Perception-Based Interactive Sound Synthesis of Morphing Solids' Interactions

Laurent Pruvost, Bertrand Scherrer, Mitsuko Aramaki, Sølvi Ystad, Richard Kronland-Martinet *
LMA, CNRS, UPR 7051, Aix-Marseille Univ, Centrale Marseille
lastname@lma.cnrs-mrs.fr

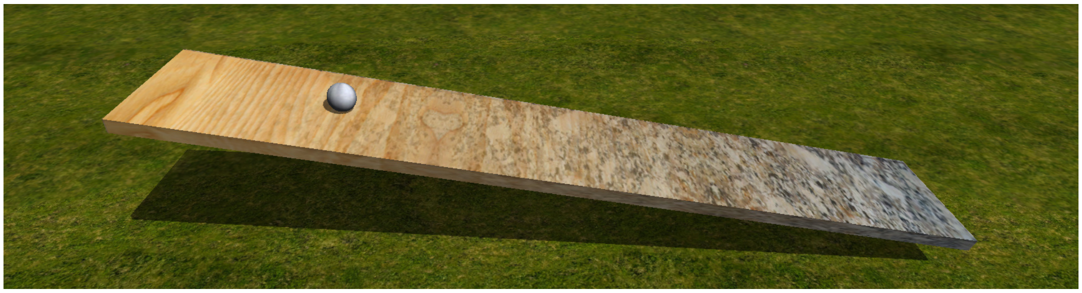

Figure 1: *A metal ball is rolling on a plate whose material cross-fades from wood to stone. How does it sound ?*

Abstract

This brief introduces a novel framework for the interactive and real-time synthesis of solids' interaction sounds driven by a game engine. The sound synthesizer used in this work relies on an *action-object* paradigm, itself based on the notion of *perceptual invariants*. An intuitive control strategy, based on those invariants and inspired by physics, was developed. The *action* and the *object* can be controlled independently, simultaneously, and continuously. This allows the synthesis of sounds for solids' interactions whose nature evolves continuously over time (*e.g.* from rolling to slipping) and/or where the objects' properties (shape, size and material) vary continuously in time.

CR Categories: H.5.5 [Information Systems]: Information Interfaces and Presentation—Sound and Music Computing I.6.8 [Computing Methodologies]: Simulation and Modeling—Types of Simulation K.8.0 [Computing Milieux]: Personal Computing—General

Keywords: perceptual invariants, procedural audio synthesis, solids' interaction, real-time

1 Introduction

This brief introduces a novel framework for the interactive and real-time synthesis of solids' interaction sounds driven by a game engine. This work situates itself within the growing research on procedural audio generation, which aims at replacing the use of pre-recorded audio samples. The sound synthesizer at the core of this work is based on an *action-object* paradigm whereby sounds are described as the result of an action on an object [Gaver 1993]. This paradigm assumes the existence of auditory invariants, *i.e.* perceptually relevant signal morphologies that carry information about the *action* and/or the *object* involved in sound production. In this context, an intuitive control of sound synthesis was developed [Aramaki et al. 2011; Conan et al. 2014]. This control allows for continuous navigation in two different spaces defining *action*'s and *object*'s properties. It is inspired by physics, however it is not intended to be very accurate in that regard. Indeed, the actual goal is perceptual relevance.

It is both the interactive nature of the framework – the sound synthesizer runs in real-time – and its ability to smoothly morph between different actions and objects that differentiates this work from previous work on the control of modal synthesis by game engine collision events [Van Den Doel et al. 2001; Zhimin et al. 2010] or on the very accurate modeling of non-linear sound production of virtual objects [Chadwick et al. 2009]. Compared to [Verron et al. 2013], the framework discussed in Sec. 2 includes recent developments in the modeling of linear interactions [Conan et al. 2014] and non-linear coupling [Thoret et al. 2013], as outlined in Sec. 3. In addition, the synthesizer's control has been overhauled. In particular, a new control space defining the *object*'s properties has been introduced. This aspect of the work is detailed in Sec. 4.

