SoundThimble: A High Resolution Gesture Sonification Framework

Ben Trovato
Authors hidden for review
1932 Wallamaloo Lane
Wallamaloo, New Zealand
trovato@corporation.com

G.K.M. Tobin
Authors hidden for review
P.O. Box 1212
Dublin, Ohio 43017-6221
webmaster@marysville-ohio.com

ABSTRACT

The installation is based on a state-of-the-art Vicon motion capture system, used alongside a Max-based platform to track, interpret and sonify the movement and gestures of a performer in 3D space.

Author Keywords

sonification, motion tracking, gesture spotting, interactive installation, synthesis

ACM Classification

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

1. INTRODUCTION

High resolution three-dimensional motion capture systems are traditionally used for animation in film and games [?], as well as for life sciences research [?] and engineering applications [?]. This technology has long been mined by the NIME community [2, 8], although in many early cases, technological limitations meant that the motion data transmission and the sound generation processes were not simultaneous [2, 6].

The *SoundThimble* project harnesses current motion tracking technology and gesture detection algorithms to develop new modes of sound exploration in an interactive installation context. Our aim is to push beyond the standard paradigms of isolated body motion audification [2, 6] or parameter mapping-based new instruments [8], towards deeper narrative structures coupled with layered arrangement of music patterns.

Our implementation uses a state-of-the-art Vicon motion capture system[1] based on 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 existing projects and technology (section 2), we describe the *SoundThimble* concept and development (section 3), we analyse two practical case studies (section 4), and we conclude with a survey of remaining challenges and future perspectives (section 5).

2. RELATED WORK

- interactive / movement sonification examples[3].
- Vicon & related projects
10+ year history of Vicon+sonification
- micro
[11]
- vicon + OSC de la iem.at

The current decade has seen qualitative advances in the interaction between human gesture and sound behaviour [4]. ... [5].

3. PROJECT DESCRIPTION

3.1 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 [9].

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) [1]. 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, once tied to motion data, becomes an innovative tool for sonic arts via sound sketching, auditory games and other realtime interaction scenarios.

3.1.1 Interaction scenario

SoundThimble can be viewed as both an interactive sound installation and an auditory game.

- 3 etape: search, manipulation, arrangement.

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

Once the object is found, the performer ... has full freedom to record a number of gestures that can be re-performed later, re-called, and used to trigger or manipulate various

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

NIME'17, May 15-19, 2017, Aalborg University Copenhagen, Denmark.
.

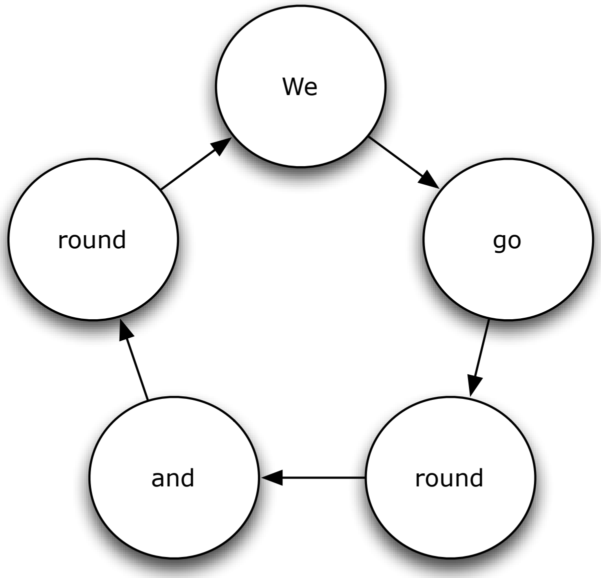

Figure 1: ... diagram

sound events/sonic shifts.

We can think of SoundThimble as having THREE different narratives: one before the object is found, where only distance in relation to the object is used to control sound and the second where gestures are being employed, after the object has been found. Generally, the level of interaction is higher after the object is found, however both situations can overlap (e.g: the performer tries to find the second object while recalling gestures that were assigned to the first object). This repeating scenario is outlined in Figure 1: objects are randomly generated, the performer finds them, defines gestures and interacts sonically with them before arranging them in a pleasing configuration.

