Physics-based Concatenative Sound Synthesis of Photogrammetric models for Aural and Haptic Feedback in Virtual Environments

Magalhães E.* DEI/UPorto
Jacob J.† LIACC/UPorto
Nilsson N.‡ Aalborg University
Nordahl R.§ Aalborg University
Bernardes G.¶ DEI/UPorto

ABSTRACT

We present a novel physics-based concatenative sound synthesis (CSS) methodology for congruent interactions across physical, graphical, aural and haptic modalities in Virtual Environments. Navigation in aural and haptic corpora of annotated audio units is driven by user interactions with highly realistic photogrammetric based models in a game engine, where automated and interactive positional, physics and graphics data are supported. From a technical perspective, the current contribution expands existing CSS frameworks in avoiding mapping or mining the annotation data to real-time performance attributes, while guaranteeing degrees of novelty and variation for the same gesture.

Index Terms: Sound Synthesis—Interaction—Virtual Reality—Sonic Interaction Design

1 INTRODUCTION

Contact between interacting objects in a natural environment often convey information about the objects themselves. For instance, if walking on a surface, the contact between our feet and the ground will provide auditory and haptic feedback, which in many cases may be sufficient for us to recognize the surface we are walking [10]. As such, surface interaction feedback is not only expected from our physical experience in natural environments, but also important whenever we need it to identify surface attributes, such as its thickness or materials. Existing methods for multimodal interaction have been assessed as expressive and natural, yet, to the best of our knowledge, no systematic evaluation has addressed the degrees of multimodal correspondence at finer temporal granularities [4, 6, 9].

In this context, our work strives for a method capable of generating audio streams with highly controllable nuances for aural and haptic feedback using concatenative sound synthesis (CSS) [8] following our previous work [5]. Despite the lack of use cases adopting this sample-based synthesis in Virtual Reality Environments (VREs), the technique has proven to be quite robust in generating dynamic, evolving and ever changing sound textures from short audio sources. Additionally, it enables creation of audio streams, at different temporal granularities. CSS corpora consists of a short collection (i.e., about 60 seconds) of synchronized 21.3 ms audio units and interaction data from a dictionary of performed gestures in the virtual modelled assets of multiple real-world surfaces. Signal annotations include data from physical interactions (e.g., impulses, forces, velocities) and material properties (e.g., diffuse and bump maps) of collision points between objects in a virtual environment. During performance, synthesis of aural and haptic feedback is automatically rendered in real-time by concatenating units form the aural and haptic corpora. Congruence across performed gestures, such as scraping, hitting and rolling, and their aural and haptic feedback is guaranteed by navigating and retrieving (aural and haptic) units from their interaction data. From a technical perspective, the current contribution expands existing CSS frameworks by avoiding mapping or mining the annotation data to real-time performance attributes, while guaranteeing degrees of novelty and variation for the same gesture. Furthermore, the integration of CSS in VREs can tackle two important limitations of the technique, as identified in [8]: 1) the evaluation of a descriptors' salience, notably the difference between the aural and the haptic descriptor spaces, and 2) the definition of targets which convey both the finer degree of user controllable actions interacting with the VREs.

Ultimately, we aim to create a unified aural and haptic framework, which stresses a high level of control, adaptation and possible a greater sense of agency and presence engendered by congruent sensory feedback. In this paper, we present a framework for expressive sonification of highly realistic graphic models, applicable for continuous sonic interaction type experiences with a high level of micro variance and expressiveness.

The paper presents a preliminary use-case scenario using an instrumented board game, where a metal sphere is used for interacting with the surface textures and geometry of a wood board structure. Using image texture features and their intrinsic structure, we aim to foster a method to dynamically map user actions-perception loop to a CSS engine. In this context, the main novelty is the use of CSS for aural and vibrotactile feedback, which to the best of knowledge hasn't been thoroughly explored and tested. Ultimately, the contributions aim to explore a larger framework that addresses spatial temporal cycle actions with a high level of visual-aural-haptic synchronism. We advance some preliminary results, which propose that sound descriptor-based models within CSS, can provide a real-time adaptive method to induce vibrotactile feedback within VREs.

2 PHYSICS-BASED CSS FRAMEWORK ARCHITECTURE

Figure 1 shows the architecture of the proposed physics-based CSS framework, and its two major components. It targets the synthesis of auditory and haptic feedback by concatenating multi-surface *corpus* of annotated audio *units*. Contrary to the traditional CSS architecture [2], the multi-dimensional *feature space* is driven from real-time collision data in VREs, instead of audio signal attributes. In the multi-dimensional feature space, audio units stemming from the same physical conditions during the capture stage result in neighbourhood locations. In other words, we ascertain that similar collision points data in the VREs have a aural and haptic perceptual resemblance, thus a smaller distance in the feature space. Furthermore, *audio source* signal and annotations are captured in the same conditions as the rendering stage. Thereby, audio units' annotations and online *target feature vector* navigation occurs in the same feature space.