2 Framework Overview

The framework we present in this brief includes a control interface (computer tablet) connected to a game engine which drives a sound synthesizer. The elements of the framework communicate using the Open Sound Control (OSC) protocol[1] (see Fig. 2).

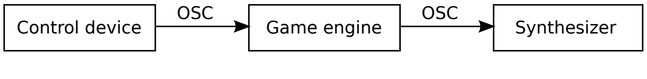

Figure 2: *Elements of the framework*

The accompanying video [SIGGRAPH 2015] includes 5 sections that illustrate our framework's capabilities when it comes to interactive action morphings (rolling/sliding in section 1, squeaking/singing in section 2) and object morphings (material morphing in section 3, size morphing in section 4). The section 5 features

*This work was funded by the French National Research Agency (ANR) under the PHYSIS project (ANR-12-CORD-0006) and SoniMove project (ANR-14-CE24-0018). The authors thank AudioGaming for their help in the development of the demo scenes.

[1] http://opensoundcontrol.org/introduction-osc

a more elaborate environment (a labyrinth game) which offers a global overview of the framework's capabilities.

3 Sound Synthesizer

3.1 Synthesis overview

In order to synthesize the sound of an interaction between two solids, we propose a double source-filter model (see Fig. 3). The two source signals – which simulate the *actions* exerted by each solid on the other one – are first processed by blocks whose function is to emulate the non-linearities that can arise in the case of a strong coupling between the solids. The resulting signals are then fed into banks of resonant filters which simulate the *objects* vibroacoustic response. Finally, the filter banks' outputs are summed to yield the interaction sound. Since multiple interactions can occur simultaneously, we use an object-oriented approach where an instance of the double source-filter module described above is created (freed) each time an interaction starts (ends). During the lifetime of an instance, parameters relevant to *actions* (*i.e.* source signals) and *objects* (*i.e.* filter banks) are refreshed on the fly at every time-step of the game's physics engine in order to handle action and/or object morphing.

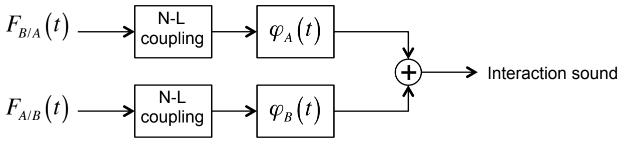

Figure 3: *Double source-filter model used to synthesize the sound of an interaction between solid A and solid B. The source signal $F_{B/A}(t)$ (resp. $F_{A/B}(t)$) simulates the action exerted by B on A (resp. by A on B). The filter bank $\varphi_A(t)$ (resp. $\varphi_B(t)$) simulates the vibrational response of A (resp. B).*

3.2 Simulation of interaction forces

In the case of an impact, the source signals ($F_{./.}(t)$ in Fig. 3) are impulses parametrized by their duration and maximal amplitude, whose values can be computed using a model that takes into account the impact speed, along with the solids' mass and elasticity. Such a model can be found in [Graff 1975].

In the case of a continuous interaction, the source signals are computed thanks to the model proposed in [Conan et al. 2014], which handles friction and rolling *actions*. In short, this kind of source signals are modeled as series of micro-impacts whose amplitudes, durations and occurrence times are random variables. The perceived nature of the synthesized *action* is controlled via the statistics of those random variables. One very important aspect of this model for the framework discussed in this brief is that it can generate source signals corresponding to hybrid *actions*, between rolling and friction. This is illustrated in section 1 of the accompanying video [SIGGRAPH 2015], where a solid alternatively rolls and slides due to the morphing of its shape.

