Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2016)
Ladislav Kavan and Chris Wojtan (Editors)

Real-time sound synthesis for paper material based on geometric analysis

Camille Schreck[1], Damien Rohmer[1,3], Doug James[2], Stefanie Hahmann[1] and Marie-Paule Cani[1]

[1]Univ. Grenoble Alpes & CNRS (LJK), Inria.
[2]Stanford University.
[3]CPE Lyon, Univ. lyon.

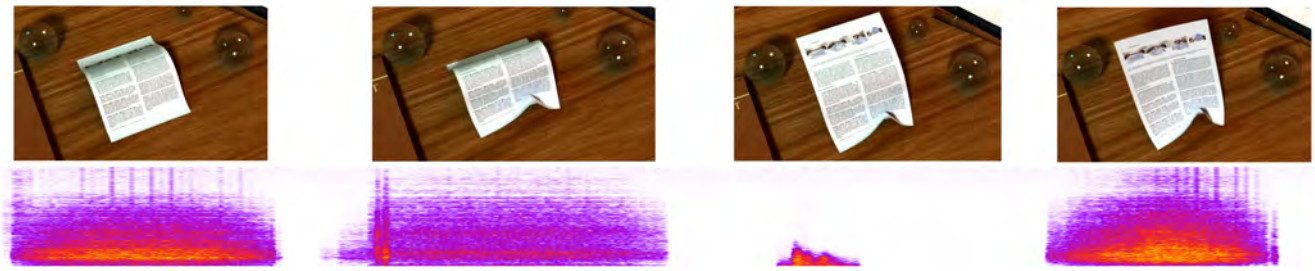


Figure 1: **Real-time paper sounds:** Our method can automatically synthesize a plausible, synchronized soundtrack (shown as a spectrogram) for interactive simulations of a rectangular sheet of paper. In this 3.3s animation, the back edge of the sheet is held while the front edge slides toward it, thereby curving the paper; then, when the back edge is released and the front edge is pinched, the back edge stands up.

Abstract

In this article, we present the first method to generate plausible sounds while animating crumpling virtual paper in real time. Our method handles shape-dependent friction and crumpling sounds which typically occur when manipulating or creasing paper by hand. Based on a run-time geometric analysis of the deforming surface, we identify resonating regions characterizing the sound being produced. Coupled to a fast analysis of the surrounding elements, the sound can be efficiently spatialized to take into account nearby wall or table reflectors. Finally, the sound is synthesized in real time using a pre-recorded database of frequency- and time-domain sound sources. Our synthesized sounds are evaluated by comparing them to recordings for a specific set of paper deformations.

Categories and Subject Descriptors (according to ACM CCS):
I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;
H.5.5 [Information Systems]: Information Interfaces and Presentation—Sound and Music Computing

1. Introduction

Thanks to the advances in the Computer Graphics field, it is now possible to create visually compelling virtual worlds. Although very common in daily life, the characteristic appearance and sound produced by a crumpled sheet of paper, or the universally recognized friction of dollar bills, are missing in virtual environments. Contrary to rigid objects whose shapes do not vary in time, or to cloth garments whose sounds are not strongly dependent on shape, the sound of paper depends heavily on its shape and also changes dramatically during crumpling phenomena. Therefore, in order to achieve real-time sound generation for interactive virtual paper models, a highly efficient analysis of the shape of paper, coupled to fast sound synthesis, is therefore required.

Although very familiar to humans, paper sound is a complex combination of different sound styles. It is a mix between continuous noisy sounds produced by frictional sliding, and discrete events produced by geometric bending and crumpling processes. These discrete sounds may also vary between long "*flap*" sounds when the sheet is still smooth, and more short "*clac*" sounds in more crumpled cases. Real-time synthesis of this variety of sounds, with strong dependence on the paper's shape and its environment, is therefore a challenging problem.

Our sound synthesis approach consists of four steps performed at run-time during the visual simulation of the sheet of paper. First, sound-producing events are detected by analyzing the paper surface animation. Second, each event is characterized using the geometry and the motion of the surface to parameterize our sound models. Third, the sound is synthesized using the estimated model parameters using computed characteristics and two pre-recorded databases of primitive sounds: one for the discrete crumpling sounds, and another for the continuous friction sounds. Fourth, the resulting sound is spatialized in 3D to account for the influence of nearby planar surfaces, such as a wall or table.

Our three main contributions are as follows. First, we develop a new shape-dependent model for friction and crumpling sounds. Our model is based on local surface regions we call *resonators*, which characterize the region where significant sound-related vibrations occur. Second, we propose a two-sided detection and characterization method for both friction and crumpling sounds, that incorporates their dependence on nearby free (unconstrained) boundaries of the sheet, leading to four different sound types: constrained and unconstrained crumpling sounds depending on the presence of a free border in the producing region; and similarly, constrained and unconstrained friction sounds. Third, we propose a new efficient approximation method for sound spatialization, adapted to model the sound of thin-sheet surfaces near flat surfaces, such as tables or walls.

2. Related Work

The generation of sound has attracted an increasing interest in computer animation over the last few years. Perceptual studies and some synthesis methods are gathered in [GA14].

Data-driven synthesis: In the traditional approach, still widely used in in motion pictures and video games, real sound effects are recorded and edited to match the visual display by Foley artists. The approach produces high quality results but is time consuming and cannot be used for interactive environments. The process can be automated for interactive applications by triggering by computer-generated events and rendering pre-recorded sound samples [TH92]. Unfortunately, this approach lacks of the fine-grained variability and synchronization needed to interactively render complex interactions occurring with paper manipulation.

Other data-driven methods are inspired by granular synthesis techniques [Roa04]. Dubnov *et al.* [DBJEY*02] and Bar-Joseph *et al.* [BJLW*99] propose to use wavelet trees to model sound texture and synthesize audio backgrounds. The "Sound-by-Numbers" method [CBBJR03] uses a granular synthesis model driven by a low-dimensional motion signal. Concatenative sound synthesis (CSS) [S*00] methods consist in selecting in an large database of sound units, the units that match the best the sound to be synthesized.

An *et al.* [AJM12] built on these methods to automatically synthesize the sound of a physically based cloth animation. First they analyze the deformation to find crumpling and friction events that drive the synthesis of a low-quality target signal. A CSS process is then used to select the best units from a database of recorded cloth sounds. Our method is inspired by the first part of this approach, and adapts it to paper material. In the case of paper, the sound is also produced by crumpling and frictional sliding mechanisms, but in addition the sound is highly dependent on the shape of the paper.

