Bounce Maps: An Improved Restitution Model for Real-Time Rigid-Body Impact

JUI-HSIEN WANG, Stanford University
RAJSEKHAR SETALURI, Stanford University
DOUG L. JAMES, Stanford University
DINESH K. PAI, University of British Columbia

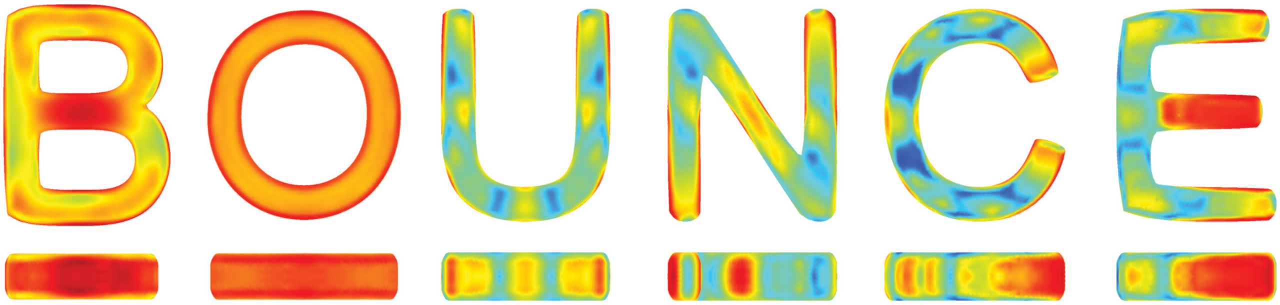


Fig. 1. **The coefficient of restitution is not constant!** Numerically computed coefficients of restitution ($blue = 0 \rightarrow red = 1$) are shown for these letters (with bottom view) and reveal significant spatial variations arising from micro-collision phenomena. Using a fast restitution analysis preprocess, we encode restitution values in Bounce Maps for fast runtime lookup during rigid-body simulation, thereby capturing natural variability in contact responses (see Figure 2). (All objects have the same physical material parameters ("steel"), and use #modes=45.)

We present a novel method to enrich standard rigid-body impact models with a spatially varying coefficient of restitution map, or *Bounce Map*. Even state-of-the art methods in computer graphics assume that for a single rigid body, post- and pre-impact dynamics are related with a single global, constant, namely the *coefficient of restitution*. We first demonstrate that this assumption is highly inaccurate, even for simple objects. We then present a technique to efficiently and automatically generate a function which maps locations on the object's surface along with impact normals, to a scalar coefficient of restitution value. Furthermore, we propose a method for two-body restitution analysis, and, based on numerical experiments, estimate a practical model for combining one-body Bounce Map values to approximate the two-body coefficient of restitution. We show that our method not only improves accuracy, but also enables visually richer rigid-body simulations.

CCS Concepts: • **Computing methodologies → Animation**; **Physical simulation**;

Additional Key Words and Phrases: Computer animation, rigid body, collision, contact, impact, chatter, coefficient of restitution, Newton's law of restitution, bounce, modal vibration.

ACM Reference format:
Jui-Hsien Wang, Rajsekhar Setaluri, Doug L. James, and Dinesh K. Pai. 2017. Bounce Maps: An Improved Restitution Model for Real-Time Rigid-Body Impact. *ACM Trans. Graph.* 36, 4, Article 150 (July 2017), 12 pages.
https://doi.org/10.1145/3072959.3073634

1 INTRODUCTION

Rigid-body impact and contact are of great importance to computer graphics. As in [Smith et al. 2012], we define contact to include resting or sliding contact, whereas impact captures instantaneous and transient collisions. Much work has been done in computer graphics to model contact and impact scenarios, as well as to develop efficient and robust techniques to handle complex collision scenarios; see Sec. 2 for a brief review.

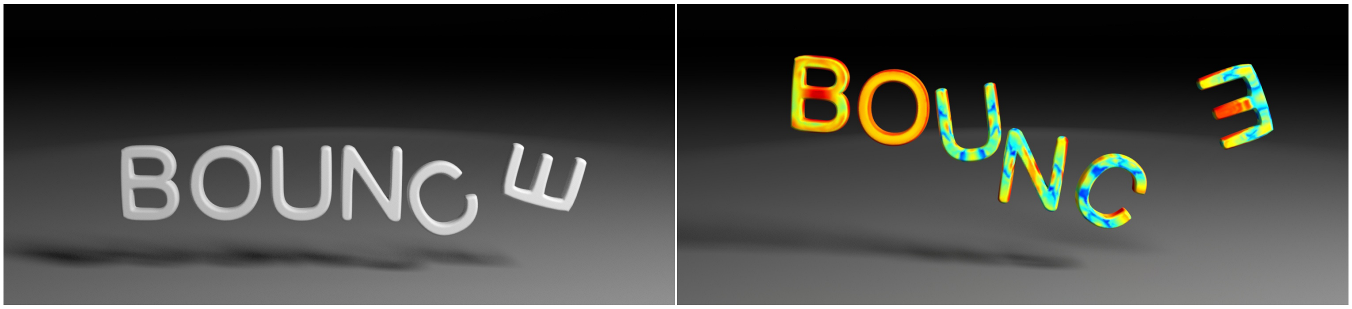
Almost all previous work in graphics and mechanics has focused on algebraic collision laws, which *assume* a scalar coefficient of restitution, often treated as a material or object property. Indeed, most use the simplest and oldest model, Newton's Law of Restitution. Newton introduced this concept in his Principia, arguing that collision dynamics follow a simple law: the post-impact relative normal velocity v_n^+ of two objects is proportional to the pre-impact relative normal velocity v_n^-, where the ratio

$$\varepsilon := -\frac{v_n^+}{v_n^-} \in [0, 1], \tag{1}$$

is a measurable and material-dependent constant known as the *coefficient of restitution*. Thanks, in part, to Baraff's [1997] influential course notes that used ε to determine the contact impulse, this model is widely used in computer graphics.