3.1.2 Performance aesthetic

- text Bogdan

3.2 Implementation

=== FIG: Framework Diagram: Camere, Nexus, C++, Max, Speakers ====

One of the first challenges was to create a method of sending real-time from Vicon to MaxMSP where data would be processed and used to generate and manipulate sound. By default, the Vicon system does not support the Open Sound Control (OSC) protocol (ref), needed to communicate with MaxMSP. To overcome this limitation, some code modifications and additions have been implemented inside Vicon's Blade sdk.

3.2.1 Character design

Figure **??** shows the Vicon Nexus software

=== FIGURE : Nexus / Bogdan ====

Implementation of the concept presented above requires Nexus, Vicon SDK and MAX sotware. Every character involved in the scene is defined by a limited number of markers. In this case, two markers are positioned on the head, one marker is positioned on elbow and the others 2 on the hand (thumb and index finger). Every marker has associated a name in Nexus and between them 6 segments are drawn. It is very important in realtime capture motion, that the marker to have assigned correct coordinates.

- **Vicon extensions (SDK plugin)**

Vicon's SDK is a versatile and simple tool for users to gain easy access to Vicon DataFlow created in Nexus, Blade or Tracker applications.The Vicon DataStream Software Development Kit (SDK) provides intuitive programable access to data with custom functions created in C++. With the help of some functions, Vicon's SDK forwardes the Vicon DataStream to other constructive softwares and plug-ins to create custom applications [4]. In combination with Open Sound Control protocol, Vicon's SDK forwards data to any software compatible with this communication protocol (eg. MAX). OSC is a protocol for communication among computers, sound synthesizers and other multimedia devices[5]. Hence, any marker can be routed in MAX using its parameters and coordinates. Also, the Vicon DataStream Software Development Kit (SDK) admits inside changes such as labeling markers, timecode generation and framerate.

3.2.2 Object generation & interaction mechanics

Manipulating objects algorithm consists of some big steps: object generation, finding the object, picking up the object, throwing the object on the floor.

Object generation is realized by random generators with the help of *drunk* object, but with certain limits. These limitations are influenced by the dimensions of the room in which the Vicon system is installed. Finding the object supposes continuous mathematic operations between the coordinates of the object and coordinates of the left hand's marker. This process comes with an audio feedback. When these coordinates are close enough one to another, the object is retreived and manipulated by performer (eg. define gesture). After all these processes, a simple comparison between the coordinates of the floor and the value of the z axes of the marker is done in order to put down the object. According to this, a performer can handle as many objects as he wants.

3.2.3 Gesture recognition

Mubu containers provided by Ircam laboratories in MAX software represent a handy tool to record and analyze gesture, captured with Vicon system [7]. Our gesture recognition algorithm is based on Hierarchical Hidden Markov Models (HHMM) implemented in *mubu.hhmm* object of MAX/MSP. HHMMs are a generalization of HMM where each state is considered to be a self-contained probabilistic model [10]. The system is trained by captured data which is essentially a gesture. This process requires a predefined indicator in order to delimitate gestures from all data flow. The algorithm analyzes all input data and generates a probability of similarity between data and saved gestures. In order to control every generated object, there are associated 2 or 3 gestures saved by the performer, but there is a limited time for the gestures to be executed. Predefined gestures offer the possibility to delete the gesture just saved and also indicate the moment the gesture is recorded.

3.2.4 Sound design

Although there is a wide range of sonification alternatives available, including triggering of pre-stored sounds, modulating/LFOing stored sounds etc. we started experimenting with a few synthesis patches with parameters that could be correlated to XYZ coordinates in a reactive and expressive manner. In order to exploit the increased spatial resolution, marker positions, grouping of markers and other possibilities of the Vicon system, the development of synthesis algorithms in

[4]See `https://www.vicon.com/products/software/datastream-sdk/`.

[5]See `http://opensoundcontrol.org`.

MaxMSP seems to be very flexible. In this way, the whole soundscape can be generated in a continuous, organic manner by correlating markers' positions with synthesis parameters.