Physically based synthesis: Another approach for simulating sound is the physically based modeling sound techniques founded on the description of the physical phenomenon involved in the sound generation. In computer graphics, these techniques usually synchronize physically based sound synthesis to physically based animation. O'Brien *et al.* [OCE01] uses an FEM (Finite Element Method) to synthesize both the animation and the sound. A wide range of complex scenes (including sheet-like objects) can be simulated but the high computation time required prevents it from being efficient enough for interactive application.

Modal vibration models [PW89, JP02] have been introduced to digital sound generation by Adrien [Adr91]. The linear modal synthesis methods have become widely used for rigid-body sound [DP98, vdDKP01, Coo02, OSG02, JBP06, ZJ10]. Moreover it is possible to model a large number of sounding objects at interactive rates using acceleration techniques [RL06, BDT*08], and also memory compression [LAJJ14]. Ren *et al.* [RYL13] presents a method to estimate modal parameters, from recorded audio clips. The linear modal synthesis has been also considered for non-linear thin-shell sounds by Chadwick *et al.* [CAJ09], but such models only support small deformations.

Power-law models of the acoustic emissions from crumpling elastic sheets [KL96] and paper [HS96] have been studied by the physic community, and have led to geometry-independent stochastic sound models of crumpling [FB03]. Procedural stochastic methods have also been used to create a simple tearing sound model for paper [LFD*15], but this model cannot capture the rich sounds produced when crumpling paper.

3. Overview

Two kinds of sounds are usually identified for thin-sheet material:

1. **Friction sounds** caused by regions of the surface sliding along either another object of the scene or another part of the same surface. The sound is produced by the sliding part bumping randomly against small irregularities of the other surface and can be approximated by a colored noise (see Figure 3).

2. **Crumpling sounds** generated in regions where the bending direction changes suddenly, such as when the sign of surface curvature changes. The transition between two equilibrium states can produce a sudden burst of energy which is quickly dampened (see Figure 4).

Paper material is extremely rigid compared to other thin materials, such as fabric. Consequently, resonant vibrations can have a strong impact on the sound. The size and the shape of theses resonating regions influence the resulting sound. A good experiment to convince oneself of the importance of this phenomenon is to listen to the difference between the sounds produced while sliding a flat piece of paper over a table, compared to sliding a curved piece of paper enabling the sound to resonate (Figure 3).

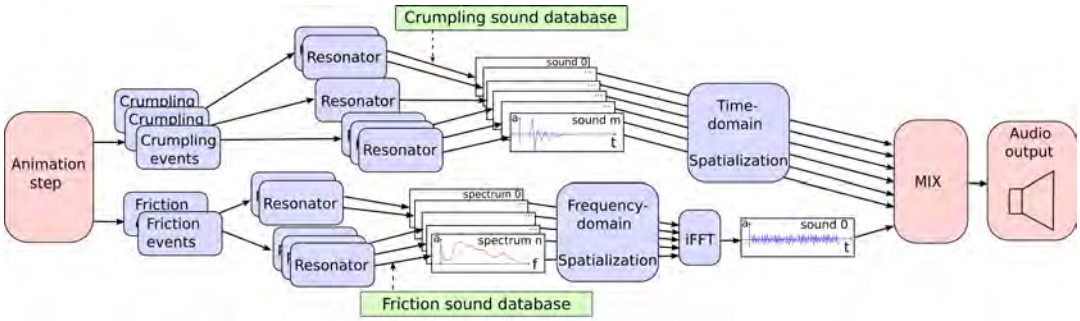


Figure 2: **Overview:** After computing the sound-producing events, each of them is associated with a set of *resonators*. Each of those *resonators* leads to a sound unit after querying it on a pre-computed sound data-base. Each sound unit is spatialized to adapt to our scene and the final resulting sound is obtained by mixing all the contributing sounds.

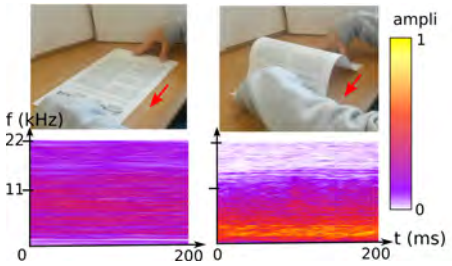


Figure 3: **Spectrogram of recorded friction sound** reveals slowly varying spectra in time, which is well approximated by a colored noise model. (left) Constrained friction sound obtained for a flat sheet of paper. (right) Unconstrained friction sound produced by a curved sheet.

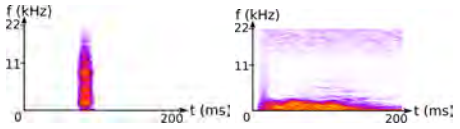


Figure 4: **Spectrogram of recorded crumpling sound:** (left) "clac" sound produced by a constrained resonating region. (right) "flap" sound produced by an unconstrained region.

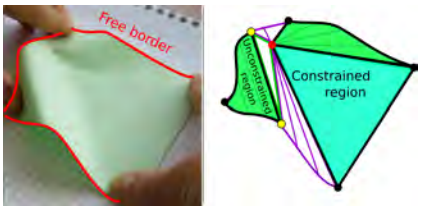


Figure 5: **Free borders** are natural edges of the paper that are curved enough to allow enough degrees of freedom to be able to oscillate (in red). A region of the surface is considered as constrained if it has no free border, and unconstrained if it contains at least one.

We call free borders natural edges of the paper that are curved enough to be able to vibrate (see Figure 5 (left)). As they have more degrees of freedom than interior edges or borders constrained to be flat, they have an notable influence on the sound. When the resonating region has no free border, we observe that the friction sound is close to a white noise (Figure 3 left) and is less loud than friction sounds produced by a bent surface containing therefore a free border (Figure 3 right). Similarly for the crumpling sounds, we register sharp "clac" sounds (Figure 4 left) – which you can hear while crumpling a paper into a ball – that are produced by constrained regions dampening the sound in a few milliseconds. On the opposite, when the vibrations can reach a free border a longer "flap" sound lasting few hundreds of milliseconds (Figure 4 right) – which you can hear while wiggling a sheet of paper held on one border – is produced. Based on these observations, we therefore make a difference between sounds produced by regions containing one or more free border(s), that we call unconstrained regions, and those produced by regions that contains none, i.e. constrained regions (see Figure 5 (right)).

Our key idea is to reproduce these resonance phenomena thanks to a procedural detection of representative regions in which the vibrations caused by a crumpling or friction event occur. This enables us produce, for each of these regions we call *resonators*, an appropriate shape-dependent sound.