The biggest attraction of a restitution model is, of course, its utter simplicity and efficiency. However, it has long been recognized that "rigid body" and "impact" are essentially contradictory, a meeting of an undeformable object with an impenetrable one. Such a modeling assumption would be ok (after all, many of the governing equations of classical mechanics are centuries old) if it incurred only a small or localized error, but the errors can be huge. For instance, Stoianovici

0730-0301/2017/7-ART150 $15.00
https://doi.org/10.1145/3072959.3073634



Constant ε =0.6 (global average)

Bounce-Mapped ε

Fig. 2. **"BOUNCE Drop" animation** frames after the letters bounce off the floor and rebound. (Left) Constant ε letters lack variety and excitement, whereas (Right) a greater variety of behaviors result from Bounce-Mapped letters, with far bouncier "B," "O" and "E" bottom impacts.

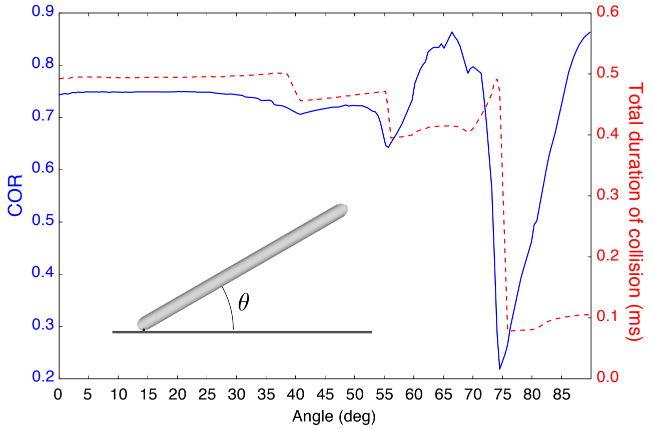


Fig. 3. **Steel rod dropped at different angles** (20 cm, steel rod): Bounce mapping reveals a flat ε response (in blue) at shallow angles, then a dramatic dip at steep angles (frictionless contact, undamped rod, $m = 140$ modes), similar to experiments performed by [Stoianovici and Hurmuzlu 1996]. These abrupt changes are often related to contact duration (in red).

and Hurmuzlu [1996] dropped steel bars on a foundation, and found that for a single bar, ε could vary between 0.1 and 0.9 as the drop-angle changes (80% of the valid range [0, 1])!

Therefore, we argue that the standard practice of using a single parameter to model rigid-body impact is a fundamental limitation. Consequently we propose to extend the desiderata of [Smith et al. 2012] to include: **(COR) Coefficient of Restitution**, which states that simulated solids should bounce like their physical counterparts. Before proposing a solution, however, it is important to inspect why exactly this limitation arises.

Why does the coefficient of restitution vary? When measured in physical experiments, ε has been shown to vary with a body's shape, material properties, as well as the velocity, location and contact normal of impact. There are several contributing factors, including phenomena such as plasticity and local conforming (non-rigid) contact; however, we argue that the most important factor is the variation in energy loss to elastic vibrations. This was previously observed by [Goldsmith 1960] and confirmed by the analysis of [Stoianovici and Hurmuzlu 1996]. They modeled the rod using only a handful of (less than 10) visco-elastically-linked segments and were able to match the significant variation in the coefficient of restitution apparent in the experimental data. They concluded that *"the mechanism that leads to the significant variations in the coefficient of restitution is mainly due to the residual energy that remains internally in the bar when the contact is lost."* Explicitly modeling stiff elastodynamics, therefore, provides greater accuracy, but, unfortunately, it is not well suited to the complex shapes and real-time constraints of many graphics applications.

In some scenarios, the detailed variation of ε over the surface of an object may not be perceptible [O'Sullivan et al. 2003], and it may be possible to get away with a constant COR or even with using stochastic models. Typical examples are background simulations, such as destruction of buildings. Indeed, there is a long tradition in computer animation of slightly perturbing contact impulses to add visually interesting variability to otherwise dull constant-ε animations [Barzel et al. 1996], and to enable simulation control [Chenney and Forsyth 2000; Popović et al. 2000; Twigg and James 2007]. However, there are many other scenarios in which capturing the variation is essential. When observing a single "hero" object over multiple bounces, e.g., a baseball bat bouncing off the floor (see accompanying video), the difference between the bouncy knob and the duller barrel is very obvious. Representing variation of ε is also essential when the simulation has to be realistic and not just plausible. For example, to train for a sport such as hockey using a VR simulation, efficiently capturing phenomena such as the well known "sweet spot" is essential (see Figure 15).

Contributions: We propose to achieve **(COR)** by precomputing the dynamics that arise when we model a rigid object as an extremely stiff deformable object. We propose a fast precomputation pipeline to estimate the effective coefficient of restitution using modal analysis. We store the spatially (and sometimes directionally) varying coefficient of restitution in a *Bounce Map* (see Figure 1). This precomputation account for both the energy loss due to residual vibrations and multiple micro-collisions known to occur within a single contact event. Our spatially varying *Bounce Map* can be easily integrated with existing rigid-body simulators, with negligible run-time costs. As Bounce Maps are inherently a single-body response, we devise extensions for the two-body case, including a contact solver, and experimentally justified methods for combining one-body Bounce Map coefficients for cheap runtime approximations. Thus we achieve the simplicity and efficiency of a restitution model, while achieving the accuracy of solving the full elastic impact problem. See Figures 1, 2, 3, 4 and 5 for illustrative examples.

