PHYSICS-BASED SOUND SYNTHESIS AND CONTROL: CRUSHING, WALKING AND RUNNING BY CRUMPLING SOUNDS

Federico Fontana

Dipartimento di Informatica
University of Verona
`fontana@sci.univr.it`

Roberto Bresin

Speech, Music and Hearing, TMH
Royal Institute of Technology, KTH
`roberto@speech.kth.se`

ABSTRACT

Three types of ecological events (crushing, walking and running) have been considered. Their acoustic properties have been modeled following the physics-based approach. Starting from an existing physically-based impact model, we superimposed to it the dynamic and temporal stochastic characteristics governing crushing events. The resulting model was triggered by control rules realizing typical walking and running time patterns.

This bottom-up design strategy was made possible because the sound synthesis and sound control models could be directly connected each other via a common switchboard of driving and control parameters. The existence of a common interface specification for all the models follows from the application of physics-based modeling, and translates in major advantages when those models are implemented as independent, self-contained blocks and procedures connected together in real-time inside a sw architecture like *pd*.

1. INTRODUCTION

Physics-based sound synthesis has its roots in traditional paradigms for the generation of sound for musical purpose. In these paradigms the synthesis model is not specified by requirements on the signal, but by understanding the generation process and its effects on sound production [1].

This approach to sound design, hence, deals with the cause and not with the effect. In other words we can say that sound is obtained by designing (or modeling) a corresponding process, whatever it is and even regardless of its effects. It is not an uncommon result that a process, in the end, comes out to be unsuitable for sound generation. Nevertheless, there are cases in which a given process produces effective and convincing sounds.

Such an approach leads to models whose control is specified in parameters having a direct relationship with the synthesis process. On the other hand, any control devoted to manipulate specific sound features must be applied indirectly through a map, that acts as an intermediate layer in between the synthesis engine and the control panel. For this reason, the physics-based approach is generally not so effective in contexts where sound can be effectively controlled using traditional signal-based operations (i.e. spectral manipulations).

In the case of a synthesis process that models the evolution of a physical system, we are in front of a *special* sound synthesizer. Once its suitability to sound generation has been proved, this synthesizer offers peculiar (i.e., physical) control parameters directly available to the user. By means of those parameters the user can directly select quantities such as *force*, *pressure*, *mass*, and *elasticity*. It is clear that a control panel like that is not optimal for all contexts. At the same time it invites the user to explore a wide range of possibilities by real-time direct manipulation of the control parameters.

Physical modeling has been successfully applied to musical instrument modeling [2, 3, 4, 5, 6, 7], mainly because the ease of control of peculiar parameters, such as the force applied to a key by the player, in a virtual piano model [8]. On the other hand, physical modeling hardly surpasses synthesis techniques based on signal sampling if only the parameter of sound reproduction accuracy is taken into account during the evaluation of a synthesizer. Physical modeling is competitive as long as the level of interactivity between the user and the virtual instrument is considered as a parameter of quality. Though, the exploration of physical models is relatively at an early stage. The horizon at which designers can look at is still heavily bounded by the amount of computational resources they can afford, as long as the real-time constraint comes into play along with reasonable limits in the cost of the final application. Unfortunately, despite a moderate need of memory, physical models demand a large amount of computational power (for this reason, techniques aiming at transforming at least part of those computational power requirements into figures of memory occupation have been proposed [9]). This is not the case of sampling: since a large amount of fast-access memory is available at a low cost, designers can store huge databases of sound samples in the application.

Physics-based (or physically-based) sound synthesis relocates the physical modeling background into a different framework, that is, the synthesis of *ecological* sounds [10]. Those sounds are normally experienced during everyday listening, and contribute to our perception of the world. Sounds accounting for events such as the bouncing of an object falling on the floor, the friction between two objects, a crushing can, the footsteps of a person, are detected, recognized and possibly localized by our hearing system. These sounds convey *vital* information, via the auditory channel, about the surrounding environment, and their reproduction is very interesting in applications involving audio virtual reality (or audio VR). The more realistic the virtual scenario, the more effective the audio VR system is. Hence, we must not be surprised if the simulation of an everyday listening environment becomes much more realistic when it conveys also the auditory information about ecological events such as those seen above [11]. In this paper we will deal in particular with *crushing*, *walking* and *running*.

