

# Crumpling Sound Synthesis

Gabriel Cirio<sup>1,2</sup>

Dingzeyu Li<sup>2</sup>

Eitan Grinspun<sup>2</sup>

Miguel A. Otaduy<sup>1</sup>

Changxi Zheng<sup>2</sup>

<sup>1</sup>URJC Madrid

<sup>2</sup>Columbia University



**Figure 1:** *Garbage in, audio out: given an animation of a buckling thin shell, our method uses substructured modal analysis and stochastic enrichment to synthesize accompanying foley.*

## Abstract

Crumpling a thin sheet produces a characteristic sound, comprised of distinct clicking sounds corresponding to buckling events. We propose a physically based algorithm that automatically synthesizes crumpling sounds for a given thin shell animation. The resulting sound is a superposition of individually synthesized clicking sounds corresponding to visually significant and insignificant buckling events. We identify visually significant buckling events on the dynamically evolving thin surface mesh, and instantiate visually insignificant buckling events via a stochastic model that seeks to mimic the power-law distribution of buckling energies observed in many materials.

In either case, the synthesis of a buckling sound employs linear modal analysis of the deformed thin shell. Because different buckling events in general occur at different deformed configurations, the question arises whether the calculation of linear modes can be reused. We amortize the cost of the linear modal analysis by dynamically partitioning the mesh into nearly rigid pieces: the modal analysis of a rigidly moving piece is retained over time, and the modal analysis of the assembly is obtained via Component Mode Synthesis (CMS). We illustrate our approach through a series of examples and a perceptual user study, demonstrating the utility of the sound synthesis method in producing realistic sounds at practical computation times.

**Keywords:** Sound synthesis, thin shells, buckling, component modal synthesis

**Concepts:** •Computing methodologies → Physical simulation; •Applied computing → Sound and music computing;

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights

## 1 Introduction

Whether we ball up a plastic bag to reduce, reuse, or recycle; open a foil-lined bag of potato chips; step and crush a soda can; or fiddle with our cellophane candy wrapper to the joy of our fellow opera-goers; one thing is certain: these activities would never be the same were it not for the characteristic sound of *crumpling*. Crumpling sounds originate primarily from *buckling*, in which a thin material bends and gives way under pressure or strain. The buckling region accelerates rapidly, exciting audible vibrations throughout the object. Whereas the dynamics and numerical computation of buckling deformations are relatively well studied, the numerical synthesis of buckling sounds has not received significant attention.

We seek to automate the synthesis of crumpling sound to accompany a given crumpling animation. The input to our method is an animated triangle mesh produced by some external thin shell simulator. We are charged with detecting buckling events, and synthesizing corresponding buckling sounds. For this, we target a physically based approach. Physics-based sound synthesis is appealing because it essentially leads to an “encoding” of the physics behind the phenomena as an algorithm, while avoiding the need of recorded sounds. However, there are two central challenges that must be overcome to make it practical.

Audible buckling events vary in spatial scale, from highly localized events invisible to the naked eye to visually significant events involving large deformations. Detecting the entire range of events

for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org). © 2016 ACM.

SA '16 Technical Papers., December 05-08, 2016, , Macao

ISBN: 978-1-4503-4514-9/16/12

DOI: <http://dx.doi.org/10.1145/2980179.2982400>

would require that they are first of all adequately captured by the input mesh animation, which would require a spatiotemporal resolution of the mesh animation that is grossly excessive from a visual perspective. To allow for typical input mesh animations targeted at visual simulations, we must devise an alternative to detecting visually imperceptible yet audible buckling events.

The spectrum of a vibrating sound is dependent on the vibration modes of the shape, but the modes are constantly changing, because of the dynamically evolving thin shell mesh. Therefore, the sound of a buckling event depends on the deformed shape at the time of buckling. Our challenge is that crumpling is a composition of myriad buckling events, each over a different deformed configuration. Brute force analysis of the vibration modes for each of the myriad deformed shapes is expensive, while approximating the analysis over a single representative (e.g., initial) shape would produce unrealistic sounds.

Our work addresses these central challenges of filling in the buckling events that are missing due to limited spatiotemporal input mesh resolution, and making the computation tractable by amortizing the modal analysis over a dynamically evolving mesh.

Using an off-the-shelf elastoplastic thin shell simulator [Narain et al. 2012; Narain et al. 2013], we produce an input animation of a crumpling scenario. We identify the visually perceptible buckling events by detecting sudden curvature changes, and use the velocity of the buckling regions to excite the linear vibration modes computed at the deformed configuration, thus producing the corresponding buckling sounds. When assembling the entire sequence of buckling sounds in this way, the initial “skeleton” of a crumpling sound emerges.

To fill in this skeleton, we employ a stochastic model to generate buckling events at visually imperceptible spatiotemporal scales (§6). The model is driven by “crumpling activity”, so that new buckling events are added only at times and regions where buckling is likely to occur. The intensity of the generated buckling events follows a distribution observed empirically by statistical physicists over various materials [Kramer and Lobkovsky 1996; Houle and Sethna 1996; Abobaker et al. 2015].

We observe that when an object crumples, parts of the object tend to move nearly rigidly during portions of the crumpling motion. Leveraging this observation, we amortize the cost of modal analysis by reusing previously computed modes in nearly rigid regions (§5). We detect mesh regions that move near-rigidly over spans of time, and adapt *Component Modal Synthesis* (CMS) [Craig and Bampton 1968; Craig 2000] to reuse the modes of such components. The result is a speedup up to  $8.5\times$  in the computation time of modal analysis.

We demonstrate the practicality and expressive range of the method through the synthesis of prosaic sounds such as crushing soda cans, handling plastic bags, and twisting candy wrappers (see Figure 1). We show that the computation times are practical, and that the sound spectra and energy distributions are comparable to empirical observations.

Further, in a modest perceptual study, we demonstrate that when human test subjects are asked to associate sound to corresponding visuals, the synthesized and recorded sounds both lead to similar rates of successful association. We also show that they are able to establish this association much better with the proposed method compared to existing methods for thin shell sound synthesis, which do not explicitly focus on crumpling.

## 2 Related Work

**Sound synthesis.** Modal vibration models have been used to synthesize audio for (nearly) rigid objects during impact or frictional contact [Cook 2002; Doel et al. 2001; O’Brien et al. ; Zheng and James 2011]. Chadwick et al. [2009] focused on nearly rigid thin shells that vibrate under small deflections. They synthesized thin-shell impact sounds which usually last long (a few seconds) in duration. Their method accounts for mode couplings using a nonlinear vibration model. By contrast, we focus on crumpling, which involves *gross* elastic and plastic deformation. Since the clicking sounds produced by buckling are typically short (50ms) in duration [Houle and Sethna 1996], we harness time-varying *linear* modal synthesis to synthesize crumpling sound.

Earlier work on sound synthesis for thin shells focused on small time step explicit integration of volumetric finite elements [O’Brien et al. 2001] or real-time synthesis based on heuristic selection of pitch frequency and decay time [Fontana and Bresin 2003]. There are also data-driven approaches specifically tailored for cloth [An et al. 2012] and paper [Schreck et al. 2016] sound synthesis, which perform friction and buckling event detection to guide a synthesis step that relies on real sound recordings. Among methods that build on physical principles rather than stored recordings, our approach is positioned to offer practical runtimes and richer, more recognizable crumpling sounds than what was previously possible.

**Thin shell simulation.** Graphical simulation of thin shells has been studied for nearly three decades [Terzopoulos et al. 1987; Baraff and Witkin 1998]. Most modern approaches rely on discrete shell models well suited for computational efficiency; these formulate simple bending energies while allowing non-flat rest-shapes [Grinspun et al. 2003; Bridson et al. 2003]. Plasticity can be incorporated [Gingold et al. 2004; Narain et al. 2013], allowing crumpling simulations with fine creases. Some investigations have focused specifically on efficient simulation of cloth buckling [Choi and Ko 2002; Zhou et al. 2008]. Narain et al. [2012; 2013], whose software we use to produce our input animations, focused on adaptive remeshing of cloth and shell simulations. By locally refining or coarsening the triangle mesh as needed, their approach is able to resolve the sharp creases formed when a thin shell buckles.

**Component Mode Synthesis (CMS).** CMS has been used to compute the modal analysis of complex structures since the pioneering work of Hurty [1965]. In CMS, a structure is subdivided into substructures, reduced-order models are computed or measured for each substructure, and a reduced-order model is computed for the entire structure by assembling the substructure models [Craig 2000]. CMS has been widely adopted in engineering to study large assembled systems [Bathe and Dong 2014]. Other use cases of CMS include the study of fluid-solid coupled vibrations [Stammberger and Voss 2008], wind turbine dynamics [Holm-Jorgensen and Nielsen 2009], or the analysis of systems by mixing measured and analytical vibration responses [De Klerk et al. 2008]. To our knowledge, CMS has not been applied to sound synthesis, or indeed in the field of computer graphics with the only notable exception of the work of Yang et al. [2013] where boundary modes are computed using a CMS formulation. We apply CMS to sound synthesis in order to amortize the cost of modal analysis over a sequence of meshes sharing similar substructures.