3 PHYSICAL INSTRUMENTED INTERACTION

We created a simple prototype to test the proposed architecture. This prototype consisted of a wooden, flat board with wooden railings and a metal sphere that is to be moved inside the board by tilting the board itself (see Figure 2). This prototype was designed with the

*e-mail: eduardom@fe.up.pt
†e-mail: joao.jacob@fe.up.pt
‡e-mail: ncn@create.aau.dk
§e-mail: rn@create.aau.dk
¶e-mail: gba@fe.up.pt

978-1-7281-6532-5/20/$31.00 ©2020 IEEE
DOI 10.1109/VRW50115.2020.0-195

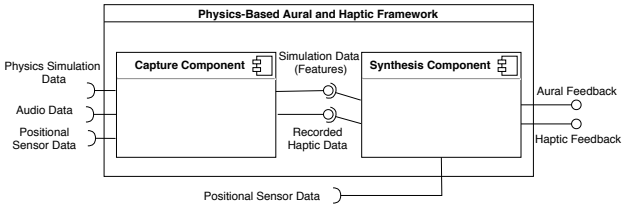

Figure 1: Overview of the two major components of the proposed framework

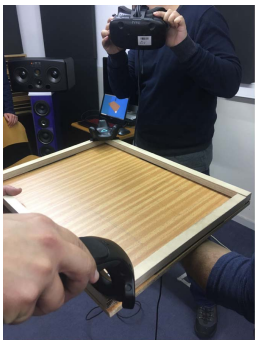

Figure 2: Instrumented board calibration

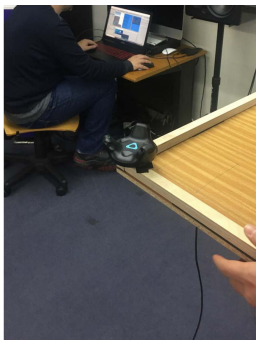

Figure 3: Positional tracker placement

goal of allowing for a user, holding and tilting the board which was coupled with an inducer speaker, while equiped with a VR headset and headphones, to experience the same visual, aural and haptic perceptions as if caused by the real ball interacting with the board.

In order to guarantee an alignment between the virtual representation of the board in the virtual environment (a captured photogrammetric model) and the board itself in the real world, it was necessary to place and correctly align two positional trackers: one fixed (an HTC Vive tracker, depicted in Figure 3) and a temporary one (a HTC Vive controller). The tracker was placed, using velcro, on one corner of the board while the controller was placed on the opposing corner. As both objects have their positions and orientations tracked, it was then necessary to align the 3D representation of the board with the 3D representation of both the controller and tracker in the virtual environment, ensuring that the virtual board would have the tracker as its pivot. After this process, the wand controller is removed, as the tracker itself is now sufficient for tracking the movements of the board. Whenever the tracker was removed from the board, this calibration process was repeated.

4 Implementation

Towards the goal of congruency and expressiveness within VREs through physical interaction, we process the physics simulation of the VER and compute the relevant features required by the CSS Engine (see Figure 7).

4.1 Physics Simulation

As Unity already features a physics engine, based on NVidia's Physx [3], we have opted to extract data that is already exposed by the API. As seen in Figure 4, we have created a specific entity, called *CollisionPoint*, that enriches the information from a *ContactPoint*, the result of a *Collision* between two colliding *GameObjects*. The data that we have extracted consists of:

- **Position**, the 3D coordinates of the collision point;

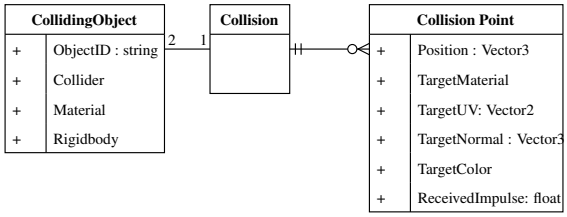

Figure 4: Entity diagram representing the entities and properties involved in the collision data extraction

- **TargetMaterial**, the material ID of the Target;
- **TargetUV**, the UV mapping coordinate associated with the collision point inside the colliding meshes' triangle, extracted through the barycentric coordinates formula (see Equation 1);
- **TargetNormal**, the normal vector of the collision point inside the colliding mesh's triangle, a combination of the 3 vertices' normals and the extracted normal perturbation from the normal map of the TargetMaterial;
- **TargetColor**, the diffuse color extracted from the Targetmaterial's diffuse texture map;
- **ReceivedImpulse**, the impulse (in $N.s$) that the Source receives.