In order to output sounds as rich as those of real paper in real time, we use pre-recorded sound units stored in Crumpling and Friction sound databases. Due to their different nature, we need the whole sound to replay a crumpling sound, while we just use the time-averaged magnitude spectrum to reproduce a friction sound (see §5). Therefore our Friction sound database only stores spectra.

Our processing pipeline is summarized in Figure 2. Given an animated mesh modeling the deformation of the paper surface through time, we extract two types of *sound source events* at each time step: (1) we compute the parts of the surface in contact with other objects, since each of these parts defines a *friction event*; and (2) we detect surface regions where the curvature changes sign, and use these features to identify *crumpling events* as in [AJM12]. For each sound source event, we then compute a set of *resonators*, i.e., the regions in which vibrations caused by this event occur (see §4). Resonators are then parameterized by their shape and size (see §5) and subdivided into two classes: *unconstrained* resonators or *constrained* resonators, depending on whether or not they have a free border that is able to oscillate. For each resonator, a sound is ex-

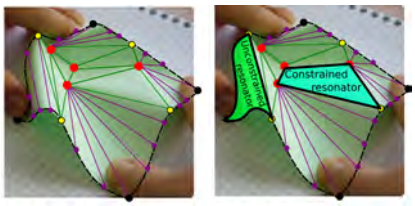


Figure 6: **Geometric structure of surface animation model:** The surface is approximated by a set of generalized cones (purple) and flat region (green). There are two kinds of singular points – interior singular points (red), where the fibers are broken and which are apices on cones, and border singular points (yellow) delimiting curved regions. Image courtesy of [Schreck et al. 2015].

tracted from the appropriate database depending on the type of resonator and the type of sound event. This leads to the four categories of sounds described above: *clac* and *flap* sounds for respectively constrained and unconstrained resonators in case of crumpling events, and different friction sounds for constrained and unconstrained resonators in case of friction events. A specific sound is extracted based on the geometric parameters of the resonator, and then reduced or amplified depending on the magnitude of the sound-source event. Finally, the assembled sound is spatially embedded within a 3D environment, by taking into account the relative positioning of the paper with respect to surrounding obstacles such as tables or walls, and the position of the listener (see §6).

4. Detecting resonators

Geometric Model: At each animation step, the first stage of our method consists in detecting the sound-source events on a 3D virtual paper surface provided by an external animation or simulation engine. As we aim at providing sound synthesis in real-time, choosing an interactive animation system for paper crumpling is mandatory. In this work, we use the model proposed by Schreck *et al.* [SRH*15], which is the only existing model capable of animating paper crumpling at interactive rates. It has the advantage of being built on high level geometric descriptors, which we can reuse. In particular, the approach combines coarse physical simulation with developable geometry processing steps. The geometry generation part is based on the hypothesis that the creased curves, which appear where microscopic fiber structure has been broken causing the surface not to be C^2 anymore, can be approximated by singular points and thus the developable surface of the paper can be approximated by a set of generalized cones. Although our system could work on top of other thin-sheet animation methods, this approach has the advantage that it directly provides a partition R_{paper} of the sheet into developable regions defined by singular points (see Figure 6). We can distinguish between singular points respectively located on the border (yellow dots), inside the surface where fibers are broken (red dots), and anchored points (black dots) where the user's fingers guide the paper deformation. The partition R_{paper} is composed of:

1. **Curved regions** (in purple in Figure 6) that segment sections of generalized cones, and are represented by the rulings of these cones (purple lines in Figure 6). Each ruling is defined by two points, with at least one of them belonging to the border of the paper.

2. **Flat regions** (in green in Figure 6) that are represented by a set of coarse triangles whose vertices are either singular points or anchored points. The dihedral angle between every pair of adjacent triangles of a same flat region is inferior to a threshold, ε_a.

We use this surface partition to help represent surface resonators by considering each region of R_{paper} as a possible resonator. As curved regions have necessarily a curved border they are considered in the following as *unconstrained regions*, whereas flat regions are considered as *constrained regions* (see Figure 6 (left)).

Event Detection: We detect the sound producing events using a heuristic similar to the one proposed in An *et al* [AJM12]. Friction events are associated with vertices in contact with an obstacle (based on proximity computations), that are spatially connected. Each connected component represents the locus of a friction event. The mean speed of its vertices represent the amplitude of the event. The location and time of crumpling events are estimated by analyzing the mean curvature over the surface; crumpling events are represented by connected components of vertices whose mean curvature has undergone the same change of sign between the current and the previous animation step. The associated amplitude is computed as the sum of the curvature changes for all the vertices contributing to the event [AJM12].

As already stressed, contrary to cloth material where crumpling and vibrations only occur locally within the material, paper is very stiff (because of strong isometry constraints), and can support spatially coherent vibrations and buckling events over large regions. The sound produced by friction and crumpling events is thus much more dependent on the current geometrical state of the surface than it is for softer materials, such as cloth. Therefore, instead of directly associating a sound to each event as in [AJM12], we first associate a set of geometrically defined resonators to each event, and then synthesize the sound accordingly. While this is a crude approximation of the complex underlying phenomena, it enables the sound to have geometric dependence which can improve realism.

Resonator Estimation: In order to compute the set of resonators for an event, defined as a collection of connected surface pieces, we need to identify the boundary of the surface region that the event causes to vibrate. We choose to define this boundary in an intuitive way as follows. The borders of the sheet naturally belong to this boundary. Moreover, paper material tends to get very stiff in the direction of highly curved folds, or "ridges" (which are aligned to some of the rulings; see Figure 6), but the sheet is more flexible in the main curvature direction, Consequently, transverse vibrations tend to be trapped in these interior resonator regions by ridges [GWV02]. Therefore, we assume that vibrations do not propagate across ridges, which we define as a ruling whose dihedral angle is above a threshold, $a_{\max}$; we use $a_{\max} = 0.5$rad in our implementation. This is done by stamping as boundaries of the propagation region all the edges with high dihedral angle, in addition to free borders and to edges with an anchored point (position

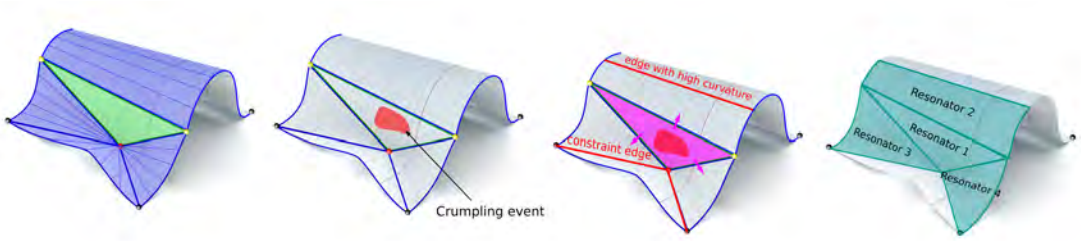


Figure 7: **Resonator estimation** (for a crumpling event): (left) curved and flat regions, (middle-left) detecting sound event, (middle-right) finding region of event and recursively finding the regions in which the vibrations can propagate, (right) resonators.

of virtual fingers manipulating the sheet) or in contact with another object, since this also prevents further propagation.