2 RELATED WORK

Rigid-body models are widely used in computer graphics; see [Bender et al. 2014] for a recent survey. The smooth motion of a rigid body is easy to simulate; the biggest challenge has been dealing with collisions. This requires both collision *detection* and collision *response*. Our focus is collision response; collision detection is a large and active research area which is orthogonal to our focus.

Following Newton's law of restitution, different variants have been proposed to address some of its limitations. Poisson's hypothesis posits that impact events are comprised of a compression and restitution phase, and that the ratio of the impulses accumulated during these phases *is constant*. Stronge's more recent hypothesis argues that (the square root of) the ratio of kinetic energies recovered and released in the restitution and compression phases *is constant*. Though these three models each have their own benefits and drawbacks, they all assume that impact dynamics are captured by a single scalar parameter (the three models are equivalent in certain circumstances [Stronge 2004]). Furthermore, there is no well-accepted set of values for various materials, and there is no clear or simple method to experimentally measure this value. Ultimately, the coefficient of restitution, in any of its forms, is a gross simplification of the complex dynamics that occur when two objects collide.

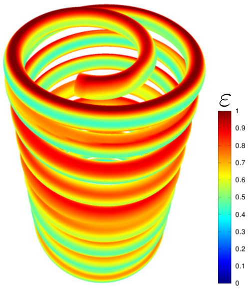


Fig. 4. **A bounce-mapped spring** bounces highest on the (dark red) end faces, where it can store and recover spring energy during longer impacts. In contrast, side impacts (in green) tend to bounce much less, losing energy to vibrations. In fact, the simulated range of ε for this example is effectively [0, 1].

There has been much recent progress towards making rigid-body impact at least consistent with known physical laws, such as linear and angular momentum conservation (e.g., see the monographs [Brogliato 2012; Goldsmith 1960; Pfeiffer and Glocker 2008; Stewart 2011; Stronge 2004]). For instance, Chatterjee and Ruina [1998] derived constraints on reasonable coeffients of restitution. The use of measure differential inclusions to model impact has clarified many of the paradoxes of rigid-body impact [Brogliato 2012; Stewart 2011]. Other important avenues include simultaneous impact (e.g., [Smith et al. 2012]), and friction (which has a rich history [Painlevé 1895]) and recent progress (e.g., [Mirtich and Canny 1995; Wang and Mason 1987] and the work cited above).

Despite this progress on rigid-body impact, it has also been well known that basic conservation laws are not sufficient to determine the value of ε. The energy losses (and hence ε) depend on many factors, but the main factor for moderate speed impacts (without plastic work and conforming contact) appears to be the energy loss due to residual vibrations upon separation. This was observed by [Goldsmith 1960] who examined the energy transferred to vibrations using the analytical methods available at that time. Elastic vibrations also result in multiple micro-collisions within a single macrospcopic impact event, calling into question the usual analysis based on locating the maximum compression. These phenomena were also documented in the seminal work of [Stoianovici and Hurmuzlu 1996] described above.

In early work, Ullrich and Pai [1998; 1999] proposed precomputing a "contact response map" on the surface of an object, but chose to represent the time-domain force response due to vibration at each location. This is both hard to compute reliably and requires high-resolution convolution to obtain the post-impact behavior. By contrast, our *Bounce Maps* directly represent the widely used coefficient of restitution, can be computed efficiently using modal models, and bake-in the effects of residual vibrations and multiple micro-collisions. *Bounce Maps* can be dropped in to work with existing rigid-body simulation code, replacing a scalar with an efficient function call (essentially a table lookup).

3 FAST RESTITUTION ANALYSIS

We describe a fast modal contact solver for a single point-like contact, and show how it can be used to efficiently estimate COR at a specific contact point and normal.

3.1 Background: Modal dynamics model

We simulate a rigid-body model augmented with linear vibration modes, as in commonly done in multibody dynamics simulation (e.g., [Shabana 2012, 2013]), and in computer animation and sound synthesis (e.g., [Kaufman et al. 2008; Zheng and James 2011]); please see these references for implementation details. For contact restitution modeling, modal vibration models have two major benefits: (1) they provide an efficient way to model the vibrational energy loss during impact events using a small number of dominant eigenmodes, and (2) their smooth spatial response permits a simplified point-like contact analysis.

Regarding notation, let q represent the generalized coordinates of the body, including its center of mass $\boldsymbol{c}$, orientation $\boldsymbol{R}$, and modal coordinates, $\boldsymbol{q} \in \mathbb{R}^m$, so that $\mathrm{q} = (\boldsymbol{c}^T, \boldsymbol{R}^T, \boldsymbol{q}^T)^T$. Let the generalized velocity be $\dot{\mathrm{q}} = (\dot{\boldsymbol{c}}^T, \boldsymbol{w}^T, \dot{\boldsymbol{q}}^T)^T \in \mathbb{R}^{m+6}$. In body coordinates, let the center of mass be located at the origin; let $\boldsymbol{X}$ denote an undeformed material point, and $\boldsymbol{N}$ an undeformed surface normal. In world coordinates, the corresponding deformed point is

$$\boldsymbol{x} = \boldsymbol{\phi}(\boldsymbol{X}, \mathrm{q}(t)) = \boldsymbol{c}(t) + \boldsymbol{R}(t)(\boldsymbol{X} + \boldsymbol{U}(\boldsymbol{X})\boldsymbol{q}(t)), \tag{2}$$