We consider that a collision must necessarily be comprised of two colliding entities, a Source and a Target (as presented in Figure 4, for the purpose of our approach. This, however, still allows for a collision between one source object and multiple target objects to be expressed as a mapping to a n-tuple (see Equation 2). It also considers that a collision between two objects will only be considered once, having only one object as a constant reference (i.e.: If object A, collided with object B, only the collision $A \rightarrow B$ is considered, and not $B \rightarrow A$).

The barycentric coordinates of a point are calculated based on an equation system 1. The values of A, B, C are the 3d coordinates of the three vertices that comprise the relevant triangle of the collided mesh. P represents the 3d coordinates of a collision point, within that triangle. The values of, u, v and w are weights that allow for the value of P to be expressed from the values of the three vertices of the triangle. With the values of u,v and w it is possible to calculate other mesh-related data, relative to point P, based on the same type of data that is known to be associated with A, B and C. As such, it is possible to calculate the UV texture mapping coordinates for point P, allowing for the easy extraction of information encoded in the material's texture maps (such as Diffuse Color, Displacement and Normal maps).

This extracted data serves as the basis for computing the feature vector that will be used by the Synthesis component: velocity, material id, angular velocity, impulse, normal and color.

$$\begin{cases} P = uA + vB + wC \\ 1 = u + v + w \\ 0 \leq u,v,w \leq 1 \end{cases} \quad (1)$$

$$Collisions : A \rightarrow ((Col(A,B)), \ldots, (Col(A,N))) \quad (2)$$

4.2 Communication flux between Components

To guarantee that the Audio Data, Positional Sensor Data, and Physics Simulation Data are all synchronized, an instruction was created in Unity3D to ensure that the source audio recordings start

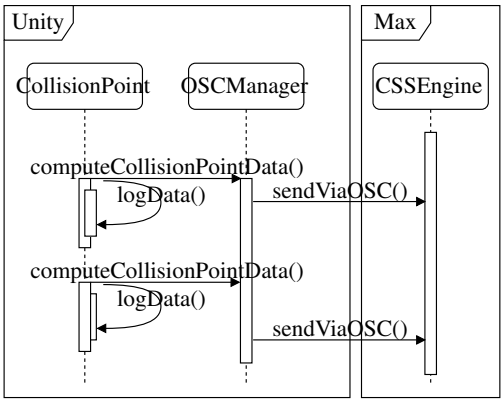

Figure 5: Sequence Diagram representing Collision and OSC Events between Unity and Max

Table 1: Employed OSC addresses and their purpose

Address	Relevant Component	Tipology
t/**record**	Capture	UI
t/**pause**	Capture	UI
t/**stop**	Capture	UI
collisionPoint/**velocity**	Synthesis	Event
collisionPoint/**materialid**	Synthesis	Event
collisionPoint/**angularvelocity**	Synthesis	Event
collisionPoint/**impulse**	Synthesis	Event
collisionPoint/**normal**	Synthesis	Event
collisionPoint/**color**	Synthesis	Event

only when the physics engine is ready and in sync, producing a continuous log of all the user´s gestures and spatial positions that transform continuously the acoustic properties of the metal sphere in interaction with the board´s materials that serve as the Simulation Data features that will be used by the Synthesis Component.

General user-input parameters, which control the framework components are in the Unity3D as a graphical user interface. It includes commands to start, pause and stop a recording, ensuring that data from different sources is collected, concomitantly. This communication is supported through the OSC message format. Several addresses were designed, to allow for the communication needs of each module, as per Table 1. We have opted to employ one address per identified feature instead of a single one for the full vector, to allow for the addition of future addresses as other features are considered relevant. As depicted in Figure 5, OSC messages are composed and sent independently from the computed collision point data events, and are sent at a fixed interval, even if no collision occurred (in which case, a completely blank message is sent). This fixed interval (given by the equation $T = W/f$) is based on the requirements of the CSS engine, responsible for resynthesizing samples, which is dependant on the window size and sample rate. Values of 1024 and 48000 were used for W and f, respectively, leading to a period of approximately 21 ms. However, as the rendering of the VRE, physics simulation and audio resynthesis occur at different intervals, there is the possibility of a delay between a subject observing two objects interacting, that interaction being simulated and aural and haptic feedback being provided back to the subject (Table 2), which may compromise congruence and presence, even without taking into account equipment response times.