## 3 Method Overview

**Input mesh animation.** The input to our algorithm is a visually plausible triangle mesh animation of crumpling behavior. Our

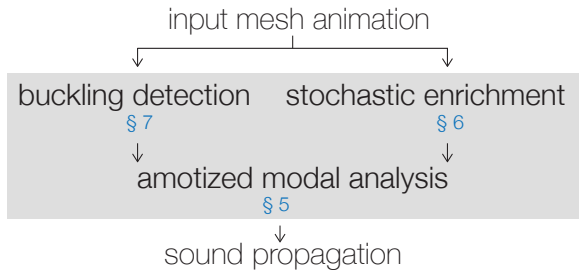


Figure 2: Method overview

method does not require us to modify the numerical method that produces this input animation. Instead, we estimate velocities and deformation energies directly from the animated mesh. In our examples, we used the off-the-shelf thin shell simulator ARC-Sim [Narain et al. 2012; Narain et al. 2013].

As depicted in Figure 2, we synthesize crumpling sound in four steps:

**Buckling detection.** First, we detect mesh buckling events at each animation frame (see Figure 9). From a kinematic perspective, buckling is characterized by a sudden brief localized deflection, with attendant rapid local change in surface curvature. We therefore use a curvature-based metric to perform the detection. Contiguous buckling vertices are grouped into *buckling patches* (Figure 3), each representing a distinct buckling event. Assuming the elastic energy of a patch is converted into vibrational energy, we estimate the post-buckling velocity of the patch, which will be used as an initial condition to drive a standard modal vibration model (§4.3). The implementation of buckling patch computations is postponed to Section 7.

**Modal vibrations (§5).** The modal vibration model is formulated in terms of a generalized eigenvalue decomposition (GED) of the material stiffness matrix  $\mathbf{K}$ . Because a shell is largely deformed by crumpling, the stiffness matrix  $\mathbf{K}$  varies significantly over the dynamic evolution. This necessitates the updating of  $\mathbf{K}$ —and crucially the recomputation of the GED—at the instant of each buckling event. Actually carrying out such frequent recomputation would be prohibitively expensive. To address this problem, we amortize the construction of the linear modal vibration model by judiciously reusing local vibrational modes of near-rigid regions using Component Mode Synthesis, as detailed later in §5. These vibration modes are excited by the velocity of the detected bucklings, producing a “clicking” sound [Kramer and Lobkovsky 1996].

**Sound enrichment (§6).** Detecting visually evident buckling events produces a sound “skeleton” of audible clicks, but lacks the fullness, or density of clicks, arising from visually imperceptible buckling events. We enrich the skeleton by exploiting the physical studies on acoustic emission statistics of crumpled sheets [Kramer and Lobkovsky 1996; Houle and Sethna 1996; Abobaker et al.

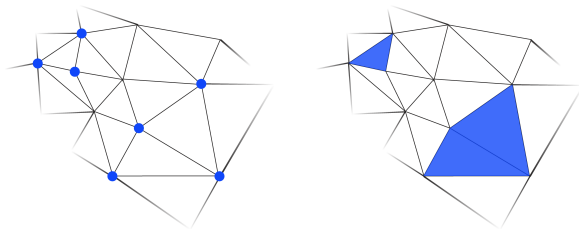


Figure 3: We group every contiguous neighborhood of buckling vertices into a buckling patch.

2015], in a way similar to the turbulence enrichment for fluid simulation [Kim et al. 2008]. Without resort to expensive high-resolution simulation, we stochastically inject small-scale buckling events such that the overall energy distribution of both the simulated and stochastically generated buckling sounds satisfies a universal power law.

**Sound propagation.** Lastly, we estimate the modal sound propagation, following the standard far-field acoustic transfer approximation [Chadwick et al. 2009; Zheng and James 2010]. We use the fast multipole boundary element method [Shen and Liu 2006] to solve the Helmholtz equation for every vibrational mode and synthesize the final sound [James et al. 2006].

## 4 Challenges to Crumpling Sound Synthesis

Thin shells have been called the *prima donna* of mechanical structures [Ramm and Wall 2003]. On the one hand, they offer the opportunity for exceptionally efficient construction of load bearing structures. On the other hand, such structures have the potential to dramatically fail, because of a shell’s fickle tendency to buckle.

### 4.1 Rationalizing the Sound of Buckling

The propensity to buckle is best understood by examining shell *geometry*, which contrasts one exceptionally thin length scale—the thickness—against a much larger length scale (shell diameter, radius of curvature, or “characteristic length”) [Balankin et al. 2015; Sethna et al. 2001; Wood 2002; Vliegenthart and Gompfer 2006]. A remarkable consequence of the contrasting length scales is that the membrane stiffness is several orders of magnitude greater than the bending stiffness, leading to a preference of bending (normal deflection) over membrane (in plane compression) modes.

Except under very special circumstances (e.g., standing on eggshells), general attempts (e.g., via forcing or boundary conditions) to store energy in a compressive membrane state are typically foiled by a rapid (<50ms) and violent normal deflection in which energy is transferred into stored elastic bending, plastic dissipation, and kinetic energy [Houle and Sethna 1996; Pandey et al. 2014]. The sound produced by a buckling event can be rationalized via this violent transition: the local buckling introduces local accelerations that excite global, audible vibrations [Kramer and Lobkovsky 1996].

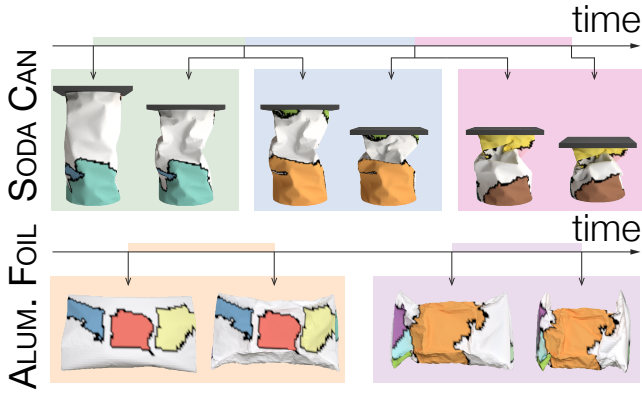
The specifics of the buckling process depend on the energy distribution, geometrical and material properties of the shell [Sun et al. 1995]. For certain geometries, a rapid cascade of buckling events leads to crumpling [Cerdeja et al. 1999]. At first glance appearing to be absolutely unorganized, the geometry of crumpling is in fact statistically reproducible [Balankin et al. 2015].

### 4.2 Thin Shell Dynamics

To synthesize buckling sound we must first lay down the foundations for numerical treatment of shell dynamics. We discretize the spatial domain of the thin shell with a triangle mesh. The dynamics of the spatially discretized shell are governed by Newton’s second law of motion,

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{D}\dot{\mathbf{u}} + \mathbf{f}_{int}(\mathbf{u}) = \mathbf{f}_{ext}, \quad (1)$$

where  $\mathbf{u}$  is the vector of nodal displacements,  $\mathbf{M}$  and  $\mathbf{D}$  are the mass and damping matrices, respectively,  $\mathbf{f}_{ext}$  is the vector of external nodal forces, such as gravity and contact, and  $\mathbf{f}_{int}(\mathbf{u})$  is the vector of internal nodal forces, which typically includes Green membrane stress and discrete bending model [Bridson et al. 2003; Grinspun et al. 2003]. Plasticity and non-flat rest



**Figure 4:** The evolution of near-rigid thin shell regions of two examples: each block indicates a time frame interval, in which the near-rigid mesh regions are color-mapped.

angles may be treated via explicit embedding of the plastic configuration [Narain et al. 2013].