and its velocity is

$$\begin{aligned} \boldsymbol{v} &= \dot{\boldsymbol{c}} + \boldsymbol{R}[\boldsymbol{w}]_\times(\boldsymbol{X} + \boldsymbol{U}\boldsymbol{q}) + \boldsymbol{R}\boldsymbol{U}\dot{\boldsymbol{q}} && (3) \\ &= \dot{\boldsymbol{c}} - \boldsymbol{R}[\boldsymbol{X} + \boldsymbol{U}\boldsymbol{q}]_\times \boldsymbol{w} + \boldsymbol{R}\boldsymbol{U}\dot{\boldsymbol{q}} && (4) \\ &= \left[\ \boldsymbol{I}_3 \ \vdots \ \text{-}\boldsymbol{R}[\boldsymbol{X} + \boldsymbol{U}\boldsymbol{q}]_\times \ \vdots \ \boldsymbol{R}\boldsymbol{U}\ \right] \dot{\mathrm{q}} = \boldsymbol{J}\dot{\mathrm{q}}, && (5) \end{aligned}$$

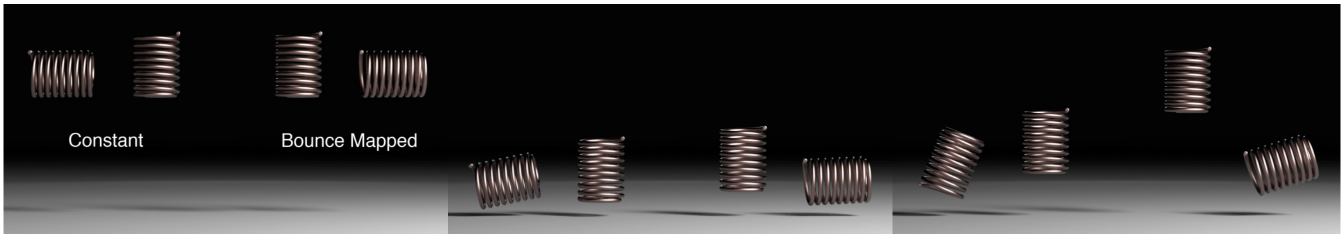


Fig. 5. **Springs of comparison:** (Left) Four springs are dropped from a similar height, in vertical and horizontal orientations. The two on the right are Bounce Mapped, and the left two springs have constant $\varepsilon = 0.6$ (the average map) values. (Middle) Springs immediately after impact. (Right) Near their maximum heights, the left springs exhibit similar bounce, whereas the two right Bounce-Mapped springs have far more dramatic responses: the vertical spring rebounds to nearly the same height, whereas the horizontal barely gets off the ground.

where the Jacobian is $\boldsymbol{J} = \boldsymbol{J}(\boldsymbol{X}, \mathrm{q}) \in \mathbb{R}^{3\times(m+6)}$, and $[\boldsymbol{v}]_\times$ denotes the skew-symmetric cross-product matrix for "$\boldsymbol{v}\times$." Let

$$\mathrm{M} = \begin{bmatrix} M_{tot}\boldsymbol{I}_3 & 0 & 0 \\ 0 & \mathbb{I} & 0 \\ 0 & 0 & \boldsymbol{I}_m \end{bmatrix} \in \mathbb{R}^{(m+6)\times(m+6)}, \tag{6}$$

denote the diagonal mass matrix in body coordinates[1]. We simulate the body without gravity, and will neglect internal damping forces initially. Therefore, in the absence of contact forces, we only integrate the internal forces, $\mathrm{f}_{int}(\mathrm{q}, \dot{\mathrm{q}})$: the quadratic velocity vector [Shabana 2013], and the modal oscillator force, $\ddot{q}_i = -\omega_i^2 q_i$.

3.2 Proxy Contact Problem

For efficiency and general evaluation, we analyze restitution response using a proxy point-plane contact model (see Figure 6). We can assume that the world and body frames are initially aligned, and that the body has no initial deformation, $\mathrm{q}^- = (0, \boldsymbol{I}, 0)^T$. Given an input surface point and normal, $(\boldsymbol{X}, \boldsymbol{N})$, in body coordinates, we define a virtual contact plane (fixed in world space) passing through $\boldsymbol{X}$ with unit normal $\boldsymbol{n} = -\boldsymbol{N}$, such that the deformed point $\boldsymbol{x}$ must always satisfy

$$C(\boldsymbol{x}) = \boldsymbol{n}^T(\boldsymbol{x} - \boldsymbol{X}) \geq 0. \tag{7}$$

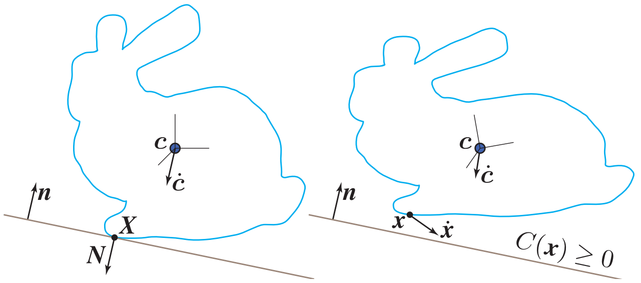


Fig. 6. **Proxy contact problem:** (Left) Initial undeformed configuration for a planar restitution analysis at $(\boldsymbol{X}, \boldsymbol{N})$ with body translating into contact at velocity $v_n^- \boldsymbol{N}$; (Right) During simulation with collisions resolved between the deformed contact point $\boldsymbol{x}$ and the planar constraint $C(\boldsymbol{x}) \geq 0$.