Defining these limits enables us to estimate the set of resonators where a sound event propagates as follows (see Figure 7): We initialize the set with the region(s) of R_{paper} where the event took place, and then recursively collect the neighboring regions until reaching an edge stamped as propagation border. Note that the edge that stops propagation may be located within a conical region of the shape. In this case, we only add the relevant sub-part of this region to the set of resonators. Each of the regions of R_{paper} collected is considered as a resonator and will produce a sound as explained in the next section.

5. Geometry-based sound synthesis

Resonator Parameters: As already stated, our method to produce rich and natural sounds in real time is to re-play pre-recorded sound units, selected from a database. We therefore need to parameterize resonators in order to match them to a specific sound unit, easy to query in a database, and we need to register sounds in the database accordingly. The number of parameters needs to be kept small in order to keep the database to a reasonable size. They should also discriminate well enough the different possible sounds.

In this work, we propose to parameterize each resonator by the following two variables: l_c the sum of lengths of the all free curved borders (red in Figure 8), and l_r the mean length of the rulings (blue in Figure 8). As already discussed, free borders of the surface are of major importance to characterize the sound. In selecting their length, we aim at extracting the parameters of the dominant sound. By associating it with the mean length of the rulings, we sample uncorrelated variables related to the area of the resonator. We therefore claim that the couple (l_r, l_c) is a good choice of parameters to correlate significantly a geometrical measurement of a region to the sound it produces. Note that we further validated this choice by performing real measurements, as described in §7.1.

Sound Databases: Let us detail how we pre-record the databases of sound units to be played at run time. Firstly, such databases are specific to a given type of paper material and to a given type of obstacle for the friction sound. In our experiments we used printer paper (80g/m^2) and a wooden table. We also recorded databases for tracing paper and paper of a bank note.

We actually created two different databases, respectively for *friction sounds* and for *crumpling sounds*. As friction sounds can easily be continuously synthesized as noise, without taking into account their temporal phase, we only store the magintude spectrum in the frequency domain of the sounds recorded in the sound database. In reverse, *crumpling sounds* have a specific time duration with a specific beginning and end, and are therefore stored in the temporal domain.

For these two databases, we sampled the (l_r, l_c) space, and for each value, we cut a rectangular sheet of paper with $(l_r, l_c/2)$ edge length, bent its edge of length $l_c/2$, recorded the sound produced by its friction, and by a change of sign of curvature (for the crumpling database). The *unconstrained friction* sound and the *flap* sounds are related to non-zero free border lengths and are the most dependent on the shape of the surface. The two other constrained types of sound, namely *constrained friction* and *clac* do not depend much on the shape of the surface and were therefore considered as special cases. To ease the recording process, we stored these two specific sounds in the $(l_r = 0, l_c = 0)$ entry of their respective database. Several sound samples were stored per entry to increase variety.

Sound Synthesis: Given the databases and the resonator, sounds are synthesized at run-time in the following way: For each resonator, we select the closest recorded (l_r, l_c) parameter in the relevant database, considering also the special case $(0,0)$ when the resonator does not have any free border, and randomly select one of the corresponding sound samples. Finally, all sounds are assembled together. Friction sounds are synthesized using the inverse Fast Fourier Transform method [RDRD92, MAKMV10]: friction spectra are summed up in the frequency domain. We add random phase to the resulting magnitude spectrum by multiplying each value of the spectrum vector by the corresponding value of a vector of complex numbers $\{c_i\} = \{e^{j\,\omega_i}\}$ where ω_i is a random angle chosen uniformely in the range $[-\pi, \pi]$. A buffer of the global friction sound is then obtain by applying an iFFT to the resulting complex vector. In order to smooth the transition between them, two consecutive buffers are overlapped. Moreover the amplitude of the sound is linearly scaled with respect to the mean speed of the associated relative frictional motion. The *crumpling sounds* are played until the end of the recorded unit. Their amplitude are also scaled linearly with respect to the magnitude of the change of curvature related to the event.

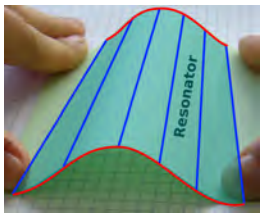


Figure 8: **Resonator parameterization** using the mean length of its (blue) rulings (l_r) and the sum of the lengths of the (red) free borders (l_c).

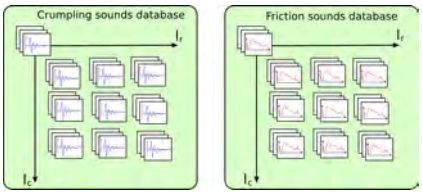


Figure 9: **Sound Databases:** Ordered in a matrix form according to the dimensions l_r and l_c. The crumpling database contains short sound clips, whereas the friction database contains sound spectra.

6. Spatialization

In this section, we describe how we embed the synthesized sound in the 3D space, by taking into account both the listener's position and sound reflection on the surrounding table or walls. Such planar surfaces interact with the sound vibrations of paper material which can be modeled as a dipole.

The dipole-like radiation behavior of paper surfaces leads to significant frequency-related interference effects above planar surfaces. For example when sliding a sheet of paper on a table to bend it more, as in Figure 1 (left), one may notice a pitch-shift. This is partly caused by the fact that the friction sound is reflecting on the table and the reflected source interferes with the original source. A pitch-shift is also noticeable when one moves a sheet of paper closer to a wall while wiggling it. As the paper gets closer to the wall, the "flap" sounds change. Note that in this work, we do not consider the interference caused by self-reflections on the surface by cavities formed by the paper.

We develop below a simplified model for these phenomena for both time-domain (crumpling) and frequency-domain (friction) sounds. Transverse vibrations of the paper sheet give rise to positive and negative pressure fluctuations on either side of the sheet. If we approximate each resonator a small vibrating disk sound sources, then we can reason about each disk's sound emission and any planar reflections.