The interactive experience can be described as having two main paradigms: object finding and object interaction. For object finding we have been experimenting with two straightforward patches in MaxMSP: the first is based on a sawtooth wave that is sent to 8 delays. These output a random signal with a settable range which continuously alter the phase and frequency of the sawtooth iterations. This chorus-inspired algorithm has three controllable parameters: main frequency, range and speed of frequency variation. The farthest someone is from the virtual object the more detuning and phase shifting occurs on each of the eight iterations, while approaching it the effect becomes less pronounced to the point where only slight variations of the signal occur. Also, main frequency is associated with movement in the vertical plane. Further modulation can be used to make the sound more complex. The second patch is somewhat based on the same principle of decorrelation: six sine waves with different frequencies are modulated in amplitude by a random object which generates control waves with a variable degree of complexity (more or less noise-like shapes). As in this case, distance between the performer and the virtual object is associated with the level of randomness and decorrelation: longer distance translates to a higher degree of decorrelation and randomness. What is interesting about this patch is the effect of amplitude modulation (AM) (referinta ce e AM) when the carrier frequency (?) goes beyond 20hz and sidebands occur. By using noise-like carriers, complex sonorities occur with a variable harmonic content. In both cases, the performer tries to find the object by listening to these variations. By correlating small and large variations to its position in the 3d field the performer receives meaningful clues about where the object might be as well as an interesting and engaging soundscape.

An additional granular synthesis (ref) patch that was initially created for object interaction also seems to be effective for object finding. The patch reads short grains from a pre-loaded sound and scatters into "clouds" in either a controlled or random manner. Among the parameters that can be mapped are: grain position, grain size, envelope shape, level of scattering, pitch, stereo width. Not only this, but by using the [pattrstorage] and [pattr] objects we can group multiple parameters' state, save them as presets and interpolate between them. By doing this, we can control an undefined number of parameters by linking only one marker/one gesture to the interpolation amount box. This patch also seems to be effective for finding the object, differentiating the two paradigms by the level of control: less control for object finding and more control for object interaction.

Real-scenario testing and simulations using the Vicon cameras is quite limited because two main reasons: we don't always have access to the space and we would always need a performer that could cope with long hours of code debugging, errors, MaxMSP programming and so on. This is the reason why in order to simulate interactions inside the sound design patches, we also created a basic interface in Jitter that receives data from Vicon Blade via OSC and represents each marker in the 3d space. By using this interface it is possible to move a particular marker anywhere along the XYZ planes, by only using the mouse, and get instant auditory feedback. The downside being that experiments done in virtual space do not always translate well to the real space: calibration by value scaling, experimenting with different function shapes etc. is tedious but absolutely necessary.

So far, 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, however multichannel sound is taken into consideration for future improvements.

- Visualisation (jitter)

4. CASE STUDIES

4.1 Interactive Installation

- performance analysis

4.2 Performance

5. CONCLUSIONS AND FUTURE WORK

- Areas of improvement
  - Eye tracking?

Future work on SoundThimble could/will include: multiple performers, eye tracking, generative visuals, spatial sound, more powerful synthesis and sound manipulation algorithms, different rules added to the narrative of the auditory game.

6. ACKNOWLEDGMENTS

Hidden for review.

7. REFERENCES

[1] BrianKane. *Sound Unseen - Acousmatic Sound in Theory and Practice*. Oxford University Press, New York.

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

[3] T. Hermann, A. Hunt, and J. G. Neuhoff. *The sonification handbook*. Logos Verlag Berlin, 2011.

[4] K. N. Jorge Solis. *Musical Robots and Interactive Multimodal Systems*. Springer-Verlag Berlin Heidelberg, Berlin, 2011.

[5] R. B. F. B. Jules Francoise, Norbert Schnell. Probabilistic models for designing motion and sound relationships. *International Conference on New Interfaces for Musical Expression*, pages 287–292, June 2014.

[6] A. Kapur, G. Tzanetakis, N. Virji-Babul, G. Wang, and P. R. Cook. A framework for sonification of vicon motion capture data. In *Conference on Digital Audio Effects*, pages 47–52, 2005.

[7] D. S. G. P. R. B. Norbert Schnell, Axel Robel. Mubu and friends assembling tools. *International Computer Music Association*, pages 423–426, August 2009.

[8] K. Nymoen, S. A. v. D. Skogstad, and A. R. Jensenius. Soundsaber-a motion capture instrument. 2011.

[9] P. Schaeffer, G. Reibel, B. Ferreyra, H. Chiarucci, F. Bayle, A. Tanguy, J.-L. Ducarme, J.-F. Pontefract, and J. Schwarz. *Solfège de l'objet sonore*. INA GRM, 1998.

[10] N. T. Shai Fine, Yoram Singer. The hierarchical hidden markov model: Analysis and aplications. *Machine Learning*.

[11] D. Worrall. Understanding the need for micro-gestural inflections in parameter-mapping sonification. Georgia Institute of Technology, 2013.

TO DO: review bib!