Without loss of generality, we initialize the impact analysis with a velocity compatible with $v_n^- = -1\ m/s$, and choose a pure translational velocity, $\dot{\mathrm{q}}^- = (v_n^- \boldsymbol{n}, 0, 0)^T$. We then consider the multibody contact dynamics problem involving the deformable model and the perfectly rigid plane constraint, $C(\boldsymbol{x}) \geq 0$, where *only* the point $\boldsymbol{x}(t)$ is checked for collisions. This allows us to ensure that collision events are local, and also separates the object's geometry from the contact problem, something that is an issue for classical analyses that consider sphere or plane contact. Frictionless point-plane contacts are resolved using impulses, the details of which are described in §3.3.

Stoianovici and Hurmuzlu [1996] note that multiple *micro-collision* events can happen over the course of a single impact (due to elastic waves) and that such micro-collisions play a fundamental role in the variation of ε. We also observed this phenomenon in our simulation experiments. Therefore, we timestep the simulation until we can ensure that no more point-plane contacts can occur, which we achieve using a spherical deformation bound (in §3.4).

Finally, to estimate the coefficient of restitution of this sequence of impact events, we consider the state of the body at the time of the last micro-collision, after which $\boldsymbol{x}$ is guaranteed to separate from the plane and the contact event is over. To estimate the post-impact normal velocity, v_n^+ for use in the restitution formula, we can not use the contact point's value $\boldsymbol{v}\cdot\boldsymbol{n}$ directly, since it is polluted with high-frequency modal oscillations. Therefore, we define its rigid velocity component $\dot{\mathrm{q}}_{rigid}$ by projecting out the deformation part, $\dot{\mathrm{q}}_{rigid} = \mathcal{R}\dot{\mathrm{q}} = (\dot{\boldsymbol{c}}^T, \boldsymbol{w}^T, \boldsymbol{0}^T)^T$. Our point's rigid post-impact velocity can then be robustly estimated as $\boldsymbol{v}_{rigid} = \boldsymbol{J}\dot{\mathrm{q}}_{rigid}$, and therefore our post-impact normal velocity estimate is $v_n^+ = \boldsymbol{n}^T \boldsymbol{J}\,\dot{\mathrm{q}}_{rigid}$ immediately following the final contact event. The total contact time, τ, and the number of contiguous contact events can also be computed.

3.3 Frictionless Point-Contact Solver

Regarding the single-point frictionless contact problem, we use the predictor-corrector scheme of [Kaufman et al. 2008] to advance the velocity from $\dot{\mathrm{q}}^t$ to $\dot{\mathrm{q}}^{t+1}$. The predictor simply integrates the internal forces ignoring contact, $\dot{\mathrm{q}}^p = \dot{\mathrm{q}}^t + \Delta t \mathrm{M}^{-1}\mathrm{f}_{int}$. Next, if the point is in contact ($C(\boldsymbol{x}) \leq 0$) and not separating ($\boldsymbol{v}^T\boldsymbol{n} < 0$) we apply an inelastic impulse $\lambda\boldsymbol{n}$ to resolve contact. The final generalized velocity is $\dot{\mathrm{q}}^{t+1} = \dot{\mathrm{q}}^p + \lambda \mathrm{M}^{-1}\mathrm{n}$ where $\mathrm{n} = \boldsymbol{J}^T\boldsymbol{n}$ is the generalized contact normal. The Signorini-Fischera condition requires that $0 \leq \mathrm{n}^T\dot{\mathrm{q}}^{t+1} \perp \lambda \geq 0$, from which it follows that, when the point is not already separating

[1] Here the inertia tensor $\mathbb{I}$ is diagonal (due to body-frame alignment with the principal axes of inertia), off-diagonal M terms disappear for small deformations and center-of-mass $\boldsymbol{c}$ at origin, and mass-normalized eigenmodes $\boldsymbol{U}$ make the modal mass matrix the identity, $\boldsymbol{I}_m$.

the contact impulse is $\lambda = -\mathrm{n}^T \dot{\mathrm{q}}^p / \mathrm{n}^T \mathrm{M}^{-1}\mathrm{n}$. Since the mass matrix is diagonal, this frictionless contact solve/integration step takes $O(m)$ flops for m modes.

3.4 Collision bounds for estimating contact duration

Given the possibility of multiple vibration-induced contact events, we need a way to conservatively estimate the minimum integration time for each contact simulation. When the proxy point is not in contact with the plane, we estimate whether or not the rigidly transforming and vibrating point can still return to contact as follows. Given the instantaneous value of each mode's energy, we can conservatively bound the point's future vibrations using a spherical deformation bound, analogous to [James and Pai 2004]. Specifically, given the i^{th} mode's position q_i and velocity $\dot{q}_i$ coordinates, we compute the mode's instantaneous energy, $E_i = \frac{1}{2}(\dot{q}_i^2 + \omega_i^2 q_i^2)$. Since energy is conserved in the absence of contact, we can bound the absolute amplitude at future contact-free times by $\max_t |q_i(t)| = \sqrt{2E_i}/\omega_i$. We can then estimate a conservative bounding sphere radius by

$$R(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \sum_{i=1}^{m} \|\boldsymbol{U}_i\| \frac{\sqrt{2E_i}}{\omega_i} = \sum_{i=1}^{m} \Delta R_i \sqrt{E_i} \quad (8)$$

(where $\Delta R_i \equiv \sqrt{2}\|\boldsymbol{U}_i\|/\omega_i$ are cached values). This bound ensures that the body-frame vertex displacement $\boldsymbol{u} = \boldsymbol{U}\boldsymbol{q}$ satisfies $\|\boldsymbol{u}\| = \|\boldsymbol{x} - \boldsymbol{x}_{rigid}\| \leq R(\boldsymbol{q}, \dot{\boldsymbol{q}})$ forward in time, where $\boldsymbol{x}_{rigid} = \boldsymbol{c} + \boldsymbol{R}\boldsymbol{X}$ is the rigidly transformed point. Then given the bounding sphere affixed to the body's underlying rigid-body frame at $\boldsymbol{X}$, we can terminate simulation when the bounding sphere is no longer in contact, $C(\boldsymbol{x}_{rigid}) > R_{bound}$ (see Figure 7).