Since physics-based sound synthesis is suitable for modeling elementary ecological sounds, such as impacts and collisions, the non-linear hammer-string interaction model [8] can be imported into a new model for the reproduction of the most relevant physical phenomena occurring during a collision (impact) between two objects [12, 13, 14]. The physics-based model yields, at its interface level, a set of controls for recreating realistic collisions between objects made of different materials. The same controls enable the user a direct interaction with the virtual objects.

Since ecological sounds describe higher-level events such as crushing or walking [15], the design of the impact model described in this paper follows a bottom-up approach, i.e. *starting from the elementary model to build up higher-level models*. In this way we follow the original physical modeling approach, in which a virtual musical instrument uses to be assembled by putting together some basic, general-purpose building blocks.

In our case, the basic building block is the impact model [12]. In Section 2 and 3 we will describe the way individual impacts can be assembled together producing a single crushing event. As we will see, crushing events are the result of a statistical rather than deterministic sequence of impacts, in a way that we are not asked to look for formulas valid in the discrete-time, expressing physical relationships between different types of sounding objects (those relationships must be found out, for example, in the study of contact sounds: manipulating them is usually not easy [14]). We instead will find a way to create consistent (from a psychophysical point of view) collections of “atomic” (impact) sounds starting from the stochastic description of so-called *crumpling sounds*.

In Section 4, we will describe how a set of crushing events can be controlled by *rules* so to produce realistic sequences of walking and running sounds.

2. CRUMPLING SOUND

Crumpling sounds occur whenever our hearing system identifies a source whose emission, for some reason, is interpreted as a superposition of microscopic crumpling events.

Aluminum cans emit a characteristic sound when they are crushed by a human foot that, for example, compresses them along the main axis of the cylinder. This sound is the result of a composition of single crumpling events, each one of those occurring when, after the limit of bending resistance, one piece of the surface forming the cylinder splits into two facets as a consequence of the force applied to the can.

The exact nature of a single crumpling event depends on the local conditions the surface is subjected to when folding occurs between two facets. In particular, the types of vibrations that are produced are influenced by shape, area, and neighborhood of each facet. Also other factors play a role during the generation of the sound, such as volume and shape of the can. The can acts as a volume-varying resonator during the crumpling process.

A precise assessment of all the physical factors determining the sound which is produced by a single crumpling event is beyond the scope of this work. Moreover, there are not many studies available in the literature outlining a consistent physical background for this kind of problems. On the other hand, it is likely that our hearing system cannot distinguish such factors but the most relevant ones. For this reason we generate individual crumpling sounds using the impact model.

Studies conducted on the acoustic emission from wrapping sheets of paper [16] concluded that crumpling events do not determine *avalanches* [17], so that fractal models in principle cannot be used to synthesize crumpling sounds [18]. Nevertheless, crumpling paper emits sound in the form of a stationary process made of single impulses, whose individual energy E can be described by the following power law:

$$P(E) = E^{\gamma} \quad , \tag{1}$$

where γ has been experimentally determined to be in between -1.3 and -1.6. On the other hand a precise dynamic range of the impulses is not given, although the energy decay of each single impulse has been found out to be exponential.

The temporal patterns defined by the events is an important factor determining the perceptual nature of the crumpling process. A wide class of stationary temporal sequences can be modeled by *Poisson's processes*; each time gap τ between two subsequent events in a temporal process is described by an exponential random variable with *density* $\lambda > 0$ [19]:

$$P(\tau) = \lambda e^{\lambda\tau} \ \text{ with } \ \tau \geq 0 \quad . \tag{2}$$

Assuming a time step equal to T, we simply map the time gap over a value kT defined in the discrete-time domain:

$$k = \text{round}(\tau/T) \quad , \tag{3}$$

where round$(\cdot)$ gives the closest integer to its argument value.