Let us consider the sound field due to a vibrating rigid disk of radius a and zero thickness. If we assume that the normal velocity is $U(t)$, and that the disk is acoustically compact, i.e., the important wavelengths associated with the frequencies in $U(t)$ are much larger than a, then one can derive the acoustic pressure field p generated by the motion (see [How02] page 81, problem 9) as:

$$p(\mathbf{R},t) \approx \frac{2\rho}{3\pi c} a^3 \frac{\cos\theta}{R} \frac{\partial^2 U}{\partial t^2}(t - \frac{R}{c}), \tag{1}$$

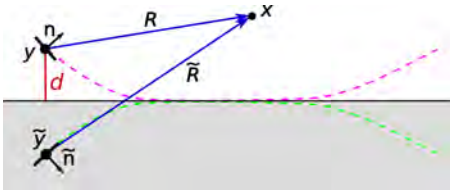


Figure 10: **Scattering:** Geometric configuration of a vibrating paper patch above a planar surface, along with its image source.

where the listener is located at $\mathbf{x}$, the center of the disk is located at $\mathbf{y}$, $\mathbf{n}$ is the unit normal vector, and c is the speed of sound in the air. The relative location is given by:

$$\mathbf{R} = \mathbf{x} - \mathbf{y}\ ,\ \ R = ||\mathbf{R}||\ ,\ \ \hat{\mathbf{R}} = \mathbf{R}/R\ ,\ \ cos\theta = \hat{\mathbf{R}} \cdot \mathbf{n} \tag{2}$$

as shown on Figure 10. This vibrating disk produces a *dipole* pressure field, with characteristic $1/R$ distance scaling, and a cosine lobe about direction $\mathbf{n}$. Without loss of generality, we can discard the scalar factor $\frac{2\rho}{3\pi c}$ (which is common to all our sound sources) and use:

$$p(\mathbf{R},t) \approx a^3 \frac{\cos\theta}{R} A(t - \frac{R}{c}), \tag{3}$$

where $A = \frac{\partial^2 U}{\partial t^2}$ is the acceleration of the patch along n.

Consider the dipole source from (3) at height d above an infinite rigid plane, as shown in Figure 10. Since the rigid plane imposes a zero boundary condition $\frac{\partial p}{\partial n} = 0$ on the normal acoustic particle velocity on the surface, a fictitious image dipole source $\widetilde{p}$ is introduced in the sound field. The global pressure field P is then the sum of the pressure fields of the dipole source and its image source:

$$P(\mathbf{x},t) = p(\mathbf{x},t) + \widetilde{p}(\mathbf{x},t). \tag{4}$$

The image source has reflected position $\widetilde{y}$ and normal $\widetilde{n}$, and corresponding relative coordinates $\widetilde{R}$ and $cos\widetilde{\theta} = \widetilde{R} \cdot \widetilde{n}$. Using this simplified model, we approximate sound reflections for crumpling and friction sounds using time- and frequency-domain reflection models, respectively, as follows.

Time-domain crumpling sounds: we scale each current time-domain crumpling sound pressure event $A(t)$ selected from the database to obtain its dipole-spatialized version and generate the following delayed sound source for the image dipole:

$$P(\mathbf{x},t) = \left(a^3 \frac{\cos\theta}{R}\right) A(t) + \left(a^3 \frac{\cos\widetilde{\theta}}{\widetilde{R}}\right) A(t - \frac{\widetilde{R} - R}{c}). \tag{5}$$

This change is done to each crumpling sound before synthesis.

Frequency-domain friction sounds: we obtain the dipole-spatialized version of each friction sound by placing Equation (5) in the frequency-domain (effectively replacing $A(t)$ by $A(\omega)e^{-i\omega t}$):

$$P(\mathbf{x},\omega) = \left(a^3 \frac{\cos\theta}{R} e^{i\omega\frac{R}{c}} + a^3 \frac{\cos\widetilde{\theta}}{\widetilde{R}} e^{i\omega\frac{\widetilde{R}}{c}}\right) A(\omega). \tag{6}$$

This multiplication factor is applied to the friction spectra $A(\omega)$ prior to region-based summation and synthesis of the final sound.

7. Results

Please see the accompanying video for the results described in this section, and listen to the video using good headphones in a quiet environment.

Implementation Details: In our results, we use sounds recorded at a sampling rate of 44100 Hz and play the generated sounds at the same rate. The spectra of the friction sounds contains 2048 frequencies which means that a buffer contains 2048 samples ($\approx$ 50 milliseconds) and we use an overlap of 1024 samples between two buffers. Our implementation makes use of the C++ library STK [CS99, SC05] to read and play sounds.

Recording of the database: Our two databases contain samples of sizes l_c and l_r, using the values $\{2.5, 5, 7.5, 10, 12.5, 15, 20, 25, 30\}$ cm. We recorded three crumpling sounds and one friction sound for each sheet size. The sounds were recorded manually under the same conditions. The microphone was placed one meter away from the sound source. For the friction sounds, the speed was measured by the time needed to go over the friction distance (2s for 20cm). The shape was retained by keeping the same amount of compression of the curved edges (30%). We recorded the unconstrained ("flap") sounds by holding both ends of a sheet (slightly curved by compression curved edges of 10 %) of the required size and twisting them to change the sign of the curvature. The constraint ("clac") sounds were isolated in a recording of crumpling paper.

7.1. Validation of the resonators' parameters

We used two parameters (l_c, l_r) to characterize the sound of a given resonator. To validate this choice in the case of the friction sound, we show that modifying these two parameters has a larger influence on the sound produced by friction than modifying other geometrical shape features with constant (l_c, l_r).

We performed the following experiment: let us consider a reference square paper sheet of size $10cm \times 10cm$. On one hand, we prepared rectangular paper sheets with smaller or longer edge length. Each rectangular sheet was then bent in the configuration shown in Figure 11 ((b) and (c)) such that their edge length respectively corresponds to $l_c/2$ and l_r. On the other hand, we used the reference paper sheet - fixing therefore (l_c, l_r) - and modified only its shape by compressing the two extremities of the sheet as illustrated in Figure 11 (a). In all these scenarios we recorded the sound produced by the friction of the sheet of paper on the same table, at same speed. The relative variation of the parameter is in the range of 50 to 75%.

To compare the different recordings, we computed the histogram corresponding to the magnitude spectrum of each sound and then compared them using the quadratic-form distance [PW10]. Table 1 gathers our results. The first four columns show that modifying the l_c or l_r parameters respectively by a factor of 50 to 75% leads to a modification of their spectrum, quantified between 0.07 and 0.16. The next two columns show that keeping constant l_c and l_r parameters but changing the curvature of the sheet leads to a change of spectrum quantified between 0.03 to 0.05, which stays always below the influence of tuning l_c or l_r.