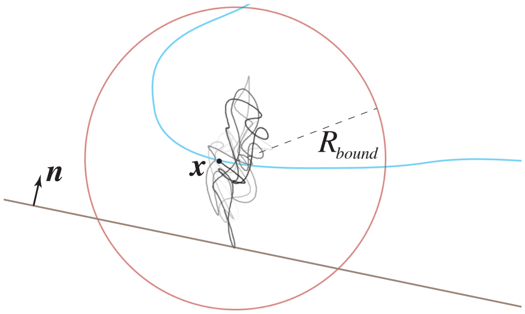


Fig. 7. **Bounded deformation collision model:** We conservatively bound the future oscillations of the point $\boldsymbol{x}(t)$ about $\boldsymbol{x}_{rigid}(t)$, and integrate dynamics until the rigidly transforming spherical bound no longer touches the plane, $C(\boldsymbol{x}_{rigid}) > R_{bound}$.

Discussion: While it is possible for the rigid-body trajectory to produce future point-plane contacts, e.g., it may spin around and hit again, we do not consider these subsequent impacts following large motions to be part of the restitution analysis. Another issue is that modal energy loss/gain due to numerical integration can lead to violation of our energy bound, and thus produce collision bound inaccuracies. However, these are minimized by our use of small timesteps and the symplectic Euler integrator, not to mention the highly conservative nature of the collision bound.

3.5 Fast Restitution Analysis Algorithm

Our method for efficiently estimating the COR for a given $(\boldsymbol{X}, \boldsymbol{N})$ is summarized in Algorithm 1. Note that quantities such as $\boldsymbol{x}$, $\boldsymbol{v}$, $\boldsymbol{J}$, $\dot{\mathrm{q}}_{rigid}$, E_i, are always evaluated using the current $(\mathrm{q}, \dot{\mathrm{q}})$ state. The timestep size Δt is selected to adequately resolve the highest-frequency modal oscillation. A representative restitution analysis time-series of $C(x)$ point-plane distance values, and v_n^{rigid} normal velocities are shown in Figure 8.

Algorithm 1: Fast Restitution Analysis

```
Function computeEpsilon()
Input: $\boldsymbol{X}$, $\boldsymbol{N}$, $\boldsymbol{U}$, $\omega$, M
/* Initialize in contact:                                                              */
$v_n^- = -1\,\frac{m}{s},\ v_n^+ = 0,\ \boldsymbol{n} = -\boldsymbol{N},\ \mathrm{q} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{I} \\ \boldsymbol{0} \end{pmatrix},\ \dot{\mathrm{q}} = \begin{pmatrix} v_n^- \boldsymbol{n} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{pmatrix}$
$R_{bound} = 1, \quad \Delta R_i = \sqrt{2}\|\boldsymbol{U}_i\|_2/\omega_i,\ i = 1 \ldots m.$
/* Time-step collisions while BD-bound overlaps:                                       */
while $C(\boldsymbol{x}_{rigid}) < R_{bound}$ do /* potentially in contact                         */
    $\dot{\mathrm{q}}$ += $\Delta t\, \mathrm{M}^{-1} \mathrm{f}_{int}(\mathrm{q}, \dot{\mathrm{q}})$
    if $C(\boldsymbol{x}) \leq 0$ then /* in contact                                                */
        if $\boldsymbol{v} \cdot \boldsymbol{n} < 0$ then /* not separating                                    */
            $\lambda \ = -\boldsymbol{v} \cdot \boldsymbol{n}\ /\ \mathrm{n}^T \mathrm{M}^{-1} \mathrm{n}$                   /* Compute impulse */
            $\dot{\mathrm{q}}$ += $\lambda\, \mathrm{M}^{-1} \mathrm{n}$                          /* Apply contact impulse */
            $v_n^+ \ = \boldsymbol{n}^T \boldsymbol{J}\, \dot{\mathrm{q}}_{rigid}$                    /* Cache post-impact speed */
            $R_{bound} = \sum_{i=1}^{m} \Delta R_i \sqrt{E_i}$                  /* Update BD bound */
    $\mathrm{q} \leftarrow$ updatePosition($\mathrm{q}$, $\dot{\mathrm{q}}$, $\Delta t$)
return $\varepsilon = -v_n^+ / v_n^-$
```

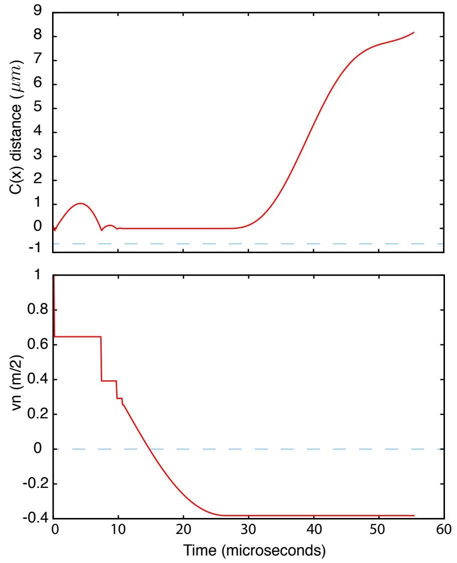


Fig. 8. **Restitution analysis time-series** reveal micro-collision events: (Top) $C(\boldsymbol{x})$ normal distance; (Bottom) v_n^{rigid} rigid-component normal velocities. Dynamics are integrated until the bounded-deformation separation distance is achieved.