4.3 Audio Corpus Creation and Annotation

For a given surface `ID`, we create two aural, a, and haptic, h corpora. Each corpus include a collection of segmented audio units i ($1 \leq i \leq n$, where n are the total number of audio units per corpus) each annotated by a thumbnail feature vector including its collision point data in the VREs. We adopt an arbitrary unit segmentation on the captured audio sources of $S \approx 21.3$ ms duration. The larger the corpus can guarantee greater quality in the resulting synthesis, as it promotes smoother transitions across concatenated audio units and a larger degree of variation for similar target input vector. In our case study, audio sources have an average duration of 60 seconds, which results in corpora with 2500 audio units.

Table 2: Theoretical minimum update intervals of modules

Component	Update interval (milliseconds)
Rendering	11.1
Physics	33.3
OSC Manager	21.3
Log Manager	21.3

Figure 6: Audio and Data Capture

The different nature of aural and haptic audio sources strive from different capture techniques using a piezzo electric microphone coupled to the structure of the board game for specific materials: plain wood, plains steel, irregular steel. For a richer database, we recorded each session using two microphones placed about 17 cm apart (AB stereo technique) and a shotgun microphone. The signals are summed into single source signal. Images of the acquisition process are shown in Figure 6.

For this particular prototype and case scenario we recorded mostly gestures that encompass different velocities and acceleration applied to the metal spheres through different board slopes which generate identifiable and continuous acoustic variations. We also recorded impacts with different intensities, micro positional contact variations and silence to test the system response and the unity selection semantic accuracy.

At runtime, the synthesis is performed using the MuBu library [7] within MaxMSP real-time programming language. Two target audio units are retrieved and concatenated from the two aural and haptic corpora from a given surface ID. A surface ID number value is outputted from the Unity3d game engine to the MaxMSP CSS engine, which will route and select the adequate aural and haptic sonic corpora with its corresponding pre-annotated data. This will ensure the mapping between target feature vector, $T(k)$, and audio unit selection, $U_i(k)$, thus promoting sensory congruence between visual display, aural and haptic feedback. At every target vector query, the selection of the audio unit to be synthesized is computed by finding the unit $1 \leq i \leq n$, which minimizes the target cost λ computed as the Euclidean distance between a target feature vector $T(k)$ and the feature vectors $U_i(k)$ from the entire collection of all corpus units , such that:

$$\lambda = \underset{i}{\operatorname{argmin}} \sqrt{\sum_{k=1}^{5} (|T(k) - U_i(k)|)^2} \quad . \tag{3}$$

A temporal sequence of target-to-corpus feature vector mappings $T \rightarrow U_i$, where i is computed by the target cost λ in Eq. 3 is then defined every 21.3 ms, i.e., the corpus unit duration, S.

To minimize undesired sound stuttering effects resulting from the temporal repetition of selected units, we introduce a penalty in

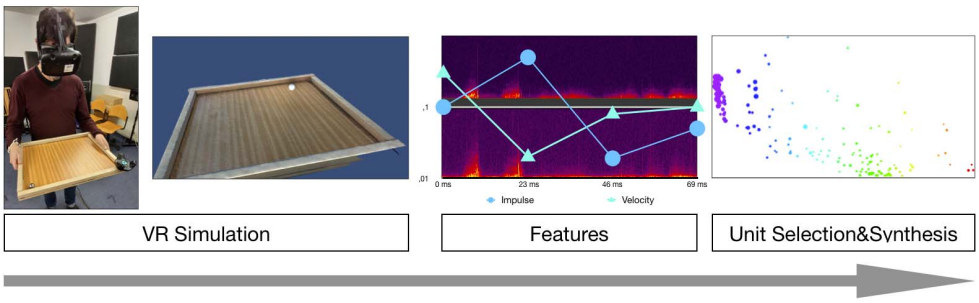

Figure 7: Simulation Pipeline

the target cost computation λ. Precisely, we follow the approach detailed in [1], which increases the target cost λ of the most recently selected units by a factor of $1 \leq \eta \leq 4$. Penalties are gradually reset to the original target cost in eight segment selections by imposing a difference of $(\eta - 1)/8$ at each target unit query.

Two audio streams for aural and haptic feedback are then synthesized by concatenating the audio signals of the selected units from the a_i and h_i corpura. Resynthesis of corpus' units uses a phase vocoder overlap-add processing, which continually reads from the audio sources. At every new selected unit, the vocoder reading pointer in the audio sources are updated to the corresponding $i \cdot S$ ms location. Selected units are then played back through its correct output channels: headphones for the aural feedback and a low frequency audio transducer for the haptic feedback.