The crumpling process consumes energy during its evolution. This energy is provided by the agent that crushes the can. The process terminates when the transfer of energy does not take place any longer, i.e., when a *reference energy*, E_{tot}, has been spent independently by each one of the impulses forming the event s_{tot}:

$$s_{\text{tot}}[nT] = \sum_i E_i s[nT - k_i T] \quad \text{with} \quad E_{\text{tot}} = \sum_i E_i \quad , \tag{4}$$

where $s(nT)$ is a signal having unitary energy, accounting for each single crumpling.

In order to determine the dynamic range, suppose to constrain the individual energy E to assume values in the range $[m, M]$. The probability P that an individual impulse falls in that range is, using the power law expressed by (1):

$$P[m \leq E < M] = \int_m^M E^{\gamma} dE = 1 \quad . \tag{5}$$

This equation allows to calculate an explicit value for m if M is set to be the value corresponding to full-scale, beyond which the signal would clip. In this case we find out the minimum value coping with (5):

$$m = \{M^{\gamma+1} - \gamma - 1\}^{\frac{1}{\gamma+1}} \quad . \tag{6}$$

2.1. Driving the impact model

During a crushing action, the creases on the object's surface become increasingly dense. Hence, vibrations over the facets increase in pitch since they are bounded within areas that become progressively smaller. This hypothesis inspires our model for determining the pitch in a single crumpling sound.

Given a segment having a nominal length D_0, initially marked at the two ends, let us start the following procedure: Each time a new impulse is triggered, a point of this segment is randomly selected and marked. Then, two distances are measured between the position of this mark and its nearest (previously) marked points. The procedure is sketched in Figure 1, and it is repeated until some

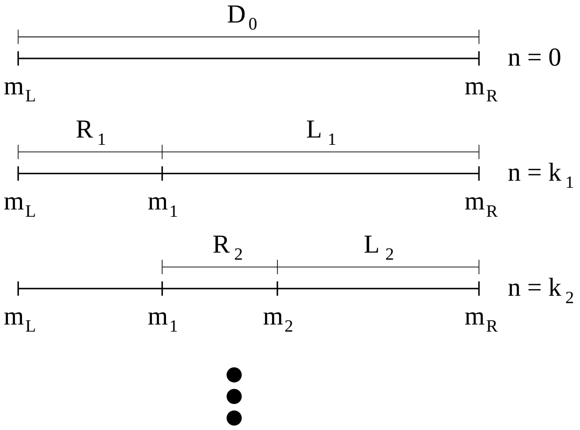


Figure 1: *Sketch of the procedure used to calculate the pitch of the impulses as long as the process evolves.*

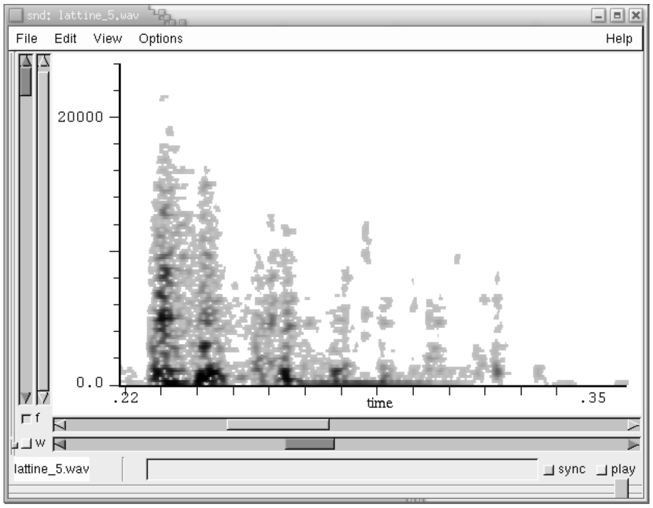


Figure 2: *Spectrogram of the prototype sound of a crushing can.*

energy, as expressed by (4), is left to the process. The values L_i and R_i, corresponding to the distances calculated between the new mark m_i (occurring at time step k_i) and the leftward and rightward nearest marks (occurred at previous time steps), respectively, are used as absolute values for the calculation of two driving frequencies, f_L and f_R, and two decay times, τ_L and τ_R, and also as relative weights for sharing the energy E_i between the two impacts, x_{f_L,τ_L} and x_{f_R,τ_R}, forming each crumpling sound:

$$\begin{aligned} E_i s[nT - k_iT] &= E_i \frac{L_i}{L_i + R_i} x_{f_L,\tau_L}[nT - k_iT] \\ &+ E_i \frac{R_i}{L_i + R_i} x_{f_R,\tau_R}[nT - k_iT] \quad , \end{aligned}$$

where the driving frequencies (decay times) are in between two extreme values, f_{MAX} (τ_{MAX}) and f_{MIN} (τ_{MIN}), corresponding to the driving frequencies (decay times) selected for a full and a minimum portion of the segment, respectively:

$$\begin{aligned} f_L &= f_{\mathrm{MAX}} - \frac{L_i}{D_0}(f_{\mathrm{MAX}} - f_{\mathrm{MIN}}) \\ f_R &= f_{\mathrm{MAX}} - \frac{R_i}{D_0}(f_{\mathrm{MAX}} - f_{\mathrm{MIN}}) \quad , \end{aligned}$$

in which the symbols f must be substituted by τ in the case of decay times.

We decided to operate on the so-called “frequency factor” and decay time of the impact model: the former is related to the size of the colliding object; the latter accounts for the object material [14]. We considered both of them to be related to the crumpling facet area: the smaller the facet, the higher-pitched the fundamental frequency and the shorter the decay time of the emitted sound.

3. CAN CRUSHING

In the case of the aluminum can, crushing occurs in consequence of some force applied to it. This action is usually performed by an agent having approximately the same size as the can surface, such as the sole of a shoe. As the agent compresses the can, sound emission to the surrounding environment changes since the active emitting surface of the can is shrinking, and some of the creases become open fractures in the surface. We suppose that the internal pressure in the can is maximum in the beginning of the crushing process. This pressure relaxes to the atmospheric value as long as the process evolves, due to pressure leaks from the holes appearing in the surface, and due to the decreasing velocity of the crushing action. These processes, if any[1], have a clear effect on the evolution in time of the spectral energy: high frequencies are gradually spoiled of their spectral content, as it can be easily seen from Figure 2 where the spectrogram of a real can during crushing has been plotted.

The whole process is interpreted in our model as a time-varying resonating effect, realized through the use of a low-selectivity linear filter whose lowpass action over the sound s_{tot} is slid toward the low-frequency as long as the process evolves. Lowpass filtering is performed using a first-order lowpass filter [20]. In the case of crushing cans we adopted the following filter parameters:

- lowest cutoff frequency $\Omega_{\mathrm{MIN}} = 500$ Hz
- highest cutoff frequency $\Omega_{\mathrm{MAX}} = 1400$ Hz.

Using those parameters, the cutoff frequency is slid toward the lowest value as long as energy is spent by the process. More precisely, the cut frequency ω_i at time step k_i is calculated according to the following rule:

$$\omega_i = \Omega_{\mathrm{MIN}} + \frac{E_{\mathrm{tot}} - \sum_{k=1}^{i} E_k}{E_{\mathrm{tot}}} \left(\Omega_{\mathrm{MAX}} - \Omega_{\mathrm{MIN}}\right) \quad . \qquad (7)$$

This kind of post-processing contributes to give a smooth, progressively “closing” characteristic to the crumpling sound.

3.1. Parameterization

Several parameter configurations have been tested during the tuning of the model. It has been noticed that some of the parameters have a clear (although informal) *direct* interpretation:

- E_{tot} can be seen as an “image” of the *size*, i.e., the height of the cylinder forming the can. This sounds quite obvious, since E_{tot} governs the time length of the process, and this

---

[1] We are still looking for a thorough explanation of what happens during crushing.