Our system also includes a non-linear friction model (the "N-L coupling" blocks in Fig. 3) proposed in [Thoret et al. 2013]. This model is able to smoothly transform "normal" rubbing sounds into squeaking or auto-oscillation sounds. These three kind of sounds are those one can hear, for instance, when tying to make a wine glass "sing" by rubbing a wet finger on its rim. Such a scene is featured in section 2 of the accompanying video [SIGGRAPH 2015] in order to demonstrate the model's capabilities. The model is in practice implemented as a bank of resonant filters. The squeaky effect is obtained by setting the filters so as to obtain an harmonic comb filter whose frequencies fluctuate with time. These random fluctuations happen around mean values which can be subject to sudden jumps depending on physical parameters of the interaction (*i.e.* relative speed and normal force). This mimicks bifurcation phenomena observed in real squeaking sounds. A smooth transformation from "normal rubbing" to "full squeaking" is achieved by a continuous increase of the filters' gain. Auto-oscillation source signals are obtained using a stationary and possibly inharmonic filter bank, whose frequencies correspond to natural frequencies of the solid. In addition, amplitude modulation is applied to the source signals since it was observed in real auto-oscillation sounds. A smooth transformation from squeaking to auto-oscillation is obtained by progressively moving the filters' frequencies from the frequencies of squeaking to natural frequencies of the solid. In the same time, the amplitude of the filter bank's frequency jitter is progressively brought to zero while the depth of amplitude modulation is smoothly increased from zero to its maximal value.

3.3 Simulation of the Object's Response

The object's vibrational response is simulated using a bank of resonant filters, where each filter corresponds to a flexural mode of the object. Each filter is fully defined by three parameters: its resonant frequency, its gain and its damping factor. Technically speaking, filters are implemented as proposed by [Mathews and Smith 2003] in order to avoid audible transients when the filters' parameters are changed rapidly. This is tantamount when synthesizing the sound of a solid subject to shape and/or material morphing. Such scenarii are featured in the accompanying video [SIGGRAPH 2015] : in section 3, a ball rolls on plates whose material continuously changes; in section 4, a ball sees its size evolve as it rolls.

In order to limit CPU load – thus making sure that the synthesizer can run in real-time – a compromise has to be sought between, on the one hand, the number of sounding objects handled simultaneously by the synthesizer and, on the other hand, the accuracy of their modeling. Let M be the number of sounding objects and N the number of filters used to model each object, the computer has to manage $M \times N$ channels of signal processing. The Bark scale defines 25 critical frequency bands over the human hearing range, and the number of filters per critical band can be limited to 7 with no dramatic loss in sound quality [Hartmann et al. 1986]. As a consequence, setting $N = 25 \times 7 = 175$ filters per object can be viewed as a reasonable target. One may consider lowering this value – to the detriment of sound quality – if many sounding objects have to be handled simultaneously. It should be noted that an extra sounding object increases the CPU load only if it differs (by means of shape, size, or material) from the other objects. Indeed, in the case where the scene features several similar objects (same shape, same size and same material), all the clones can share the same filter bank with a single input formed by summing all the signals that simulate the interaction forces applied to each clone.

4 Control of the Synthesizer

The synthesizer discussed above is controlled using high-level semantic descriptors since it was initially developed for sound designers [Aramaki et al. 2010; Conan et al. 2014]. This original control strategy is based on research on the auditory perception of both *action*'s and *object*'s characteristics. In the following, we present these high-level parameters that are communicated from the game engine to the synthesizer in order to describe the ongoing interaction as well as the solids involved in it. We also detail the mapping between these high-level parameters and the synthesizer's low-level parameters.

4.1 Action control

Source signals control As stated in Sec. 3.2, in the case of an impact the synthesizer generates the relevant source signal thanks to a model whose inputs are the impact speed, the mass and the elasticity (Young's modulus) of the colliding solids. The impact speed is directly provided by the game's physics engine, but the game's developer has to manually set the density and Young's modulus of each object. Since solids are allowed to change size, the game engine updates the objects mass by multiplying their density with their volume. Besides impacts, the synthesizer handles a continuum of solids' interactions that spans from friction to rolling [Conan et al. 2014]. From the kinematic point of view, it corresponds to a continuum ranging from pure slipping to rolling without slipping. During the course of an interaction, the relative kinematics of the solids are computed on the fly and the *action*'s nature is described using a scalar ranging from 0 (for pure friction) to 1 (for pure rolling). Besides this value, a few other perceptually relevant parameters are continuously sent to the synthesizer in order to generate the proper source signals: the objects' size, asymmetry, roughness, and relative speed. For now, the roughness and asymmetry parameters are set manually while the size and speed parameters are automatically and continuously updated by the game engine.

