptive gesture recognition with variation estimation for interactive systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 4, 4 (2015), 18.

[3] Christopher Dobrian and Frédéric Bevilacqua. 2003. Gestural control of music: using the vicon 8 motion capture system. In *Proceedings of the international conference on New interfaces for musical expression*. National University of Singapore, 161–163.

[4] Gerhard Eckel, David Pirro, and Gerriet K Sharma. 2009. Motion-enabled live electronics. In *Proceedings of the 6th Sound and Music Computing Conference, Porto, Portugal*.

[5] Shai Fine, Yoram Singer, and Naftali Tishby. 1998. The Hierarchical Hidden Markov Model: Analysis and Aplications. *Machine Learning* (1998), 41–62.

[6] Jules Françoise, Norbert Schnell, Riccardo Borghesi, and Frédéric Bevilacqua. 2014. Probabilistic models for designing motion and sound relationships. In *Proceedings of the 2014 international conference on new interfaces for musical expression*. 287–292.

[7] Thomas Hermann, Andy Hunt, and John G Neuhoff. 2011. *The sonification handbook*. Logos Verlag Berlin.

[8] Andy Hunt and Ross Kirk. 2000. Mapping strategies for musical performance. *Trends in Gestural Control of Music* 21 (2000), 231–258.

[9] Brian Kane. *Sound Unseen - Acousmatic Sound in Theory and Practice*. Oxford University Press, New York.

[10] Ajay Kapur, George Tzanetakis, Naznin Virji-Babul, Ge Wang, and Perry R Cook. 2005. A framework for sonification of vicon motion capture data. In *Conference on Digital Audio Effects*. 47–52.

[11] Gary S Kendall. 1995. The decorrelation of audio signals and its impact on spatial imagery. *Computer Music Journal* 19, 4 (1995), 71–87.

[12] Kristian Nymoen, Ståle Andreas van Dorp Skogstad, and Alexander Refsum Jensenius. 2011. Soundsaber-a motion capture instrument. In *Proceedings of the international conference on New interfaces for musical expression*.

[13] Pierre Schaeffer, Guy Reibel, Beatriz Ferreyra, Henri Chiarucci, François Bayle, Agnès Tanguy, Jean-Louis Ducarme, Jean-François Pontefract, and Jean Schwarz. 1998. *Solfège de l'objet sonore*. INA GRM.

[14] Norbert Schnell, Axel Röbel, Diemo Schwarz, Geoffroy Peeters, Riccardo Borghesi, and others. 2009. MuBu and friends–assembling tools for content based real-time interactive audio processing in Max/MSP. In *International Computer Music Conference*.

[15] Ståle Andreas van Dorp Skogstad, Alexander Refsum Jensenius, and Kristian Nymoen. 2010. Using IR optical marker based motion capture for exploring musical interaction. (2010).

[16] Min-Ho Song and Rolf Inge Godøy. 2016. How Fast Is Your Body Motion? Determining a Sufficient Frame Rate for an Optical Motion Tracking System Using Passive Markers. *PloS one* 11, 3 (2016), e0150993.

[17] Gabriel Vigliensoni and Marcelo M Wanderley. 2012. A Quantitative Comparison of Position Trackers for the Development of a Touch-less Musical Interface.. In *NIME*.

[18] Greg Welch and Eric Foxlin. 2002. Motion tracking: No silver bullet, but a respectable arsenal. *IEEE Computer graphics and Applications* 22, 6 (2002), 24–38.

[19] David Worrall. 2013. Understanding the Need for Micro-Gestural Inflections in Parameter-Mapping Sonification. In *International Conference on Auditory Display*.