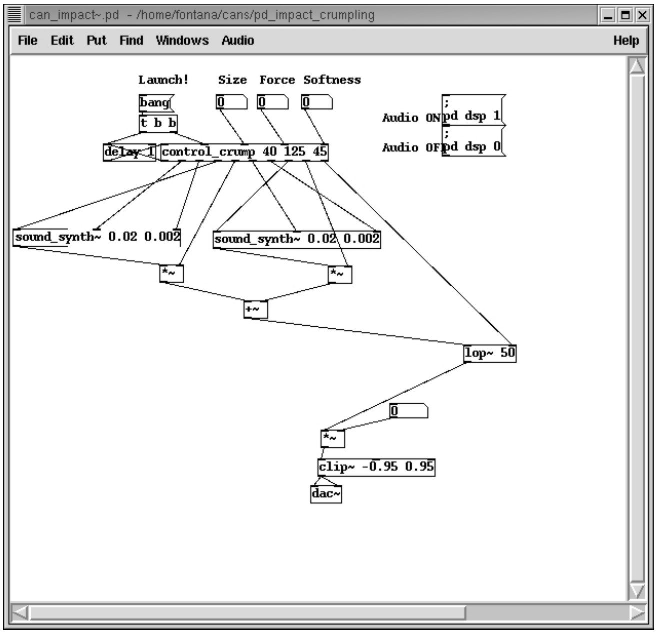


Figure 3: *Screenshot of the* `pd`*-module implementing the crushing can model.*

length can be in turn reconducted to the can size. Sizes which are compatible with a natural duration of the process correspond to potential energies ranging between 0.001 and 0.1;

- low absolute values of γ result in more regular realizations of the exponential random variable, whereas high absolute values of the exponential statistically produce more peaks in the event dynamics. Hence, γ can be seen as a control of *force* applied to the can. This means that for values around -1.5 the can seems to be heavily crushed, whereas values around -1.15 evoke a softer crushing action. Thus, γ has been set to range between -1.15 and -1.5;
- "soft" alloys forming the can can be bent more easily than stiff alloys: holding the same crushing force, a can made of soft, bendable material should shrink in fewer seconds. For this reason, the parameter p_s governing the frequency of the impulses in the Poisson process can be related to the material stiffness: the higher p_s, the softer the material. *Softness* has been set to range between 0.001 (stiff can) and 0.05 (soft can).

We noticed that variations in the definition and parameterization of the crumpling sound s can lead to major differences in the final sound. On the other hand the statistical laws which we used for generating crushing sounds are general enough for reproducing a wide class of events, including paper crumpling and plastic bottle crushing. In that case s must be properly shaped to accommodate for different kinds of events.

3.2. Implementation as *pd* patch

Crushing cans have been finally implemented as a `pd` *patch* [21]. The modular implementation of `pd` allows to have the crumpling sound synthesis model, the higher-lever statistical module, and the time-varying post-processing lowpass filter decoupled inside the same patch (see Figure 3). The only limitation with this implementation is represented by the firing rate used by the statistical module (labeled as `control_crump`) to feed control data to the two sound synthesis modules (labeled as `sound_synth~`) producing the two signals x_{f_L} and x_{f_R}. This limitation comes from the presence of a structure containing the statistical module in loop-back with a `delay` block, which limits the shortest firing rate to 1 ms.

On the other hand the chosen implementation allows a totally independent design of the statistical module and of the sound synthesis module. The statistical module has been realized in C language, as a `pd` class, whereas the sound synthesis module has been implemented as a sub-patch nesting inside itself several preexisting building blocks, which come together with `pd`. This modular/nested approach leaves the patch open to independent changes inside the modules, and qualifies the patch in its turn for use as an individual block inside higher-level patches. For this reason we could straightforwardly integrate crushing sounds in a framework including specific rules for producing walking and running sounds.

4. WALKING AND RUNNING SOUNDS

Most of sound synthesis techniques available today are capable of reproducing excellent sounds. At the same time these techniques do not take into consideration the fact that sounds usually appears in a sequence, such in the case of a music melody or of a sequence of footsteps. For example the synthesis of a sequence of trumpet sounds can give poor results if the musical context and the performance style, such as in *legato* and *staccato* articulation, is not taken into consideration. Therefore, in order to arrange single events in a meaningful way, we need control functions for the specific sound synthesis technique that is used [22].

Physics-based synthesis techniques allow a direct manipulation of parameters connected to physical properties of the sound model. These techniques are therefore particularly suitable to be controlled for the production of organized sound sequences.