### 4.3 Linear Modal Analysis (LMA) for Sound Synthesis

Having established the dynamics, we turn to modal sound synthesis to produce audio from shell vibrations [Chadwick et al. 2009]. Assume for the moment that a thin shell is *nearly rigid*: deformations are limited to visually imperceptible vibrations. Under this assumption, we may analyze vibrational deformations at some configuration of interest (i.e., an animation frame) by linearizing (1),

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{D}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}_{ext}, \quad (2)$$

where  $\mathbf{K} = \partial_{\mathbf{u}} \mathbf{f}_{int}(\mathbf{u})$  is the stiffness matrix at the deformed configuration  $\mathbf{u}$ . *Linear modal analysis* (LMA) then seeks a decoupled basis for the linear system by solving the generalized eigenvalue problem [Shabana 2012]

$$\mathbf{K}\mathbf{U} = \mathbf{M}\mathbf{U}\mathbf{S} \quad (3)$$

to compute the modal shape matrix  $\mathbf{U}$  and a (diagonal) eigenvalue matrix  $\mathbf{S}$ . Each column of  $\mathbf{U}$  encodes the vibration (oscillatory displacement) of a mode, while the corresponding diagonal entry of  $\mathbf{S}$  encodes the mode's squared undamped natural frequency (i.e.,  $\mathbf{S}_{i,i} = \omega_i^2$ ). Substituting  $\mathbf{u} = \mathbf{U}\mathbf{q}$  rewrites the displacements  $\mathbf{u}$  as a vector of *modal amplitudes*  $\mathbf{q}$ . Premultiplying  $\mathbf{U}$  on both sides of (2) then yields the decoupled family of ordinary differential equations, each univariate equation

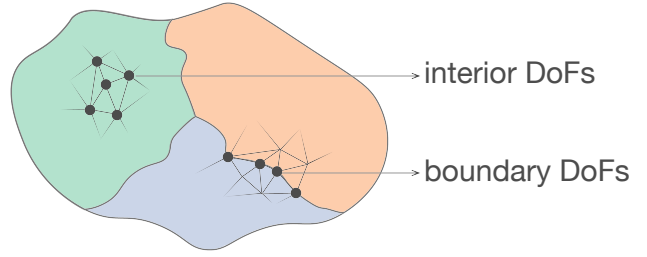
$$\ddot{\mathbf{q}}_i + d_i \dot{\mathbf{q}}_i + \omega_i^2 \mathbf{q}_i = \mathbf{U}_i^T \mathbf{f}_{ext} \quad (4)$$

describing temporal evolution of a single complex-valued amplitude  $\mathbf{q}_i$ , where  $d_i$  is the damping coefficient of mode  $i$ , and  $\mathbf{U}_i$  is the  $i$ -th column of  $\mathbf{U}$ .

**The amortization challenge.** Crumpling, comprised of a complex cascade of buckling events, introduces drastic deformations, with attendant alterations of the modal frequency and shape matrices ( $\mathbf{U}, \mathbf{S}$ ). Each alteration of the matrices induces a corresponding change to the solution of the generalized eigenvalue problem (3). Efficient, high-quality synthesis of crumpling sound therefore hinges on accurate amortization of the computation (3) over a sequence of shapes  $((\mathbf{U}^{(1)}, \mathbf{S}^{(1)}), (\mathbf{U}^{(2)}, \mathbf{S}^{(2)}), \dots)$  that are related by temporal, dynamic evolution.

## 5 Amortized Modal Analysis

When a thin shell crumples, alternating parts of the shell tend to move almost rigidly across multiple simulation frames, while



**Figure 5:** Substructures in CMS: each colored region represents a substructure (or mesh component) of the thin shell. They are coupled through boundary vertices.

others are deformed (refer to Figure 4). The segmentation of the geometry into rigid and deformable parts itself varies over time: a region that has held near-rigid for a duration may crumple before returning to near-rigid motion. Since LMA is applicable to near-rigid materials, amortizing the cost of LMA requires a method to reuse modal information for near-rigid parts, while incorporating updated modal frequency and shape matrices for deforming parts.

*Component Mode Synthesis* (CMS) [Bathe and Dong 2014; Craig and Kurdila 2006] offers a natural solution for partial recomputation of LMA. Given a segmentation of a shape into parts, CMS first performs LMA on individual parts and then combines the analyses to model the vibrational mode of the entire shape. We propose to identify parts of the thin shell that evolve near-rigidly over spans of time, computing the LMA for such parts just once. We can then harness CMS to combine the LMAs of different parts to discover the global vibration modes. To exploit this idea, we must decompose the mesh animation in time and then space to maximize (LMA reuse of) near-rigid regions.

After introducing CMS (§5.1), we will analyze its complexity for our application (§5.2) and then apply dynamic programming and clustering to decompose the mesh animation in time and space, respectively (§5.3).

### 5.1 Component Mode Synthesis

Component Mode Synthesis (CMS), which breaks down the analysis of complex structures via analysis of substructures [Craig 2000; De Klerk et al. 2008], comes in fixed- and free-interface variants. We adopt the fixed-interface approach, which assumes fixed boundaries between substructures when computing the LMA, and then adds those boundaries directly as degrees of freedom during assembly [Craig and Bampton 1968].

Concretely, we are concerned with the procedure of solving the generalized eigenvalue problem (3). Suppose the mesh is decomposed into a number of mesh components incident at their boundaries (Figure 5). CMS essentially amounts to an algebraic substructuring method, in which we first permute the rows and columns of the (symmetric) mass and stiffness matrices,  $\mathbf{M}$  and  $\mathbf{K}$  so that they can be partitioned as

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_{bb} & \mathbf{M}_{bi} \\ \mathbf{M}_{bi}^T & \mathbf{M}_{ii} \end{pmatrix} \text{ and } \mathbf{K} = \begin{pmatrix} \mathbf{K}_{bb} & \mathbf{K}_{bi} \\ \mathbf{K}_{bi}^T & \mathbf{K}_{ii} \end{pmatrix},$$

where the submatrices  $\mathbf{K}_{bb}$  and  $\mathbf{M}_{bb}$  correspond to the mesh degrees of freedom (DoFs) at all component boundaries;  $\mathbf{K}_{bi}$  and  $\mathbf{M}_{bi}$  correspond to the mesh DoFs interior to all the components. We further structure each submatrix into components: both  $\mathbf{K}_{ii}$  and  $\mathbf{M}_{ii}$  are block diagonal matrices, each block corresponding to



a mesh component. Finally, the components are coupled together at the boundaries through the submatrices  $\mathbf{M}_{bi}$  and  $\mathbf{K}_{bi}$ .

CMS begins by analyzing each component in isolation to obtain a matrix of “internal” modes  $\mathbf{U}_{ii}$ , which has the same block diagonal structure as  $\mathbf{M}_{ii}$  and  $\mathbf{K}_{ii}$ . To determine each block of  $\mathbf{U}_{ii}$ , we solve an LMA over the corresponding blocks of  $\mathbf{M}_{ii}$  and  $\mathbf{K}_{ii}$ . Here, the assumption of *fixed* boundary positions is invoked when performing the LMA *only* over nodes *internal* to a component.

Next, the modes  $\mathbf{U}_{ii}$  are augmented with *constraint* modes  $\mathbf{U}_c = -\mathbf{K}_{ii}^{-1}\mathbf{K}_{ib}$  that account for the displacement of the interior of components due to displacements of component boundaries. CMS then constructs the Craig-Bampton transformation

$$\Psi = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{U}_c & \mathbf{U}_{ii} \end{pmatrix}, \quad (5)$$

which yields a new matrix pair

$$\widehat{\mathbf{K}} = \Psi^T \mathbf{K} \Psi \text{ and } \widehat{\mathbf{M}} = \Psi^T \mathbf{M} \Psi. \quad (6)$$

The eigenvalues of  $(\widehat{\mathbf{M}}, \widehat{\mathbf{K}})$  are identical to those of  $(\mathbf{M}, \mathbf{K})$ , and the transformed and original eigenvectors are related by  $\Psi \widehat{\mathbf{U}} = \mathbf{U}$ .

However, from a computational point of view, solving the transformed problem appears less desirable: while  $(\mathbf{M}, \mathbf{K})$  are sparse, their transformed counterparts are typically dense.

To offset the advantage of sparsity, when we work with the dense system we must reduce it to a manageable size. We do this by discarding less significant vibration modes (columns) from  $\mathbf{U}_{ii}$ , which (originally square) becomes rectangular. In truncating modes from  $\mathbf{U}_{ii}$  we must strike a balance between making the dense eigenvalue problem manageable but keeping the solution accurate.

Propagating the truncation to (5) and (6) gives a rectangular matrix  $\Psi_r$ , and in turn square *but smaller* mass and stiffness matrices  $(\mathbf{M}_r, \mathbf{K}_r)$ . Following the standard Rayleigh-Ritz theory [Parlett 1980], the eigenvalues of  $(\mathbf{M}_r, \mathbf{K}_r)$  are good approximants to those of  $(\mathbf{M}, \mathbf{K})$ , and original eigenvectors are approximately recovered by  $\mathbf{U} \approx \Psi \mathbf{U}_r$ .

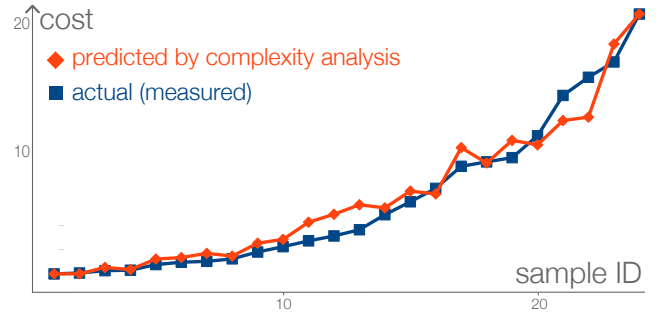
In short, component modal synthesis sacrifices accuracy in the solution of the global linear modal analysis in exchange for the freedom to solve independently (and potentially in parallel), or in our application *reuse*, the solution of component subproblems.