4 SAMPLING AND RUNTIME EVALUATION

Now that we can perform one-body restitution analyses to evaluate, $\varepsilon(\boldsymbol{X}, \boldsymbol{N})$, for a specified contact point $\boldsymbol{X}$ and normal $\boldsymbol{N}$, we briefly consider how to sample ε over an object's surface.

Smooth surfaces: When the normal is uniquely defined everywhere as a function of $\boldsymbol{X}$, so that $\boldsymbol{N} = \boldsymbol{N}(\boldsymbol{X})$, we can simply tabulate restitution as a spatially varying function on the surface, $\varepsilon = \varepsilon(\boldsymbol{X})$. Far from being a global constant, $\varepsilon(\boldsymbol{X})$ is spatially varying, and we find that it exhibits highly nonsmooth behavior even on flat surfaces (see Figure 9).

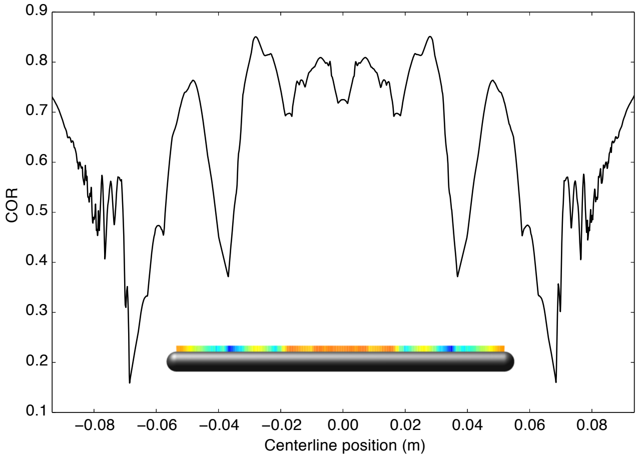


Fig. 9. **Bounce Maps are not smooth:** A high-res sampling of $\varepsilon(\boldsymbol{X})$ values on the side ($\boldsymbol{N} = const$) of a 20 *cm* steel rod (shown inset) reveals a highly nonsmooth function due to complex micro-collisions in the restitution analysis ($m = 140$ modes).

Without loss of generality, we densely sample ε at a suitably fine mesh resolution, sampling $\boldsymbol{X}$ at triangle centroids with $\boldsymbol{N}$ the triangle normal. At runtime, we use the piecewise constant ε value on the contact triangle, or locally interpolate to a vertex. Even smooth surfaces can exhibit significant ε variations due to highly varying normals (see Figure 10).

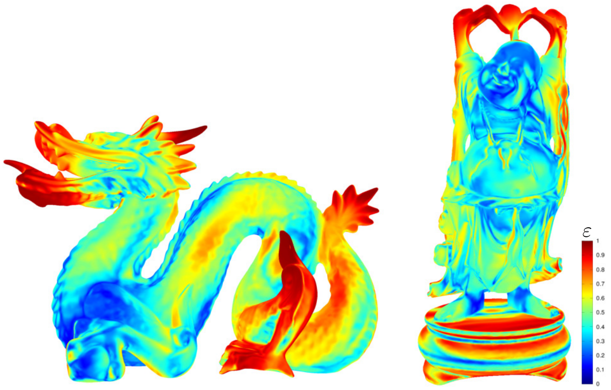


Fig. 10. **Strong normal dependence** of restitution values are demonstrated by bounce-mapped Dragon and Buddha models.

Non-smooth surfaces, such as a cube, complicate sampling because potential contact normals are not uniquely defined everywhere, and ε is strongly dependent on the direction of the contact normal (see Figure 11). The possible contact normals at a point $\boldsymbol{X}$ lie in a cone, $\boldsymbol{N}(\boldsymbol{X}) \in \mathcal{N}(\boldsymbol{X})$ that defines a family of possible contact planes. For non-smooth surfaces, we must therefore parameterize restitution using both contact position and normal variables, $\varepsilon = \varepsilon(\boldsymbol{X}, \boldsymbol{N})$. We generate samples of this function on triangle meshes by sampling three separate subspaces (illustrated in Figure 12)

(1) *Faces* are sampled using the single normal (as with smooth surfaces),
(2) *Edges* are sampled at multiple directions by interpolating edge-flap normals,
(3) *Vertices* are sampled at multiple directions that lie inside the cone of possible contact normals.

In our examples, we interpolated edge samples at 9 degree intervals, and vertex samples were selected from 2562 uniformly sampled spherical directions restricted to the contact cone.