4.1. Control models

Some recent works reported strong relations between body motion and music performance. This is mainly due because sound production is a result of body movements such as hands, wrists, harms, shoulders and feet in pianists and in percussionists, lips, lungs, and tung in singers and in wind instrument players. Besides these obvious relations, some researchers have found more indirect associations between music performance and body motion. Friberg and Sundberg demonstrated the way the *Final Retard* in performances of Baroque music can be derived from measurements of stopping runners [24].

In another work [26] it was found that *legato* and *staccato* playing in piano performance can be associated to walking and running respectively. During walking there is an overlap time when both feet are on the ground, and in *legato* scale playing there is an overlap time (key overlapped time) when two fingers are pressing two keys simultaneously. During running there is an time interval when both feet are in the air, and in *staccato* scale playing there is a time interval (key detached time) when all fingers are in the air simultaneously.

These and other results contributed to the design of some performance rules implemented in the Director Musices performance system [27, 28] and related to body motion. In previous experiments it has been demonstrated that *legato* and *staccato* are among the expressive parameters which help in synthesizing performances which are recognized as sad and respectively as happy by listeners [23, 29].

Given the considerations above, it was tempting trying to apply the performance rules, which present similarities with walking and running, to the control of crumpling sounds.

4.2. Controlling footstep sounds

In a first experiment, performance rules where used for the control of the sound of one single footstep extracted from a recorded sequence of walking and running sounds on gravel [26]. Stimuli including both sequences of footsteps controlled with the rules, and sequences of footsteps obtained by looping the same sound were produced. These stimuli were proposed to subjects in a listening test. Subjects could distinguish between walking and running sounds, and classified the stimuli produced with the performance rules as more natural. Still the sound controlled was a static sound. Thus, a natural development is to control a physics-based model implementing footstep sounds.

Informal listening test of the sound produced by the crushing can model, presented in Section 3, classified this sound as that of a footstep on cold snow. This result suggested the possibility of controlling the model using performance rules such those for the control of *Final Retard*, and *legato* and *staccato* articulation.

A `pd`-module implementing the control of footstep sounds is represented in Figure 4. A sequence of footstep sounds with IOI set by the "Tempo" slider is produced by pressing the `Walk` button. If the `Slow_down` button is pressed, the IOIs become larger, i.e. slower tempo. This effect is implemented using the *Final Retard* rule [24]. Parameters `Size`, `Force`, and `Softness` are set to values which give a realistic sound of footstep on cold snow. In the case of walking/running sounds these three parameters represent the size of the foot, the force applied by the foot, and the softness of the ground.

The spectrogram of walking sounds produced by the module of Figure 4 is shown in Figure 5. The IOI between footsteps increases because the `Slow_down` button was pressed and the "Final Retard" rule was activated.

An outcome of the present work is that analogies between locomotion and music performance, briefly presented in Section 4, combined with the control properties of physics-based sound modeling, open new possibilities for the design of control models for artificial walking sound patterns, and in general for sound control models based on locomotion.

5. ONGOING RESEARCH

In the particular case of walking/running sounds, we have observed that the "Force" and "Softness" parameters should vary with "Tempo". Higher tempi should correspond to larger values of the force and lower degree of softness. A next step in our research will be therefore the identification of such a mapping. Studies on expressive walking could help toward a possible solution. In a recent study [25] ground reaction force by the foot during different gaits was measured. Value and time envelope of this force varied with the different walking technique adopted by subjects. These techniques were also characterized by different tempi.