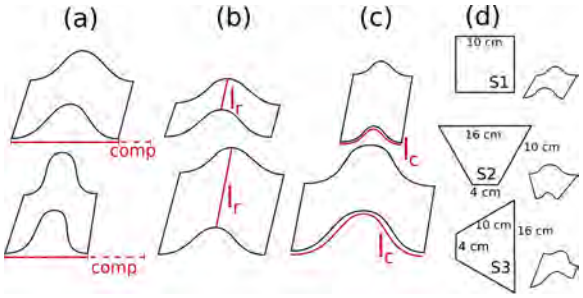


Figure 11: **Validation experiment:** We compare sounds obtained while varying three parameters: amount of compression (a), l_r (b) and l_c (c). We also measured the sounds obtained for 3 sheet shapes with the same compression, l_r and l_c: S1 is a square 10 cm x 10cm (d top); S2 is a trapezoid whose bases, of respective sizes 4cm and 16cm, are the curved borders (d middle); S3 is a trapezoid whose bases of sizes 4cm and 16cm, are the flat borders (d bottom).

To further study the influence of shape, we also compared the sound produced by three different sheets of shape S_1, S_2 and S_3, shown in Figure 11 (d). Each piece of paper is compressed by a factor of 30% and the results on their respective histogram distances computed using the same approach as above are gathered in the last two columns of Table 1. We obtained distances comparable to those obtained for change of curvature, and once again lower than those obtained for changes of l_r and l_c.

l_r	d_r
5 vs. 10	0.10
10 vs. 20	0.07
5 vs. 20	0.16

$l_c/2$	d_c
5 vs. 10	0.04
10 vs. 20	0.12
5 vs. 20	0.13

comp	d_{comp}
20 vs. 40	0.03
40 vs. 80	0.05
20 vs. 80	0.03

s	d_s
s1 vs. s2	0.04
s1 vs. s3	0.01
s2 vs. s3	0.03

Table 1: **Distances between friction spectra** of rectangular sheets of paper while varying the parameters as shown in Figure 11. l_r and $l_c/2$ are the considered edge length parameters in cm, d_r and d_c are the respective histograms distances. *comp* is the relative compression ratio, d_{comp} the associated histogram distance. d_s are the distances between friction spectra of the shapes s described in Figure 11(d) compressed of 30%. s refers to the shape pattern show in Figure 11 (right), and d_s corresponds to the histograms distances. For example, the first line of the first table indicates that the histogram distance is 0.10 between the friction magnitude spectrum produced by a resonator with $l_r = 5\,\text{cm}$ and the one produced by a resonator with $l_r = 10\,\text{cm}$.

7.2. Comparison with real paper material

To validate our resonator model, we compared the results obtained using our method with sounds produced by real sheets of paper

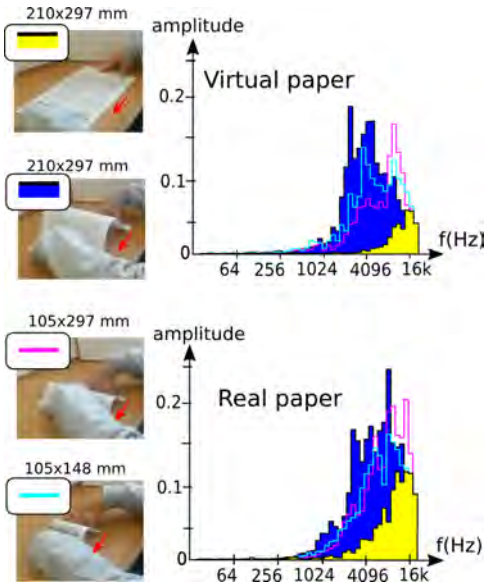


Figure 12: **Real-virtual comparison of friction sound spectra** for different sheet sizes.

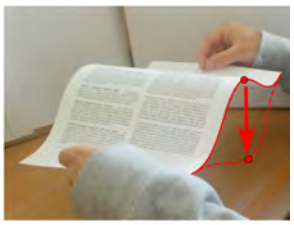


Figure 13: **Real-virtual comparison of "flap" sounds** for different sheet sizes, obtained by changing the curvature of bent paper.

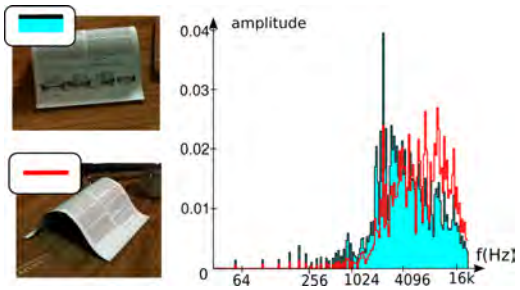


Figure 14: **Spatialization:** The video shows the result of spatialization for friction sound from two different points of view.

of different sizes. First, we focused on friction sounds produced by sheets of paper slid over a table. We used rectangular sheets of different sizes –210x297 millimeters (A4 format), 210x148 (half A4),105x297 (A5), 105x148 (A6)– and folded them as shown in Figure 12 (left). We also recorded the sound of an A4-sized flat sheet. We compared the resulting five spectra of virtual and real sounds in log scaled space (respectively top-right and bottom-right of Figure 12). The results show that our model follows a similar evolution of the magnitude spectrum with respect to the sheet size, i.e. a general translation of the spectrum towards higher frequencies when the size decreases. This evolution –and particularly the clear difference between flat paper and folded paper being slid along the same table– can also be heard in the video. Note that our synthesized sound is still less rich in terms of frequencies variations than the real one, in particular we do not reproduce phenomena such as the stick-slip sound that can be heard when real paper begins to slide. Some more advanced interactions can also have an influence – see for instance the case of musical instruments like violon [PW98].

Next, we compared real and virtual crumpling *flap* sounds for different sizes of sheets. Considering the same sizes than in the previous experiment, we folded each sheet of paper as shown in Figure 13 and revert the sheet while recording its sound. The attached video plays the resulting sounds. In both (real and virtual) cases, we get distinctive sounds depending on the size of the sheet.

7.3. Spatialization

Here, we conducted two experiments. Figure 14 (left and middle) and the associated video present the result of our frequency-domain spatialization method for the friction sound. The friction sound of our virtual paper is recorded for an A4 sheet compressed by a factor of 30% first and 10% then, leading to a change of height of the sound source. One can hear in the video that the friction sound depends on the height of the sound source, as it is the case for real paper sound. We can also hear the slight pitch shift as the sheet is bent. We also show in the video the modification of sound depending on the position of the listener in front and on the side of the sheet of paper (Figure 14). Figure 15 corresponds to a sheet of paper being wiggled while changing its distance with respect to the wall. The time-domain spatialization of the crumpling sounds causes the sound shift we can hear in the video.