Non-linear coupling control Research on modeling the kinematics underlying the *actions* of rubbing, squeaking and auto-oscillation goes beyond the scope of this brief. For that reason, a simplistic model was implemented to demonstrate the capabilities of the synthesizer to morph between rubbing, squeaking, and auto-oscillation sounds (see section 2 of the accompanying video [SIGGRAPH 2015]). In this model, the *action*'s nature is simply mapped to the relative velocity of the interacting solids.

4.2 Object control

High-level controls of the objects relate to their shape, size, material, stiffness, and contact point. Although stiffness physically depends on shape and material, we chose to decouple these variables in order to add a degree of freedom in the sound design process.

Material Modal damping was found to play a major role in the auditory identification of materials [Aramaki et al. 2011; Rakovec et al. 2013]. Moreover the couple (α_G, α_R), which appears in the damping model given in Eq.1, was shown to characterize a given material.

$$\alpha(f) = e^{\alpha_G + 2\pi f \alpha_R} \tag{1}$$

Besides, the aforementioned studies identified several materials that a listener can faithfully recognize. Among them, five are handled by our synthesizer: wood, stone, plastic, glass, and metal. Not only can they be used as such, but they can also be combined to get hybrid materials simply by taking a weighted average of the α_G and α_R values of the 5 basic materials. The weights can be changed on the fly, either discretely or continuously in order to get a *material morphing* effect (see section 3 of the accompanying video [SIGGRAPH 2015]).

Shape From [Tucker and Brown 2002] we inferred that the objects' shape could be described using a rough model with little loss in the realism of the synthesized sound. Besides, [Rakovec et al. 2013] found that the dimension[2] and the massive/hollow character of an object are perceptually relevant parameters when it comes to describing the shape of a sounding object. We now introduce an alternative set of descriptors that spans the same shape-space as Rakovec's set (see Fig. 4): width, thickness, curvature, and size. This new description of an object's shape was preferred because it seems more intuitive. Besides, the shape control based on these new descriptors allows for a continous shape morphing, as each descriptor is a continuous variable. Regarding acoustics, shape morphing results in timbre morphing. In practice, changing the shape of an object modifies the parameters of the filter bank that simulates its response. At the heart of our morphing system lies the well-known equation that gives the natural frequencies for the flexural modes of a rectangular membrane fixed on its edges:

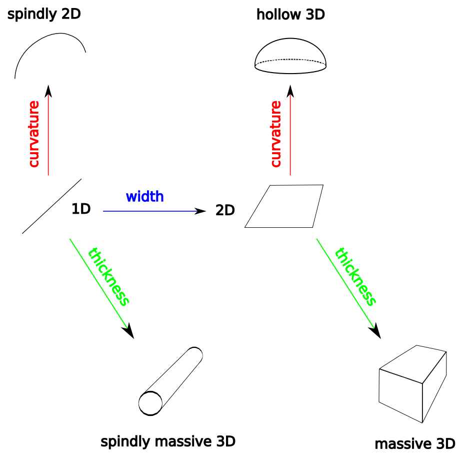

Figure 4: *Object space spanned by the width, curvature, and thickness controls.*

$$f_{mn}(\alpha, L) = f_0(L)\sqrt{\frac{m^2}{\alpha^2} + n^2} \xrightarrow[\alpha\to 0]{} f_n(L) = f_0(L)n \tag{2}$$

where L is the length (its greater dimension) of the membrane. $\alpha \in]0,1]$ is such that the membrane's width is $W = \alpha L$. Here f_0, which depends on the object's largest dimension L, stands for the lowest natural frequency f_{01}, and will be referred to as "fundamental frequency" in the following. According to Eq. 2, $f_{mn}(\alpha, L)$ tends[3] towards $f_n(L)$, which are the natural frequencies for the the flexural modes of a string when $\alpha \to 0$. This formula (Eq. 2) can therefore be used to produce a continuous timbre transformation corresponding to the morphing of a 2D object into a 1D object (and vice-versa).