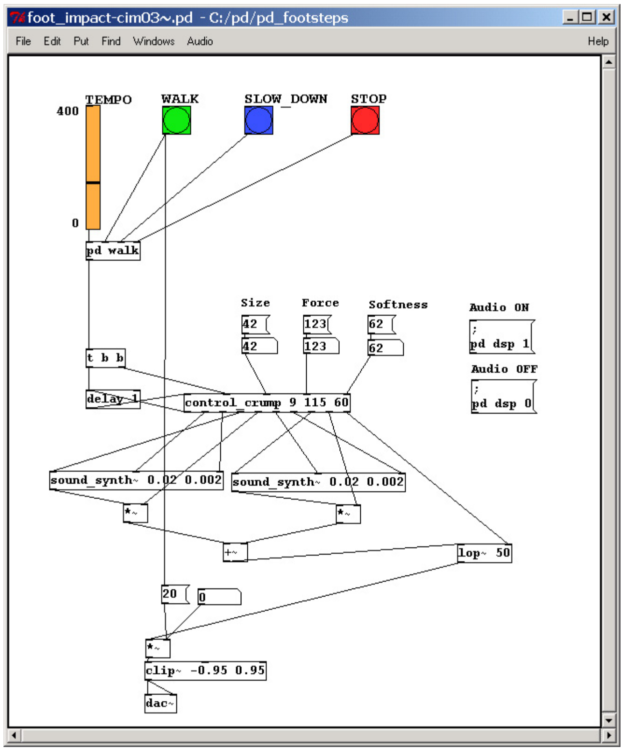
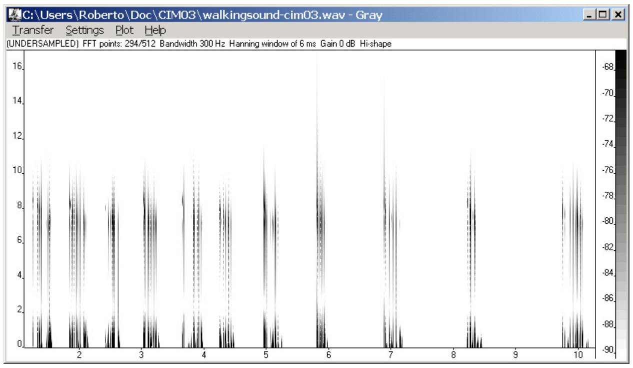


Figure 4: *Screenshot of a* `pd`*-module implementing the control of footstep sounds.* `Tempo` *(in bpm),* `Walk`*,* `Slow_Down`*, and* `Stop` *control the timing of walking steps.* `Size`*,* `Force` *and* `Softness` *determine the foot size, its force on the ground and its softness. The model controlled is that of the crushing can presented in Figure 3.*

In general, the proposed rule-based approach for sound control is only a step toward the design of more general control models that respond to physical gestures. Ongoing research in the framework of the Sounding Object project is dealing with the implementation of rule-based control of impact sound models and of friction sound models.

6. ACKNOWLEDGMENTS

This work was supported by the European Commission: SOb - The Sounding Object project, no. IST-2000-25287, `http://www.soundobject.org`.



(UNDERSAMPLED) FFT points: 294/512 Bandwidth 300 Hz Hanning window of 6 ms Gain 0 dB Hi-shape

Figure 5: *Spectrogram of a stopping walking sound produced by the* `pd`*-patch of Figure 4: the inter-onset-interval (IOI) between footsteps becomes larger.*

7. REFERENCES

[1] G. De Poli, A. Piccialli, and C. Roads, *Representations of Musical Signals*, MIT Press, Cambridge, Mass., 1991.

[2] K. Karplus and A. Strong, "Digital Synthesis of Plucked String and Drum Timbres," *Computer Music Journal*, vol. 7, no. 2, pp. 43–55, 1983.

[3] M. Karjalainen, U. K. Laine, T. I. Laakso, and V. Välimäki, "Transmission-line modeling and real-time synthesis of string and wind instruments," in *Proc. Int. Computer Music Conf.*, Montreal, Canada, 1991, ICMA, pp. 293–296.

[4] G. Borin, G. De Poli, and A. Sarti, "Algorithms and structures for synthesis using physical models," *Computer Music Journal*, vol. 16, no. 4, pp. 30–42, 1992.

[5] J. O. Smith, "Physical modeling synthesis update," *Computer Music Journal*, vol. 20, no. 2, pp. 44–56, 1996.

[6] C. Cadoz, A. Luciani, and J.-L. Florens, "CORDIS-ANIMA: A Modeling and Simulation System for Sound Synthesis - The General Formalism," *Computer Music Journal*, vol. 17, no. 1, pp. 19–29, Spring 1993.

[7] G. De Poli and D. Rocchesso, "Physically-based sound modeling," *Organised Sound*, vol. 3, no. 1, 1998.