## 5.2 CMS Complexity Analysis

In order to optimize the spatiotemporal mesh decomposition for CMS, we first analyze the computational complexity of CMS as applied to crumpling sound synthesis. For the sake of this analysis, consider  $L$  consecutive frames. Without loss of generality, assume that the mesh animation is dense in buckling events, i.e., each of the  $L$  frames includes a buckling event. To lift this assumption, we may consider the subsequence of animation frames that represent buckling events.

Suppose that the mesh is decomposed into  $R$  near-rigid components (denoted as  $\mathbb{G}$ ) and  $D$  deforming components (denoted as  $\mathbb{D}$ ), and this composition holds for all  $L$  consecutive frames. The cost of CMS-based sound synthesis can be broken down qualitatively into three actions:

- (i) for each rigid component, solve the truncated LMA *just once*
- (ii) for each deformable component  $i \in \mathbb{D}$ , recompute the truncated LMA *every frame*,
- (iii) coalesce by solving the reduced LMA  $(\mathbf{M}_r, \mathbf{K}_r)$  *every frame*.



**Figure 6:** We compare the cost of CMS (vertical axis, in minutes), as predicted by (7) (red) versus actual measurements (blue), over an arbitrary sequence of 25 different decompositions of increasing complexity. The decompositions are sorted (horizontal axis) based on the actual cost of CMS; a perfect predictor would therefore also be monotonically increasing. Our estimate, (7), has relative error of mean 15% and standard deviation 10%.

Therefore, the overall computational complexity is

$$C = \underbrace{\sum_{i \in \mathbb{G}} \Gamma(d_i, m_i)}_{\text{rigid}} + L \underbrace{\sum_{i \in \mathbb{D}} \Gamma(d_i, m_i)}_{\text{deformable}} + L \underbrace{\Gamma(d_T, S)}_{\text{reduced}}, \quad (7)$$

where  $d_T$  denotes the total DoFs of the entire mesh,  $S = \sum m_i$  denotes the total number of truncated modes over all components, and  $\Gamma(d, m)$  denotes the cost of LMA for a mesh with  $d$  DoFs and  $m$  modes. The form of  $\Gamma(d, m)$  depends on the specific numerical algorithm (e.g., whether it is a Lanczos method or Arnoldi method) [Golub and Van Loan 2012]. For example, through experimental analysis, we have observed that the Matlab’s eigs implementation has a complexity of  $O(dm^2)$ .

This cost estimation does not include the construction of  $\Psi$ ,  $\mathbf{M}_r$ , and  $\mathbf{K}_r$ , via (5) and (6); that cost is relatively negligible. For purposes of selecting a decomposition well suited to CMS, (7) serves as a good predictor of actual cost (see Figure 6).

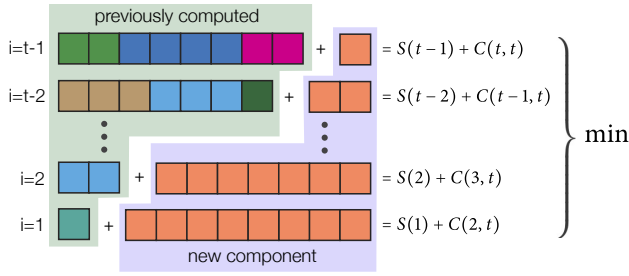
## 5.3 Optimizing CMS Sequence

Having modeled the complexity of CMS, we can now seek a good temporal and spatial decomposition of the given mesh animation, one that reduces the aggregate cost of CMS over the complete input animation. Searching the entire space of spacetime decompositions appears to be a challenging combinatorial problem; by adding additional structure and focusing our attention on *time-then-space* decompositions, we can formulate a polynomial time algorithm that can be better characterized in terms of its optimality.

**Time-then-space decomposition.** We partition a timeline of  $N$  frames into a list of frame intervals, whose concatenation spans the timeline (see Figure 4). For every interval, we partition the mesh combinatorics into (non-overlapping) submeshes, whose union forms the complete mesh; the mesh partition is invariant over that interval. Such decompositions, which partition first in time and then in space, define our search space.

**Temporal decomposition via dynamic programming.** Temporally partitioning the sequence of  $N$  frames into non-overlapping intervals, and seeking to minimize the aggregate cost of CMS, is a combinatorial optimization problem. However, our problem can be solved efficiently and optimally using dynamic programming.

Let  $S(t)$  denote the *minimal* aggregate cost of CMS up to the frame  $t$ . Let  $C(a, b)$  denote the cost of CMS over a single inter-



**Figure 7:** Depiction of the  $t$ 'th dynamic programming iteration: prior to this iteration, we have already computed the optimal decomposition for the  $t-1$  intervals  $1 \dots i$ , and their associated costs  $S(i)$ ,  $1 \leq i < t$ . In this iteration we are tasked with computing  $S(t)$ . We consider the  $t-1$  possibilities of concatenating a previous decomposition  $1 \dots i$  with a new interval  $i+1 \dots t$ , and select the least expensive among these.

val spanning frames  $a \dots b$ . The dynamic program follows the causality condition (see Figure 7)

$$S(t) = \min_{i=1 \dots t-1} [S(i) + C(i+1, t)].$$

The causality condition leads to a pair of nested iterations each of complexity  $O(N)$ ; as we shall see below,  $C(i+1, t) \in O(N)$ ; therefore, the overall complexity is  $O(N^3)$ . In addition, this process can be made even more efficient thanks to multi-core parallelization.

**Spatial decomposition in a frame interval.** The temporal decomposition makes reference to  $C(a, b)$ , the cost of CMS over a given frame interval  $a \dots b$ . We evaluate  $C(a, b)$  by first spatially decomposing the mesh and then evaluating CMS complexity (7).

The complexity of CMS is reduced by identifying components that evolve near-rigidly. We adopt the Mean-Shift clustering of per-triangle rotations proposed by James and Twigg [2005], and refer the reader to details therein. In short, for each triangle  $j$  we compute via polar decomposition the rotations  $R_j^{a+1} \dots R_j^b \in SO(3)$  that best map its position in frame  $a$  to its positions at frames  $a < i \leq b$ . We concatenate the  $(b-a+1)$  rotation matrices of the triangle into a vector  $\mathbf{z}_j \in \mathbb{R}^{9(b-a+1)}$ . We then perform mean shift clustering of the  $\{\mathbf{z}_j\}$  rotation vectors. This yields a set of triangle clusters that are labeled as moving near-rigidly.

We further split these clusters into connected mesh components. All the remaining triangles not belonging to near-rigid regions are labeled as deformable and grouped into deformable components.

Observe that the length of  $\mathbf{z}_j$  is  $O(N)$ ; the complexity of the spatial partitioning is therefore linear in the number of frames  $(b-a+1)$ .

Having identified a good spatial partition for the given interval, we use (7) to compute  $C(a, b)$ , the complexity of CMS over the interval  $a \dots b$ .

## 5.4 Implementation Details

We highlight a few notable implementation details.

Both for spatial decomposition of the mesh and the CMS, it is fundamental to have a sequence of meshes with a fixed mesh topology throughout the entire timeline. While our thin-shell simulation remeshes adaptively, we remesh again the output meshes by uniformly resampling the material space. As a side effect, this step also improves the mesh quality for LMA, eliminating large coarse regions which would impair the mode shapes. We enforce

that the resampled number of vertices is enough to produce vibrations within the entire audible range (20Hz - 20kHz). In practice, we remesh to get approximately 10K nodes.

To construct  $\Psi$  in (5), we compute  $\mathbf{U}_c = -\mathbf{K}_{ii}^{-1} \mathbf{K}_{ib}$  by solving the linear problem  $\mathbf{K}_{ii} \mathbf{U}_c = -\mathbf{K}_{ib}$ . We use the standard Preconditioned Conjugate Gradient solver with an incomplete Cholesky factorization as preconditioner. The amount of linear system solves, one per column of  $\mathbf{U}_c$ , is equal to the number of boundary DoFs, and they can easily be run in parallel on a multi-core CPU.

The modes of near-rigid regions are reused within a frame interval, but they must be rotated to account for rigid region rotations introduced by plastic deformations. We compute the rotation of each region following [James and Twigg 2005], i.e., we compute the area-weighted average triangle rotation of the region, and then perform a polar decomposition [Moakher 2002].

A final notable detail is the policy for mode truncation after LMA is executed on each mesh component. In practice, we set a target number of modes  $m_t$  for all components combined, and each component retains a portion proportional to the number of DoFs of that component. With this policy, the size of the CMS subspace is  $S = m_t + m_c$ , with more modes assigned to larger components.

## 6 Power-Law Buckling Sound Enrichment

The acoustic energy emitted from buckling events varies over six orders of magnitude [Kramer and Lobkovsky 1996; Houle and Sethna 1996; Abobaker et al. 2015]. This variation is mainly due to the extremely inhomogeneous distribution of elastic stresses in a crumpled configuration. As such, the variation is not specific to one material; the aforementioned works report this observation for diverse materials such as paper, plastic, and aluminum.