7.4. Computational Performance

Memory requirements: The crumpling database contains 249 sound units whose lengths are going from 200 to 800 milliseconds. The friction database contains 63 spectra sampling 2048 frequencies –between 0 and 22kHz. The memory footprint of the database is around 10MB.

Computation times: The detection of the events and the computation of their resonators are done once per frame. Table 2 compares the mean time per frame for these audio operations to the computation time of the animation. We computed the sound of the animation represented Figure 18 (right) with three different resolutions. The computation time of the curvature depends on the resolution, so real-time performances get harder to reach when the resolution increases. Still, the time required for the audio-related operations always remains negligible compared to the time required for the animation.
Playing sounds is done using different threads than computing the animation To get real time, it is necessary to generate at least 44100 audio samples per second. A thread is created for each crumpling sound. We only applied scaling operations to the recorded samples before sending them to the audio output. A single thread is used for

Figure 15: **Spatialization:** The sound may be modified by the proximity of a wall. It is noticeable for example when one wriggles a sheet of paper near a wall. A change a pitch can be heard while moving the sheet away from the wall. The accompanying video shows similar effect obtains thanks to the spatialization step.

friction sounds. A new buffer is computed every 1024 samples using the weighted sum of the spectra of all the friction sounds. Then the actual sound is computed by applying an Inverse Fast Fourier Transform. This computation has to be done 44 times per second and took 4ms in average in our examples.

examples (Fig. number)	animation (ms per frame)	audio (ms per frame)
1	168	5
17	182	6
18 (left) (183 triangles)	252	6
18 (left) (446 triangles)	1322	17
18 (left) (1333 triangles)	11860	75

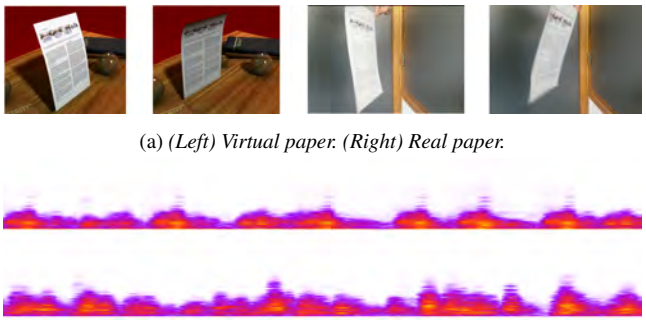
Table 2: **Mean computation time** for the animation and mean computation time for the audio operations, done once per simulation step

7.5. Others results

In this section, we present some more complex examples, which are also present in the attached video. Figures 1 and 16 show, respectively in the virtual case and in the real case, the spectrogram of the sound generated by while sliding one edge of a sheet along the table thereby curving the paper, then pinching the front border causing the back to stand up. This example mixes all the different categories of sounds we considered. We note that the friction sounds are much more regular than the synthetic ones. This is mainly due to the fact the it very difficult to reproduce a smooth movement "by hand." Nonetheless, we obtain the same type of event at the same moment, notably the "flap" sound when the sheet stands up and different kinds of friction sound for the different shapes that the sheet takes. Figure 17 shows a sheet (real and virtual) held by one border and wiggled and the corresponding spectrograms. Figure 18 (left) shows the example of a sheet being crumpled. We also simulated the sound of different categories of paper material by recording different databases, such as for tracing paper and paper of a bank note (Figure 18 (middle) and (right)).

8. Discussion and conclusion

We presented the first method that automatically generates plausible sounds fitting an interactive animation of paper material. We



(a) *(Left) Virtual paper. (Right) Real paper.*

(b) *Spectrogram obtained for (Top) virtual paper, and (Bottom) real paper. Both examples show 1.5s of the sound*

Figure 17: **An A4 sheet of paper** held by two corners is lightly shaken.

Figure 18: **Different paper materials** being crumpled can be synthesized in recording different databases: (Left) printer paper, (Middle) tracing paper, (Right) a bank note.

achieved real-time sound synthesis through a trade-off between speed and accuracy, leading to several noticeable limitations. First, the range of possible sounds is limited when using pre-recorded sound databases. Moreover, each database is associated with a specific type of paper and obstacle materials (for friction sounds), and changing them requires recording a new database.

Another limitation is due to the small set of physical parameters taken into account during the synthesis process. By requiring only animated mesh sequences, our method can handle a large variety of animation inputs. However, since it does not have access to physical contact force information, it can not, for instance, include dependence of friction sound volume on contact force. Moreover, our resonator model approximates paper vibrations in a resolution-dependent manner, and does not actually model vibrations directly. Challenging but familiar scenarios, such as crumpling paper into a ball, remain outside the scope of this work, since they would require detailed simulation of self contact, as well as more sophisticated modeling of sound inter-reflections and scattering.

The visual simulation of paper is also a bottleneck, and currently there are no methods that are both interactive and able to model very crumpled paper, especially with self contact. We used an interactive method to demonstrate our fast sound synthesis, and also because it enabled efficient estimation of resonators. Also although we did not obtain very crumpled paper, our results depict a large range of scenarios.

Finally, in future work, we would also like to take into account the specific properties of developable surfaces to allow for more accurate vibration modeling without sacrificing computational efficiency.

Figure 16: **Comparison:** Spectrogram obtained while reproducing the experiment shown in Figure 1 with real paper.

Acknowledgements: This research was partially funded by the ERC advanced grant no. 291184 EXPRESSIVE.

References

[Adr91] ADRIEN J.-M.: The missing link: Modal synthesis. In *Representations of Musical Signals*, De Poli G., Piccialli A., Roads C., (Eds.). MIT Press, Cambridge, MA, USA, 1991, pp. 269–298. 2

[AJM12] AN S. S., JAMES D. L., MARSCHNER S.: Motion-driven concatenative synthesis of cloth sounds. *ACM Transactions on Graphics (Proc. SIGGRAPH)* (2012). 2, 3, 4

[BDT*08] BONNEEL N., DRETTAKIS G., TSINGOS N., VIAUD-DELMON I. L., JAMES D.: Fast modal sounds with scalable frequency-domain synthesis. *ACM Transactions on Graphics (Proc. SIGGRAPH) 27*, 3 (2008). 2

[BJLW*99] BAR-JOSEPH Z., LISCHINSKI D., WERMAN M., DUBNOV S., ELYANIV R.: Granular synthesis of sound textures using statistical learning. In *Proceedings of the International Computer Music Conference* (1999), pp. 178–181. 2