5 Conclusions and Future Work

In this paper, we presented a work in progress towards a framework for real time continuous interactions for haptic and aural feedback in VREs. We detailed a framework for robust and efficient mapping decomposition and resynthesis of aural and haptic feedback from sound units based on the real-time capture and synchronization of real-world objects interaction. Our approach ultimately aims at automatic gestural mapping and expressive control of contact sounds derived from the interaction with photo-realistic (photogrammetric) assets and congruent sensory identification. The interaction control from the graphic engine feeds a novel framework for CSS, which is then responsible for the navigation in a sound corpus towards dynamic expressive simulation for aural and haptic feedback.

The major contribution of the framework relies in the potential to increase the feeling of realism and immersion through a sense of agency in interactive VREs, notably by enforcing perceptual cues through the capture and rendering to depart from realistic sources, combined with high quality graphics and dynamics simulation. Furthermore, we integrate physical modelling concepts in CSS to map sound units in a feature-based space driven by physical attributes of real-world objects for real-time virtual navigation and synthesis. In avoiding an feature space driven from audio descriptions in CSS, we minimize the drawbacks of artificial mappings across corpora and ensure perceptual congruence between features, while guaranteeing diversity for the same physical gestures. In sum, the multimodal visual, aural and haptic feedback ultimately foster a correspondence across the physical and the experienced interaction.

In addressing perceptual multimodal domains, our most pressing step is to evaluate the framework through multiple user studies. We aim to assess the experience of congruent interactions and optimize the multiple parameterizations of the system behind our case study. Moreover, the studies will assess participants' sense of agency and presence during interaction

Furthermore, user studies will inform optimal factors and parameterizations for each case study, such as finding a sensible bag-of-features from the interaction primitives which better projects the sonic properties of interaction gestures; optimal unit duration per surface or the degree of penalty in the target cost computation for minimizing stuttering. An anticipatory model for predicting longer sequences trajectory ahead of real-time physical interactions will be considered to minimize the latency that is introduced by the multiple components elements of the framework. Finally, the feature vector presented in the case of study is not definitive and we plan to expand and implement the described features after testing and stressing both technical limitations and perceptual variables as stated above. However, we believe that this system can benefit several simulations environments and contribute to a better integration of real world sources and simulated experiences through better realism approximations and the accuracy of sonic interaction design.

Acknowledgments

This work was supported by the Portuguese Foundation for Science and Technology (PD/BD/114140/2015).

References

[1] G. Bernardes, L. Aly, and M. E. Davies. Seed: Resynthesizing environmental sounds from examples. In *Proceedings of the Sound and Music Computing Conference*, 2016.

[2] M. Casey. Soundspotting: A new kind of process? In *The Oxford Handbook of Computer Music*. 2009.

[3] L. S. Chan and Kup-Sze Choi. Integrating physx and openhaptics: Efficient force feedback generation using physics engine and haptic devices. In *2009 Joint Conferences on Pervasive Computing (JCPC)*, pp. 853–858, Dec 2009. doi: 10.1109/JCPC.2009.5420068

[4] A. Di Scipio. Synthesis of environmental sound textures by iterated nonlinear functions. In *Proceedings of the 2nd COST G-6 Workshop on Digital Audio Effects (DAFx99)*, pp. 109–117, 1999.

[5] E. Magalhães, E. R. Høeg, G. Bernardes, J. R. Bruun-Pedersen, S. Serafin, and R. Nordahl. Combining texture-derived vibrotactile feedback, concatenative synthesis and photogrammetry for virtual reality rendering. pp. 348–355, 2019.

[6] D. Rocchesso, P. Polotti, and S. D. Monache. Designing continuous sonic interaction. 3(3):13–25. doi: 10.1109/MC.2004.129

[7] N. Schnell, A. Röbel, D. Schwarz, G. Peeters, R. Borghesi, et al. Mubu and friends–assembling tools for content based real-time interactive audio processing in max/msp. In *Proceedings of the International Computer Music Conference*, 2009.

[8] D. Schwarz. Concatenative sound synthesis: The early years. *Journal of New Music Research*, 35(1):3–22, 2006.

[9] K. van den Doel, P. G. Kry, and D. K. Pai. FoleyAutomatic. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01*, pp. 537–544. doi: 10.1145/383259.383322

[10] Y. Visell, F. Fontana, B. L. Giordano, R. Nordahl, S. Serafin, and R. Bresin. Sound design and perception in walking interactions. *International Journal of Human-Computer Studies*, 67(11):947–959, 2009.