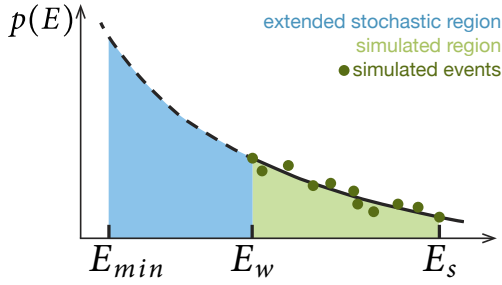
Even a tiny buckling event, which can be hardly visible, produces audible sounds. Yet, any crumpling simulation has limited spatial and temporal resolution. Increasing the resolution would allow capturing smaller buckling events, but would also result in a prohibitive increase of simulation cost.

Fortunately, only the buckling events that are visible need to have perfectly synchronized and perceptually matching acoustic counterparts. We therefore propose an algorithm that generates small-scale buckling sounds procedurally, thereby enriching the simulated buckling sounds without increasing the simulation cost.

**Universal power law.** Crumpling of a thin shell has been extensively studied in statistical physics. An important discovery is a universal power law distribution of the buckling acoustic energies (i.e.,  $p(E) \sim E^{-\alpha}$ ); one power law holds for a variety of shell sizes and materials [Houle and Sethna 1996]. The emergence of the power law is attributed to the quenched curvature disorder (or random, stationary network of creases and ridges) produced during the crumpling process and the attendant complex energy landscape [Kramer and Lobkovsky 1996]. We make the assumption that this power law applies throughout the crumpling process.

**Method in brief.** Adopting the power law distribution as our *ansatz*, our method extends the buckling energy spectrum: whereas mechanical simulation (§5) populates the higher-energy infrequent tail of the spectrum, our stochastic process fills in the low-energy frequent events. Referring to Figure 8, suppose that the mechanical simulation produces acoustic energies  $[E_w, E_s]$  following a power law distribution. We aim to fill in those less energetic buckling events  $[E_{\min}, E_w]$  that complete the spectrum.

We first trigger stochastic buckling events as a function of crumpling activity, and assign their buckling energies that are smaller



**Figure 8:** Stochastic sound enrichment: the simulated buckling events have a band-limited buckling energy spectrum, which, based on statistical physics, should satisfy a power law [Houle and Sethna 1996]. We extend the energy spectrum by sampling small-scale low-energy buckling events following the power law.

than  $E_w$  and follow the same power-law distribution as the simulated bucklings. Lastly, like the simulated buckling events, the stochastically generated buckling events also excite the modal vibration and produce sound details.

### 6.1 Triggering Stochastic Buckling Events

We first trigger buckling events stochastically on the simulated mesh. This process is guided by the crumpling activity of the material. Intuitively, the more the material deforms in a given period of time, the more likely it is to buckle during that time. We measure crumpling activity for triangle  $i$  from frame  $t-1$  to frame  $t$  by computing the variation of its elastic energy,  $\Delta E_i$ , during the frame interval. We then estimate a probability  $p_i$  for every triangle  $i$  becoming buckled following

$$p_i(t) = C \Delta E_i, \quad (8)$$

where the constant  $C$  is a scaling constant, introduced to control the overall number of stochastic buckling events that are triggered. The value of  $C$  is determined automatically such that simulated and stochastic energy distributions match, as explained later in §6.3. After computing  $p_i$ , we randomly assign each triangle as *buckling triangle* with the probability  $p_i$ .

### 6.2 Stochastic Buckling Energy

Next, we assign each buckling triangle a buckling energy value following the power law distribution. Formally, we use a power-law Pareto distribution [Feller 2008], whose cumulative distribution function  $P$  is

$$P(E) = 1 - \left( \frac{E_{\min}}{E} \right)^\alpha, \quad (9)$$

which describes the probability of a buckling energy less than or equal to  $E$ . Observed in many studies [Kramer and Lobkovsky 1996; Houle and Sethna 1996], the power-law coefficient  $\alpha$  is in the range of  $[1, 1.3]$ , depending on the type of material. We fit the  $\alpha$  value based on the energy distribution of simulated buckling events.  $E_{\min}$  is the minimal energy level that can be generated by the power law. Kramer [1996] observed six orders of magnitude in the energy spectrum. Thus, we set  $E_{\min}$  as  $1 \times 10^{-6}$  times the highest energy released by simulated buckling events. In practice, energy values are relative since the sound, when stored in a wave form file, is normalized with respect to the highest amplitude.

The cumulative distribution function allows us to use importance sampling to assign energy values satisfying the power law distribution (9) [Rubinstein and Kroese 2011]. For each buckling triangle  $i$ , we assign an energy value  $E_i = E_{\min}(1 - \chi)^{-\frac{1}{\alpha}}$ , where  $\chi$  is a random variable uniformly distributed in  $(0, 1]$ .

With the sampled buckling energy  $E_i$  at the triangle  $i$ , we grow a region as the buckling patch. Adopting the same idea of energy-based sound excitation for simulated buckling events (§3), we grow a patch starting from the triangle  $i$  and incrementally including the 1-ring neighbors of the triangles in the region. We stop growing the patch when the sum of the elastic energy of all the triangles inside the patch is above  $E_i$ . This results in a weak coupling between buckling energy and the size of the buckling patch, coherent with observations in previous work [Kramer and Lobkovsky 1996; Houle and Sethna 1996]. Once a stochastic buckling event is fully defined by its location, size, and energy, we compute its effective vibrational velocity and produce the modal vibration, following the sound excitation step described in §3.

**Energy buildup and release.** Every buckling event releases energy locally, preventing other events from triggering in the neighborhood until enough elastic energy builds up again. Since our stochastic model runs as a post-process after the crumpling simulation, we approximate this behavior by introducing a cooldown period. This draws inspiration from the work of DiDonna [2002], who observed that, after initial ridge formation, the buckling threshold energy of the ridge is an approximately constant scaling  $s$  of its initial energy. Theoretical derivation and experimental simulation lead to an  $s$  value around 1.2.

We model the energy buildup that leads to a buckling event by leveraging this constant scaling  $s$ , with  $s = 1.2$ . In practice, once a region has buckled, its triangles are not allowed to trigger a new stochastic event until their energy has been scaled by  $s$ . We use as reference energy for each triangle the lowest energy since it last buckled. We note that simulated bucklings automatically contribute to the cooldown effect by naturally releasing energy: a stochastic buckling is less likely to happen in a region that buckled recently in the simulation.

### 6.3 Continuity of Energy Distributions

We expect the combination of simulated and stochastically generated bucklings to satisfy the power-law energy distribution. The scaling constant  $C$  in (8) allows us to control the amount of stochastic buckling events, and thereby enforce continuity of the simulated and stochastic energy distributions.

Suppose there exist  $N_{\text{sim}}$  simulated buckling events, among which the strongest and weakest have the energies  $E_s$  and  $E_w$ , respectively. Then, the cumulative distribution of buckling energy distributed in  $[E_w, E_s]$  is

$$F([E_w, E_s]) = P(E_s) - P(E_w) = \left( \frac{E_{\min}}{E_w} \right)^\alpha - \left( \frac{E_{\min}}{E_s} \right)^\alpha. \quad (10)$$

Since we know the distribution  $F([E_w, E_s])$  yields  $N_{\text{sim}}$  simulated bucklings, the total cumulative distribution (which is equal to 1) must yield an expectation of  $N_{\text{tot}} = N_{\text{sim}} / F([E_w, E_s])$  bucklings. The scaling constant  $C$  is therefore iteratively adjusted (using a binary search in practice) until the stochastic model produces  $N_{\text{tot}} - N_{\text{sim}}$  bucklings up to an acceptable margin of error. Our experience shows that starting at  $C = 1$ , up to 3 iterations are required when using a margin of error of 10%.