[8] G. Borin and G. De Poli, "A hammer-string interaction model for physical model synthesis," in *Proc. XI Colloquium Musical Informatics*, Bologna, Italy, Nov. 1995, AIMI, pp. 89–92.

[9] V. Välimäki, C. Erkut, and M. Laurson, "Sound synthesis of plucked string instruments using a commuted waveguide model," in *Proc. of the 17th International Congress on Acoustics (ICA)*, Rome, Italy, Sept. 2001, vol. 4, Available only on CD.

[10] W. Gaver, "How do we hear in the world? explorations in ecological acoustics," *Ecological Psychology*, vol. 5, no. 4, pp. 285–313, Apr. 1993.

[11] G. Kramer, *Auditory Display: Sonification, Audification, and Auditory Interfaces*, Addison-Wesley, Reading, MA, 1994.

[12] F. Avanzini and D. Rocchesso, "Controlling material properties in physical models of sounding objects," in *Proc. Int. Computer Music Conf.*, La Habana, Cuba, Sept. 2001, pp. 91–94.

[13] F. Avanzini and D. Rocchesso, "Modeling collision sounds: Non-linear contact force," in *Proc. Conf. on Digital Audio Effects (DAFX-01)*, Limerick, Ireland, Dec. 2001, pp. 61–66.

[14] M. Rath, D. Rocchesso, and F. Avanzini, "Physically based real-time modeling of contact sounds," in *Proc. Int. Computer Music Conf.*, Göteborg, Sept. 2002.

[15] P. R. Cook, *Real Sound Synthesis for Interactive Applications*, A. K. Peters, L.T.D., 2002.

[16] P. A. Houle and J. P. Sethna, "Acoustic emission from crumpling paper," *Physical Review E*, vol. 54, no. 1, pp. 278–283, July 1996.

[17] J. P. Sethna, K. A. Dahmen, and C. R. Myers, "Crackling noise," *Nature*, , no. 410, pp. 242–250, Mar. 2001.

[18] M. R. Schroeder, *Fractal, Chaos, Power Laws: Minutes from an Infinite Paradise*, W.H. Freeman & Company, New York, NY, 1991.

[19] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, New York, 2nd edition, 1984.

[20] S. K. Mitra, *Digital Signal Processing. A computer-Based Approach*, McGraw-Hill, New York, 1998.

[21] M. Puckette, "Pure data," in *Proc. Int. Computer Music Conf.*, Thessaloniki, Greece, Sept. 1997, ICMA, pp. 224–227.

[22] R. Dannenberg and I. Derenyi, " Combining Instrument and Performance Models for high-quality Music Synthesis," *Journal of New Music Research*, vol. 27, no. 3, 1998, pp. 211–238.

[23] R. Bresin and A. Friberg, "Emotional coloring of computer controlled music performance," *Computer Music Journal*, vol. 24, no. 4, 2000, pp. 44–62.

[24] A. Friberg and J. Sundberg, "Does music performance allude to locomotion? A model of final ritardandi derived from measurements of stopping runners," *J. Acoust. Soc. Amer.*, Vol. 105, No. 3, 1999, pp. 1469–1484.

[25] A. Friberg, J. Sundberg, and L. Frydén, "Music from Motion: Sound Level Envelopes of Tones Expressing Human Locomotion," *Journal of New Music Research*, Vol. 29, No. 3, 2000, pp. 199-210.

[26] R. Bresin, A. Friberg, and S. Dahl, "Toward a new model for sound control", in *Proceedings of the COST-G6 Conference on Digital Audio Effects - DAFx01*, Limerick, Ireland, 2001, pp. 45-49.

[27] A. Friberg, V. Colombo, L. Frydn, and J. Sundberg, "Generating Musical Performances with Director Musices", *Computer Music Journal*, vol. 24, no. 3, 2000, pp. 23-29.

[28] R. Bresin, A. Friberg, and J. Sundberg, "Director musices: The KTH performance rules system", in *Proceedings of SIGMUS-46*, Kyoto, 2002, pp. 43-48.

[29] P. N. Juslin, A. Friberg, and R. Bresin, "Toward a computational model of expression in performance: The GERM model," *Musicae Scientiae*, Special issue 2001-2002, 2002, pp. 63-122.