This basic shape living between 1D and 2D can be viewed as a *seed*, to whom curvature and thickness can be given so as to create more complex shapes.

In order to take thickness into account, Eq. 2 is extended to include a third parameter, in analogy to the formula for the resonance frequencies of a rectangular cavity:

$$f_{mnp}(\alpha, \beta, L) = f_0(L)\sqrt{\frac{m^2}{\alpha^2} + \frac{p^2}{\beta^2} + n^2} \xrightarrow[\beta\to 0]{} f_{mn}(\alpha, L) \tag{3}$$

with $\beta \in]0, \alpha]$. The thickness is $T = \beta L$, with $T \leq W \leq L$. It should be noted that $f_{mnp}(\alpha, \beta, L)$ tends – again in the perceptual

[2] Here "dimension" has to be understood in the sense of mathematical dimensionality, *i.e.* the minimal number of dimensions of the space in which the object is embedded.

[3] This limit has to be understood in the perceptual sense. Indeed, when $\alpha \to 0$ the m-related modes are pushed beyond the upper limit of the human hearing range; they can then be neglected, leaving only the n-related modes to synthesize.

sense – towards $f_{mn}(\alpha, L)$ when $\beta \to 0$. Hence Eq. 3 can be used to produce a continuous timbre transformation corresponding to the morphing of a massive 3D object into a 2D object (and vice-versa). Besides, we found that thickness is better perceived if the global damping (α_G in Eq. 1) is increased.

Eq. 4 gives the formula we use to produce the timbre variation corresponding to the curving of a solid. For instance, it allows to transform a rod into a bow, or a square plate into a spherical cap.

$$f_{mn(p)} \longmapsto \sqrt{f^2_{mn(p)} + \frac{E}{\rho R}} \tag{4}$$

In Eq. 4, R is the curvature radius. E and ρ are the Young's modulus and density of the solid's material, respectively. This transformation was inspired by the work of [Soedel 2004], who studied the effect of curvature on the natural frequencies of plates.

Contact point The vibrational response of an object depends on the point where the mechanical excitation is applied. In our system, the contact point is used to set the gains of the resonant filters (the A_{mn} in Eq. (5)), which are continuously updated during the course of an interaction to obtain relevant sound effects. In practice, the coordinates of the contact point are first transformed into the *seed*'s local coordinates system, yielding $(X, Y) \in [0, W] \times [0, L]$, then the A_{mn} are given the amplitudes of the *seed*'s flexion mode shapes at (X, Y) for the (m, n) mode:

$$A_{mn} = \sin\left(\frac{m\pi X}{W}\right) \sin\left(\frac{n\pi Y}{L}\right) \tag{5}$$

Size The auditory perception of an object's size is related to the pitch. In our system, the pitch is modified as a function of the object's largest dimension L via the fundamental frequency: $f_0(L) = f_0^{\text{ref}} L^{\text{ref}}/L$. Since objects' dimensions are unitless inside the game engine, the synthesizer needs a $(L^{\text{ref}}, f_0^{\text{ref}})$ reference couple to be able to set an absolute f_0 given L. This reference couple has to be set manually by the game's developer.