## 7 Implementation

Our approach requires knowledge of vertex positions and velocities, the global stiffness and mass matrices, and the elastic energy of each triangle, for each simulated frame. Open-source simulators can be slightly modified to output this data. Black-box simulators may output simply vertex positions, from which velocities can be estimated by finite differences, and the remaining quantities can be estimated from positions and velocities. We lightly



|                     | Avg. #<br>triangles | Length<br>(s) | Sim.<br>time (hr) | Full LMA<br>cost (hr) | CMS<br>cost (hr) | Clust.<br>cost (hr) | CMS<br>€ | Speed<br>up | Sound<br>modes | Eq. (4)<br>cost (s) |
|---------------------|---------------------|---------------|-------------------|-----------------------|------------------|---------------------|----------|-------------|----------------|---------------------|
| SODA CAN            | 9350                | 1             | 1.2               | 15.9                  | 2.5              | 0.02                | 0.06     | 6.3×        | 2500           | 62                  |
| PLASTIC BAG         | 8660                | 5             | 2.9               | 40.3                  | 5.1              | 0.24                | 0.09     | 7.9×        | 2000           | 584                 |
| CANDY WRAPPER       | 7433                | 6             | 7.2               | 21.9                  | 2.6              | 0.38                | 0.11     | 8.4×        | 2500           | 105                 |
| PLASTIC BOTTLE      | 12036               | 6             | 3.5               | 35                    | 6                | 0.47                | 0.06     | 5.8×        | 2100           | 302                 |
| PLASTIC BOTTLE FAST | 12925               | 1.5           | 1.4               | 9.3                   | 1.8              | 0.05                | 0.06     | 5.2×        | 2100           | 123                 |
| ALUMINUM FOIL       | 8452                | 5             | 2.5               | 11.3                  | 1.8              | 0.23                | 0.05     | 6.3×        | 2500           | 136                 |

Table 1: Representative timings and parameters for every example.

| MATERIAL     | Elastic<br>mod.(Pa) | Density<br>(kg/m <sup>3</sup> ) | Thick.<br>(m) | Damping  |         |
|--------------|---------------------|---------------------------------|---------------|----------|---------|
|              |                     |                                 |               | $\alpha$ | $\beta$ |
| ALUM. (CAN)  | 7e10                | 2.8e3                           | 1e-4          | 1e-8     | 200     |
| ALUM. (FOIL) | 7e10                | 2.8e3                           | 1.6e-5        | 1e-8     | 500     |
| PET          | 2.8e                | 1.4e3                           | 1.3e-5        | 3e-7     | 700     |
| CELLOPHANE   | 3e8                 | 1.5e3                           | 2e-5          | 4e-8     | 900     |
| HDPE         | 8e8                 | 9.7e2                           | 2.4e-4        | 5e-7     | 500     |

Table 2: Material parameters used in all our examples.

modifies ARCSim to output the requisite quantities. For damping, we use the Rayleigh model with mass and stiffness-proportional terms controlled by  $\alpha$  and  $\beta$  respectively.

**Buckling detection.** We use a curvature rate metric in order to detect buckling events. Compared to the simple heuristics proposed in [An et al. 2012] where buckling events are detected by checking the sign change of the mean curvature across two consecutive frames, our metric does not miss the buckling events where only one of the two principal curvatures changes without overall sign inversion. We label the vertex  $i$  as a *buckling vertex* when its mean curvature rate  $\frac{dk_i}{dt}$  is above a threshold  $\tau$ . We calibrate  $\tau$  by choosing a value that closely captures all the buckling events detected by the curvature sign change metric. In practice, we estimate  $\frac{dk_i}{dt}$  using finite difference and use  $\tau = 500m^{-1}s^{-1}$  for all our examples. Figure 9 shows that our metric yields a cleaner buckling event detection compared to the sign metric (see orange boxes of Figure 9), while also detecting entire bucklings that did not undergo sign inversion (see green boxes of Figure 9).

**Sound excitation.** The detected buckling events are used to excite the modal vibration (2) of the thin shell. While the shell is deformed by an external load, the work done to strain the shell is stored as elastic energy. When the elastic energy in a local region suffices to overcome the potential barrier to a neighboring energy-minimum state, the shell buckles, and the stored energy is released as vibrations [Kramer and Lobkovsky 1996]. This physical picture motivates us to use an energy-based model to excite buckling sound: We first construct a graph consisting of the detected buckling vertices and their incident edges. We then find all connected component of this graph and consider each component  $\mathbb{C}$  as a *buckling patch* (Figure 3). Suppose that the vibrational energy of  $\mathbb{C}$  is converted from the release of its elastic energy  $E$ , estimated by summing up the stretching and bending energies at every vertex of  $\mathbb{C}$ . We estimate the effective post-buckling velocity  $v_{\mathbb{C}}$  of the patch  $\mathbb{C}$  using  $E = \frac{1}{2} \sum_{i \in \mathbb{C}} m_i v_{\mathbb{C}}^2$  where  $m_i$  is the mass of vertex  $i$ . This effective velocity excites the patch  $\mathbb{C}$  along its averaged surface normal direction  $\mathbf{n}_{\mathbb{C}}$ . This process is also reminiscent of the acoustic emissions from micro-fracturing [Petri et al. 1994; Zheng and James 2010]. Note that when having different crumpling objects in the same scene, using an energy-based excitation naturally takes care of the relative loudness balance between the different objects.

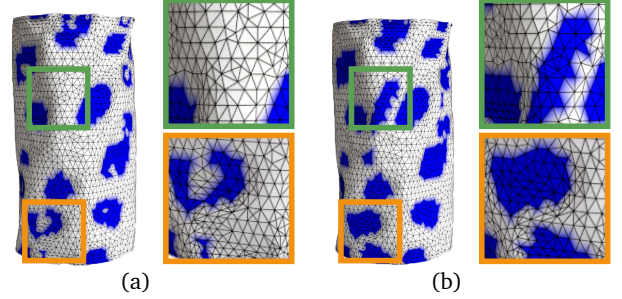


Figure 9: Buckling detection: Using one crumpling frame, we compare (a) the method of An et al. [2012] with (b) our buckling detection method. Every blue region is a detected buckling patch.

## 8 Results

We used our pipeline on a variety of crumpling simulations involving different objects and materials (see Figure 1). All our examples were simulated on a 3.33 GHz Quad-core Intel Core i7-975 CPU with 6GB of memory. The parameter values for the materials used in these examples are listed in Table 2. Representative timings are summarized in Table 1. Please see our accompanying video for all our animation and audio results.

### 8.1 Crumpling examples

We test the wide expressive range of our approach through a set of examples designed around different objects, materials and crumpling motions.

**Soda can.** We simulated a 12cm×6cm aluminum can being rapidly crushed downwards. Notice the initial loud and lower frequency burst, corresponding to the sudden formation of large bucklings across the surface.

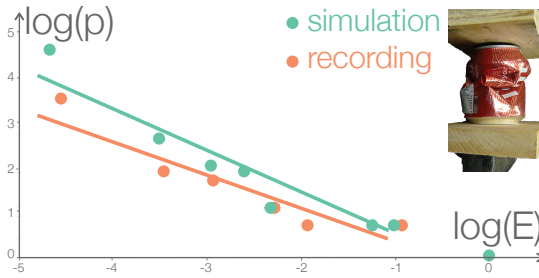
**Plastic bag.** We animate a plastic bag by grabbing it from its rim and executing vertical and lateral motions. The bag is 32cm high, made of PET, and its bottom is fixed to the ground. We are able to synthesize the characteristic and continuous rustling of a plastic bag being manipulated, naturally synchronized with the motion of the bag.

**Candy wrapper.** We slowly wrap and unwrap a cellophane candy wrapper around a candy. The resulting sound is a characteristic sequence of clicky pulses.

**Aluminum foil.** We gently fold the flaps of a 40cm×50cm sheet of aluminum foil wrapped around a box. The sheet produces metallic crinkles as it crumples.

**Plastic bottle.** We crush a small High-density Polyethylene (HDPE) plastic bottle in two different ways: rapidly downwards in a vertical motion, and in a discontinuous sequence of motions that involve vertical as well as lateral deformation. The synthesized sounds are seamlessly synchronized with their respective crumpling motions.





**Figure 10: Power law distributions** for simulated and recorded soda can sounds. Power law coefficients are  $\alpha = 1.02$  and  $\alpha = 0.79$  respectively.

## 8.2 Validation

**Buckling energy distribution.** We compare the energy distributions of buckling events between simulated and recorded sounds. To this end, we recorded the sound emitted by a soda can being slowly crushed by a vice clamp. We then manually identified and extracted each individual buckling event and computed its energy (amplitude squared). Figure 10 shows the log scale plot of recorded (red) and simulated (blue) energy distributions for the soda can. Both plots follow a power law with coefficients of 0.79 and 1.02 respectively. The value discrepancy is possibly due to our manual buckling identification process, or the approximations inherent to a graphics-oriented thin-shell simulator.

**Comparison to real recordings.** We show that our synthesized sounds exhibit similar overall characteristics to their recorded counterparts by looking at their corresponding spectrograms. Figure 11 compares the spectrograms of our simulated sounds (left) and of real recordings (right) for the candy wrapper (top), the aluminum foil (middle), and the plastic bag (bottom). Our simulated sounds convey a similar structural distribution of events as well as similar frequency ranges compared to recordings. This is further illustrated in the supplementary video where the reader can listen to both simulated and recorded sounds.