[CAJ09] CHADWICK J. N., AN S. S., JAMES D. L.: Harmonic shells: A practical nonlinear sound model for near-rigid thin shells. *ACM Transactions on Graphics 28*, 5 (2009), 119:1–119:10. 2

[CBBJR03] CARDLE M., BROOKS S., BAR-JOSEPH Z., ROBINSON P.: Sound-by-numbers: Motion-driven sound synthesis. *Proc. Symposium on Computer Animation* (2003), 349–356. 2

[Coo02] COOK P. R.: *Real Sound Synthesis for Interactive Applications*. A. K. Peters, Ltd., Natick, MA, USA, 2002. 2

[CS99] COOK P. R., SCAVONE G.: The synthesis toolkit (stk). *Proc. International Computer Music Conference* (1999), 164–166. 7

[DBJEY*02] DUBNOV S., BAR-JOSEPH Z., EL-YANIV R., LISCHINSKI D., WERMAN M.: Synthesizing sound textures through wavelet tree learning. *IEEE Computer Graphics and Applications 22*, 4 (2002), 38–48. 2

[DP98] DOEL K. V. D., PAI D. K.: The sounds of physical shapes. *Presence: Teleoper. Virtual Environ. 7*, 4 (1998), 382–395. 2

[FB03] FONTANA F., BRESIN R.: Physics-based sound synthesis and control: crushing, walking and running by crumpling sounds. In *Proc. Colloquium on Musical Informatics* (2003), pp. 109–114. 2

[GA14] GIORDANO B. L., AVANZINI F.: Perception and synthesis of sound-generating materials. In *Multisensory Softness*. Springer, 2014, pp. 49–84. 2

[GWV02] GOPINATHAN A., WITTEN T. A., VENKATARAMANI S. C.: Trapping of vibrational energy in crumpled sheets. *Phys. Rev. E 65* (2002). 4

[How02] HOWE M. S.: *Theory of Vortex Sound*. Cambridge University Press, 2002. Cambridge Books Online. 6

[HS96] HOULE P. A., SETHNA J. P.: Acoustic emission from crumpling paper. *Physical Review E 54*, 1 (1996), 278. 2

[JBP06] JAMES D. L., BARBIC J., PAI D. K.: Precomputed acoustic transfer: Output-sensitive, accurate sound generation for geometrically complex vibration sources. *ACM Transactions on Graphics (Proc. SIGGRAPH)* (2006), 987–995. 2

[JP02] JAMES D. L., PAI D. K.: Dyrt: dynamic response textures for real time deformation simulation with graphics hardware. *ACM Transactions on Graphics 21*, 3 (2002), 582–585. 2

[KL96] KRAMER E. M., LOBKOVSKY A. E.: Universal power law in the noise from a crumpled elastic sheet. *Physical Review E 53*, 2 (1996). 2

[LAJJ14] LANGLOIS T. R., AN S. S., JIN K. K., JAMES D. L.: Eigenmode compression for modal sound models. *ACM Transactions on Graphics 33*, 4 (2014). 2

[LFD*15] LEJEMBLE T., FONDEVILLA A., DURIN N., BLANC-BEYNE T., SCHRECK C., MANTEAUX P.-L., KRY P. G., CANI M.-P.: Interactive procedural simulation of paper tearing with sound. In *Motion In Games (MIG)* (2015). 2

[MAKMV10] MARELLI D., ARAMAKI M., KRONLAND-MARTINET R., VERRON C.: Time-frequency synthesis of noisy sounds with narrow spectral components. *IEEE Transactions on Audio, Speech and Language Processing 18*, 8 (2010), 1929–1940. 5

[OCE01] O'BRIEN J. F., COOK P. R., ESSL G.: Synthesizing sounds from physically based motion. *Proc. of ACM SIGGRAPH* (2001), 529–536. 2

[OSG02] O'BRIEN J. F., SHEN C., GATCHALIAN C. M.: Synthesizing sounds from rigid-body simulations. *Proc. of ACM SIGGRAPH* (July 2002), 175–181. 2

[PW89] PENTLAND A., WILLIAMS J.: Good vibrations: Modal dynamics for graphics and animation. *SIGGRAPH Comput. Graph. 23*, 3 (July 1989), 207–214. 2

[PW98] PITTEROFF R., WOODHOUSE J.: Mechanics of the contact area between a violin bow and a string. part i: Reflection and transmission behaviour. *Acta Acustica united with Acustica 84*, 3 (1998), 543–562. 8

[PW10] PELE O., WERMAN M.: The quadratic-chi histogram distance family. In *Computer Vision - ECCV*, Daniilidis K., Maragos P., Paragios N., (Eds.), vol. 6312 of *Lecture Notes in Computer Science*. Springer, 2010, pp. 749–762. 7

[RDRD92] RODET X., DEPALL P., RODET X., DEPALLE P.: Spectral envelopes and inverse fft synthesis. In *Proc. of Audio Engineering Society Convention* (1992). 5

[RL06] RAGHUVANSHI N., LIN M. C.: Interactive sound synthesis for large scale environments. In *Proc. Symp. on Interactive 3D Graphics and Games* (2006), ACM, pp. 101–108. 2

[Roa04] ROADS C.: *Microsound*. The MIT Press, 2004. 2

[RYL13] REN Z., YEH H., LIN M. C.: Example-guided physically based modal sound synthesis. *ACM Transactions on Graphics 32*, 1 (2013), 1–16. 2

[S*00] SCHWARZ D., ET AL.: A system for data-driven concatenative sound synthesis. In *Digital Audio Effects (DAFx)* (2000), pp. 97–102. 2

[SC05] SCAVONE G., COOK P. R.: Rtmidi, rtaudio, and a synthesis toolkit (stk) update. *Int. Computer Music Conference* (2005). 7

[SRH*15] SCHRECK C., ROHMER D., HAHMANN S., CANI M.-P., JIN S., WANG C. C., BLOCH J.-F.: Non-smooth developable geometry for interactively animating paper crumpling. *ACM Transactions on Graphics 35*, 1 (2015). 4

[TH92] TAKALA T., HAHN J.: Sound rendering. *Proc. SIGGRAPH 92, ACM Computer Graphics* (1992), 211–220. 2

[vdDKP01] VAN DEN DOEL K., KRY P. G., PAI D. K.: Foleyautomatic: Physically-based sound effects for interactive simulation and animation. *Proc. ACM SIGGRAPH* (2001), 537–544. 2

[ZJ10] ZHENG C., JAMES D. L.: Rigid-body fracture sound with precomputed soundbanks. *ACM Transactions on Graphics (Proc. SIGGRAPH) 29*, 3 (2010). 2