Stiffness In our system, stiffening is rendered by an increase in spectral inharmonicity thanks to Eq. 6, which is taken from [Aramaki et al. 2010]. Using this formula, setting $\sigma = 1$ and $c = 0.5$ for instance will make a string (resp. a membrane) sound more like a bar (resp. a plate).

$$f_{mn} \longmapsto f_{mn} \cdot \left(1 + \sigma \cdot \left(\frac{f_{mn}}{f_0}\right)^2\right)^c \text{, with } \sigma \in [0, 1] \tag{6}$$

5 Conclusion

We present a novel framework for the interactive and real-time synthesis of solids' interaction sounds driven by a game engine. The models and the sound synthesis engines associated to different interactions were developed in previous studies. The control of the synthesizer was adapted to enable its connection to a game engine.

References

Aramaki, M., Gondre, C., Kronland-Martinet, R., Voinier, T., and Ystad, S. 2010. Imagine the sounds: An intuitive control of an impact sound synthesizer. In *Auditory Display*, S. Ystad, M. Aramaki, R. Kronland-Martinet, and K. Jensen, Eds., vol. 5954 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 408–421.

Aramaki, M., Besson, M., Kronland-Martinet, R., and Ystad, S. 2011. Controlling the Perceived Material in an Impact Sound Synthesizer. *IEEE Transactions on Audio, Speech and Language Processing 19*, 2 (February), 301–314.

Chadwick, J. N., An, S. S., and James, D. L. 2009. Harmonic shells: A practical nonlinear sound model for near-rigid thin shells. In *ACM Transaction on Graphics (SIGGRAPH ASIA Conference Proceedings)*.

Conan, S., Thoret, E., Aramaki, M., Gondre, C., Ystad, S., Kronland-Martinet, R., and Derrien, O. 2014. An intuitive synthesizer of continuous interaciton sounds: Rubbing, scratching and rolling. *Computer Music Journal 38*, 4, 24–37.

Gaver, W. W. 1993. How do we hear in the world? Explorations in ecological acoustics. *Ecological psychology 5*, 4, 285–313.

Graff, K. 1975. *Wave Motion in Elastic Solids*. Dover Books on Engineering Series. Dover Publications.

Hartmann, W. M., McAdams, S., Gerzso, A., and Boulez, P. 1986. The discrimination of spectral density. *Journal of the Acoustical Society of America 79*, 1915–1925.

Mathews, M., and Smith, J. 2003. Methods for synthesizing very high Q parametrically well behaved two pole filters. In *Proceedings of the Stockholm Musical Acoustics Conference (SMAC 2003)(Stockholm)*, Royal Swedish Academy of Music.

Rakovec, C.-E., Aramaki, M., and Kronland-Martinet, R. 2013. Perception of Material and Shape of Impacted Everyday Objects. In *Proc. of 10th CMMR*.

SIGGRAPH, 2015. The accompanying video to the present brief can be found at: `http://www.lma.cnrs-mrs.fr/~kronland/siggraph2015/index.html`.

Soedel, W. 2004. *Vibrations of Shells and Plates*, third ed. Marcel-Dekker, New York.

Thoret, E., Aramaki, M., Gondre, C., Kronland-Martinet, R., and Ystad, S. 2013. Controlling a non linear friction model for evocative sound synthesis applications. In *International Conference on Digital Audio Effects (DAFx), Maynooth, Ireland, Sept. 2013*.

Tucker, S., and Brown, G. J. 2002. Investigating the perception of the size, shape and material of damped and free vibrating plates. Tech. Rep. CS-02-10, Department of Computer Science, University of Sheffield.

Van Den Doel, K., Kry, P. G., and Pai, D. K. 2001. FoleyAutomatic: physically-based sound effects for interactive simulation and animation. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, 537–544.

Verron, C., Aramaki, M., Gonot, A., Scotti, T., Rakovec, C.-E., Mingasson, A., and Kronland-Martinet, R. 2013. Event-driven interactive solid sound synthesis. In *Proceedings of the 10th International Symposium on Computer Music Multidisciplinary Research*.

Zhimin, R., Hengchin, Y., and Lin, M. C. 2010. Synthesizing contact sounds between textured models. In *Proceedings of the 2010 IEEE Virtual Reality Conference*, IEEE Computer Society, vol. 139-146.