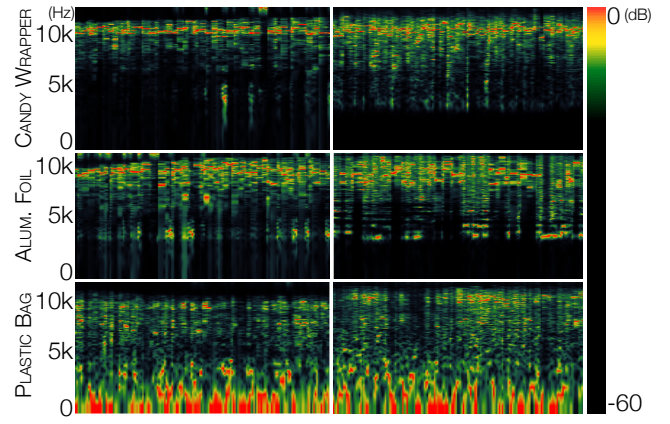
**CMS acceleration.** Table 1 reports the speed-up due to our amortized modal analysis approach for all of our examples. As shown in the supplementary video, these sounds have no perceptually significant difference compared to the brute-force approach.

By adjusting the tolerance  $\epsilon$  of the mean-shift clustering (see [James and Twigg 2005] for details), we can control the aggressiveness of the near-rigid clustering step. A more aggressive clustering usually leads to a higher speedup during the CMS step, but might often be at the expense of sound quality. In the supplementary video, we compare the sounds generated for the aluminum foil using 4 different  $\epsilon$  values: 0.05 ( $6.3\times$  speedup), 0.08 ( $8.3\times$  speedup), 0.13 ( $9.8\times$  speedup) and 0.2 ( $10.1\times$  speedup). The candy wrapper and the plastic bag, two objects that exhibit shorter buckling pulses, seem to support a more aggressive clustering without quality loss, as shown in Table 1.

## 8.3 Perceptual User Study

To evaluate the subjective expressiveness of the sounds generated with our approach, we conducted two user studies. In both studies, we evaluate three conditions:

- (S) sounds synthesized with our approach,
- (R) recorded sounds, and
- (D) sounds synthesized without stochastic enrichment and using the curvature sign buckling criterion [An et al. 2012].



**Figure 11: Spectrograms** of simulated (left) and recorded (right) sounds for the candy wrapper (top), the aluminum foil (middle), and the plastic bag (bottom). Each clip is around 1 second long.

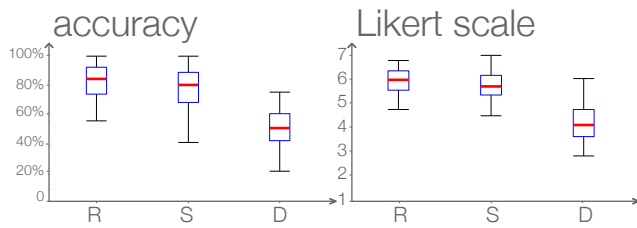
Study A assesses how recognizable the sounds are. We adopted the protocol from [Winnemöller et al. 2006] where recognition rate is also evaluated. Study B seeks to provide further insight by asking for a subjective rating for each sound. We emphasize that the studies do not measure realism, nor correspondence of two sounds to each other. The studies measure human opinion of the correspondence of a sound to an image. Critically, we ensured that subjects were unaware that this study relates to or includes computer-generated sounds. 31 subjects (21 male, 10 female, aged 20 to 39) participated in these studies.

**Materials.** We designed the studies around the five examples presented in Figure 1. For condition R, we recorded the sounds made by the five objects when crumpled. We selected a representative photograph for each of the five objects, which conveyed the correct type of object, material, and crumpling motion. No video was used in this study. We used high quality headphones.

**Procedure.** In study A, each trial presented one sound and the five images with text labels. Subjects were asked to select which image best matched the sound. In study B, each trial consisted of an image and one of the sounds that corresponded to that image. Subjects were asked if the sound they heard corresponded to the object in the image. Subjects answered using a Likert scale (1:Not at all, 7:Definitely yes). In both studies, trial order was randomized and trials were repeated four times. Subjects could listen to the sound in each trial an unlimited number of times.

**Results and Discussion.** Results were analyzed using standard tests well suited for each study, namely a one-way Analysis of Variance (ANOVA) test for study A and a Friedman test for study B, and are reported in Figure 12. In both cases, statistical significance can be claimed if the reported p-value is less than a significance level of 0.05. For Post-hoc analysis (in our case, pairwise comparisons between conditions), a standard approach is to use the Bonferroni method and Wilcoxon signed-rank tests respectively, with a significance level adjusted for multiple comparisons (0.016 since three comparisons are made).

There was a statistically significant difference for both studies (A and B) among the three conditions ( $p = 5e-14$ ,  $p = 2e-10$  respectively). Post-hoc analysis showed as well similar conclusions for both studies: a statistically significant difference between recorded and simplified sound ( $p = 2e-13$ ,  $p = 3e-9$ ), and between simulated and simplified sounds ( $p = 7e-10$ ,  $p = 4e-7$ ). In both studies, there was no statistically significant difference between recorded and simulated sounds ( $p = 0.29$ ,  $p = 0.64$ ).



**Figure 12: User study results.** Boxplots showing for each condition (R,S,D) and each study (A, B) the median, the interquartile range, minimum and maximum values.

These results suggest that, by outperforming the simplified approach, our contributions have a clear impact in the subjective appreciation of sounds by users. More importantly, for these five different examples taken together our approach conveys as much expressiveness as recorded sounds. It is interesting to note that while subjects recognized and rated the recorded plastic bag sound higher than the simulated sound, it is the exact opposite for the soda can, where recognition rate for simulated sounds is almost twice than for the recorded ones.

## 9 Conclusion

We have presented a physically based approach for the synthesis of crumpling sounds from thin shell buckling phenomena, using linear modal analysis of the deformed thin shell to generate the sound of buckling events. For buckling events that cannot be captured by the simulation but that would still produce audible sound, we propose an enrichment algorithm that stochastically instantiates events following a power-law distribution of energies, mimicking real world crumpling behavior. To speed-up the LMA step, and observing that regions of a crumpled object tend to move nearly rigidly over spans of time, we amortize the overall computational cost of LMA by only computing the modes of near-rigid regions once and reusing them many times through Component Mode Synthesis. As a result, we are able to produce fully synthetic high-quality sounds in practical time, seamlessly synchronized with the crumpling motion and conveying as much expressiveness as recorded sounds.

Our approach and implementation are not without limitations, and there are many opportunities for future work. By using linear modal analysis we are limited to modeling phenomena in the linear regime. Yet, thin shells can exhibit mode coupling nonlinear effects [Chadwick et al. 2009]. These nonlinearities could account for the remaining differences between our simulated sounds and their recordings, and it would be interesting to explore the incorporation of mode coupling into our approach while keeping tractable computation times. Friction sounds can also account for an important part of the sound emitted during some crumpling motions. Although friction is less important for our examples, a wider range of motions and materials can be addressed if our method is extended to account for friction.

We chose to use the fixed-interface approach to Component Mode Synthesis, due to its simplicity of implementation and widespread use in engineering. The alternative free-interface method computes the free vibration modes of each component under free boundaries [MacNeal 1971; Rubin 1975]. The resulting modal space is smaller than for the fixed-interface variant, but it requires the computation of additional sets of modes. It would be interesting to evaluate the advantages and disadvantages of the free-interface variant in the context of sound synthesis.

We make the assumption that the same power law distribution ap-

plies throughout the crumpling process. However, since the emergence of the power law is attributed to the quenched curvature disorder—random, stationary network of creases and ridges—it would be interesting to investigate whether a modified stochastic model would better serve the beginning of a crumpling process prior to formation of an extensive crease network.

Sound produced by crumpling can easily be confused with *crumbling* sounds, such as stepping on dry leaves or gravel. Although the source of the sound is not the same, it could be interesting to address crumbling sounds using a similar approach, capturing both visually significant and insignificant sounds.

## Acknowledgments

We thank the anonymous reviewers for their feedback, as well as Anne-Hélène Olivier, Julien Pettré, and Alec Jacobson for insightful discussions, and Breannan Smith for help with the submission. This work was supported in part by the National Science Foundation (CAREER-1453101, IIS-13-19483, IIS-14-09286, IIS-12-08153, and IIS-17257), the Spanish Ministry of Economy (TIN2015-70799-R) and the European Research Council (ERC Starting Grant no. 280135 Animetrics). The work of Gabriel Cirio was supported in part by the Spanish Ministry of Science and Education through a Juan de la Cierva Fellowship, as well as the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 706708. We are grateful for generous support from Pixar, Intel, Disney, Altair, and Adobe. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or others.

## References

- ABOBAKER, M., BOUAZIZ, O., LEBYODKIN, M., LEBEDKINA, T., AND SHASHKOV, I. V. 2015. Avalanche dynamics in crumpled aluminum thin foils. *Scripta Materialia* 99 (Apr.), 17–20.
- AN, S. S., MARSCHNER, S., AND JAMES, D. L. 2012. Motion-driven Concatenative Synthesis of Cloth Sounds. *ACM Trans. Graph.* 31, 4.
- BALANKIN, A. S., CRUZ, M. A. M., CARACHEO, L. A. A., HUERTA, O. S., RIVAS, C. D., MARTÍNEZ, C. L., OCHOA, D. S., RUIZ, L. M., GUTIÉRREZ, S. M., ORTIZ, J. P., AND ORTIZ, M. P. 2015. Mechanical properties and relaxation behavior of crumpled aluminum foils. *Journal of Materials Science* 50, 13.
- BARAFF, D., AND WITKIN, A. 1998. Large steps in cloth simulation. In *Proc. SIGGRAPH ’98*.
- BATHE, K.-J., AND DONG, J. 2014. Component mode synthesis with subspace iterations for controlled accuracy of frequency and mode shape solutions. *Computers & Structures* 139 (July).
- BRIDSON, R., MARINO, S., AND FEDKIW, R. 2003. Simulation of clothing with folds and wrinkles. In *Proc. SCA ’03*.
- CERDA, E., CHAIEB, S., MELO, F., AND MAHADEVAN, L. 1999. Conical dislocations in crumpling. *Nature* 401, 6748 (Sept.), 46–49.
- CHADWICK, J. N., AN, S. S., AND JAMES, D. L. 2009. Harmonic shells: a practical nonlinear sound model for near-rigid thin shells. *ACM Trans. Graph.* 28, 5.
- CHOI, K.-J., AND KO, H.-S. 2002. Stable but Responsive Cloth. In *Proc. SIGGRAPH ’02*.
- COOK, P. R. 2002. *Real Sound Synthesis for Interactive Applications*, 1st edition ed. A K Peters/CRC Press.

- CRAIG, R. R., AND BAMPTON, M. C. C. 1968. Coupling of substructures for dynamic analyses. *AIAA Journal* 6, 7, 1313–1319.
- CRAIG, R. R., AND KURDILA, A. J. 2006. *Fundamentals of Structural Dynamics*, 2 edition ed. Wiley, Hoboken, N.J, July.
- CRAIG, J. R. 2000. Coupling of substructures for dynamic analyses: An overview. In *41st Structures, Structural Dynamics, and Materials Conference and Exhibit*.
- DE KLERK, D., RIXEN, D. J., AND VOORMEEREN, S. N. 2008. General framework for dynamic substructuring. *AIAA Journal* 46, 5, 1169–1181.
- DIDONNA, B. A. 2002. Scaling of the buckling transition of ridges in thin sheets. *Physical Review E* 66, 1 (July), 016601.
- DOEL, K. V. D., KRY, P. G., AND PAI, D. K. 2001. FoleyAutomatic: physically-based sound effects for interactive simulation and animation. In *Proc. SIGGRAPH '01*.
- FELLER, W. 2008. *An introduction to probability theory and its applications*, vol. 2. John Wiley & Sons.
- FONTANA, F., AND BRESIN, R. 2003. Physics-based sound synthesis and control: Crushing, walking and running by crumpling sounds. In *Proc. of the XIV Colloquium on Musical Informatics*.
- GINGOLD, Y., SECORD, A., HAN, J. Y., GRINSPUN, E., AND ZORIN, D. 2004. A Discrete Model for Inelastic Deformation of Thin Shells. Technical Report.
- GOLUB, G. H., AND VAN LOAN, C. F. 2012. *Matrix computations*, vol. 3. JHU Press.
- GRINSPUN, E., HIRANI, A. N., DESBRUN, M., AND SCHRÖDER, P. 2003. Discrete shells. In *Proc. SCA '03*.
- HOLM-JORGENSEN, K., AND NIELSEN, S. R. K. 2009. A component mode synthesis algorithm for multibody dynamics of wind turbines. *Journal of Sound and Vibration* 326, 753–767.
- HOULE, P. A., AND SETHNA, J. P. 1996. Acoustic Emission from crumpling paper. *Physical Review E* 54, 1 (July), 278–283.
- HURTY, W. C. 1965. Dynamic Analysis of Structural Systems Using Component Modes. *AIAA Journal* 3, 4, 678–685.
- JAMES, D. L., AND TWIGG, C. D. 2005. Skinning mesh animations. *ACM Trans. Graph.* 24, 3.
- JAMES, D. L., BARBIC, J., AND PAI, D. K. 2006. Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources. *ACM Trans. Graph.* 25, 3.
- KIM, T., THÜREY, N., JAMES, D. L., AND GROSS, M. H. 2008. Wavelet turbulence for fluid simulation. *ACM Trans. Graph.* 27, 3.
- KRAMER, E. M., AND LOBKOVSKY, A. E. 1996. Universal Power Law in the Noise from a Crumpled Elastic Sheet. *Physical Review E* 53, 2 (Feb.), 1465–1469.
- MACNEAL, R. H. 1971. A hybrid method of component mode synthesis. *Computers & Structures* 1, 4 (Dec.), 581–601.
- MOAKHER, M. 2002. Means and Averaging in the Group of Rotations. *SIAM Journal on Matrix Analysis and Applications* 24.
- NARAIN, R., SAMII, A., AND O'BRIEN, J. F. 2012. Adaptive anisotropic remeshing for cloth simulation. *ACM Trans. Graph.* 31, 6.
- NARAIN, R., PFAFF, T., AND O'BRIEN, J. F. 2013. Folding and Crumpling Adaptive Sheets. *ACM Trans. Graph.* 32, 4.
- O'BRIEN, J. F., SHEN, C., AND GATCHALIAN, C. M. Synthesizing sounds from rigid-body simulations. In *Proc. SCA '02*.
- O'BRIEN, J. F., COOK, P. R., AND ESSL, G. 2001. Synthesizing sounds from physically based motion. In *Proc. SIGGRAPH '01*.
- PANDEY, A., MOULTON, D. E., VELLA, D., AND HOLMES, D. P. 2014. Dynamics of snapping beams and jumping poppers. *EPL (Europhysics Letters)* 105, 2, 24001.
- PARLETT, B. N. 1980. *The symmetric eigenvalue problem*, vol. 7. SIAM.
- PETRI, A., PAPARO, G., VESPIGNANI, A., ALIPPI, A., AND COSTANTINI, M. 1994. Experimental evidence for critical dynamics in microfracturing processes. *Physical Review Letters* 73, 25.
- RAMM, E., AND WALL, W. 2003. *Shell Structures - a Sensitive Interrelation Between Physics and Numerics*. Bericht. SFB 404.
- RUBIN, S. 1975. Improved Component-Mode Representation for Structural Dynamic Analysis. *AIAA Journal* 13, 8, 995–1006.
- RUBINSTEIN, R. Y., AND KROESE, D. P. 2011. *Simulation and the Monte Carlo method*, vol. 707. John Wiley & Sons.
- SCHRECK, C., ROHMER, D., JAMES, D., HAHMANN, S., AND CANI, M.-P. 2016. Real-time sound synthesis for paper material based on geometric analysis. In *Proc. SCA '16*.
- SETHNA, J. P., DAHMEN, K. A., AND MYERS, C. R. 2001. Crackling noise. *Nature* 410, 6825 (Mar.), 242–250.
- SHABANA, A. A. 2012. *Theory of vibration: Volume II: discrete and continuous systems*. Springer Science & Business Media.
- SHEN, L., AND LIU, Y. J. 2006. An adaptive fast multipole boundary element method for three-dimensional acoustic wave problems based on the Burton-Miller formulation. *Computational Mechanics* 40, 3 (Oct.), 461–472.
- STAMMBERGER, M., AND VOSS, H. 2008. Automated Multi-Level Substructuring for a Fluid-Solid Vibration Problem. In *Numerical Mathematics and Advanced Applications*. 563–570.
- SUN, B. H., YEH, K. Y., AND RIMROTT, F. P. J. 1995. On the buckling of structures. *Technische Mechanik* 15, 2, 129–140.
- TERZOPOULOS, D., PLATT, J., BARR, A., AND FLEISCHER, K. 1987. Elastically deformable models. In *Proc. SIGGRAPH '87*.
- VLIEGENTHART, G. A., AND GOMPPER, G. 2006. Forced crumpling of self-avoiding elastic sheets. *Nature Materials* 5, 3, 216–221.
- WINNEMÖLLER, H., OLSEN, S. C., AND GOOCH, B. 2006. Real-time video abstraction. *ACM Trans. Graph.* 25, 3.
- WOOD, A. 2002. Witten's lectures on crumpling. *Physica A: Statistical Mechanics and its Applications* 313, 1.
- YANG, Y., XU, W., GUO, X., ZHOU, K., AND GUO, B. 2013. Boundary-Aware Multidomain Subspace Deformation. *IEEE TVCG* 19, 10.
- ZHENG, C., AND JAMES, D. L. 2010. Rigid-body fracture sound with precomputed soundbanks. *ACM Trans. Graph.* 29, 4.
- ZHENG, C., AND JAMES, D. L. 2011. Toward high-quality modal contact sound. *ACM Trans. Graph.* 30, 4.
- ZHOU, C., JIN, X., AND WANG, C. C. L. 2008. Shear buckling and dynamic bending in cloth simulation. *Computer Animation and Virtual Worlds* 19, 3-4 (Jan.), 493–503.