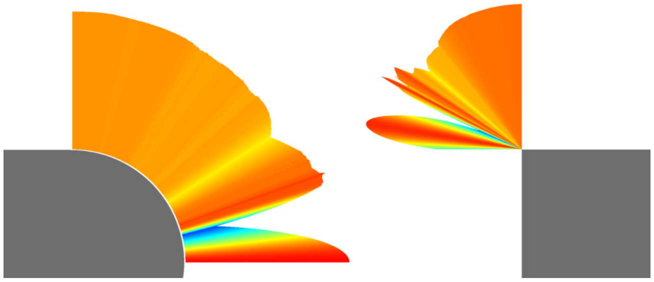


Fig. 11. **Sampling normal variations:** Restitution values can be strongly dependent on $\boldsymbol{N}$, and vary rapidly at sharp features, as demonstrated here for two steel bars (20 cm long) with different end caps. Normal-dependent ε variations occur (Left) spatially along the smooth spherical end cap, and (Right) at different normals for a single nonsmooth-edge vertex on the rectangular bar. Notice the pronounced (blue) dip in ε values in both cases.

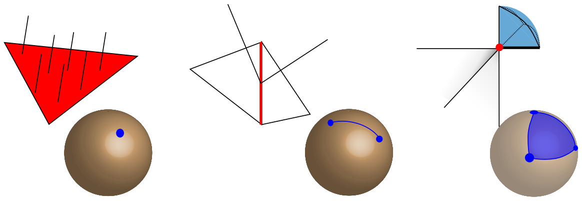


Fig. 12. **Sampling nonsmooth mesh features:** (Left) *faces* are sampled using a single normal, (Middle) *sharp edges* are sampled using interpolated edge-flap normals, and (Right) *sharp vertices* are sampled using normals inside the cone of possible contact normals.

Runtime Lookup: Given a runtime query for $\varepsilon(\boldsymbol{X}, \boldsymbol{N})$ on a non-smooth mesh, we require a way to identify, and potentially combine, nearby samples. We use a simple Euclidean distance metric between normals by exploiting the fact that unit normals differing by an angle θ satisfy

$$\|\boldsymbol{N} - \boldsymbol{N}'\|_2 = 2\sin(\theta/2) \overset{\theta\to 0}{\approx} \theta, \tag{9}$$

and so provides a useful measure of angular variation for nearby normals. We measure joint position/normal distance using

$$\text{dist}^2((\boldsymbol{X},\boldsymbol{N}),(\boldsymbol{X}',\boldsymbol{N}')) = \|\boldsymbol{X}-\boldsymbol{X}'\|_2^2 + \gamma^2\|\boldsymbol{N}-\boldsymbol{N}'\|_2^2, \tag{10}$$

where $\gamma = h_x/h_\theta$, and h_x is the sampling length scale compatible with the angular length scale, h_θ, used during sampling. Then, given a runtime query for $\varepsilon(\boldsymbol{X},\boldsymbol{N})$ on a nonsmooth mesh, we first identify all samples near $\boldsymbol{X}$ (in a geodesic sense) using the mesh data structure. We can then generate a piecewise constant approximation to ε using the value associated with the nearest neighbor using (10). Alternately we can generate a linear-degree moving least squares (MLS) approximation [Jin et al. 2009] to more smoothly interpolate ε values in $(\boldsymbol{X},\boldsymbol{N})$ space.

5 TWO-BODY RESTITUTION MODEL

We have considered an idealized one-body contact scenario, and now we address the problem of estimating restitution coefficients for the more complex two-body vibro-impact problem. In this section, we describe how to compute two-body restitution values (EPS2) in §5.1, and, justified by numerical experiments, we show in §5.2 how to approximate EPS2 values using one-body restitution values (EPS1) available in Bounce Maps.

5.1 Two-body Point-Contact Solver

It is straightforward to generalize the single-point plane contact solver from §3 to the two-body case. Given two bodies, A and B, and two proxy contact points, $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$, we replace the one-body unilateral contact condition (7) by

$$C(\boldsymbol{x}_A,\boldsymbol{x}_B) = \boldsymbol{n}^T(\boldsymbol{x}_A-\boldsymbol{x}_B) \geq 0, \tag{11}$$

where $\boldsymbol{n}$ is the fixed contact normal, and the bodies are oriented to be in contact, e.g., with opposing contact point normals. Following standard notation for the flexible multibody problem [Kaufman et al. 2008], the same equations from before apply with suitable substitutions, e.g., the generalized velocity vector is the concatenation of each body's velocity vector, $\dot{\mathrm{q}} = (\dot{\mathrm{q}}_A,\dot{\mathrm{q}}_B) \in \mathbb{R}^{(m_A+m_B+12)}$, the Jacobian matrix is $\boldsymbol{J} = [J_A \;\; -J_B] \in \mathbb{R}^{3\times(m_A+m_B+12)}$, the mass matrix is the block-diagonal extension, the generalized normal is still $\mathrm{n} = \boldsymbol{J}^T\boldsymbol{n}$, etc. The BD bound is now the sum of each body's R_{bound} value.

Discussion: The two-body contact solver (direct generalization of Algorithm 1) can used to compute restitution values between any pair of points, and we use it to explore this interesting function. However, for precomputation strategies, since EPS2 mapping would have to be performed pairwise, between each pair of points, on each pair objects, it has an undesirable quadratic precomputation cost and runtime storage. Furthermore, it requires all object pairs be processed ahead of time, which can be expensive for many objects.

We therefore propose two practical alternatives:

- **On-the-fly COR evaluation** using fast two-body restitution analysis can provide accurate estimates, but requires special runtime data structures and is not real time.
- **EPS Combiners:** Approximate EPS2 values can be obtained by combining EPS1 values, as is common practice; some possibilities include max, min, arithmetic mean, harmonic mean, etc. In the following section, we investigate how Bounce-Mapped EPS1 values should be combined for best results.

5.2 EPS-Combiner Investigation

In this section, we explore mathematical models for combining two one-body EPS1 values, ε_i and ε_j, to approximate the true two-body EPS2 value,

$$\varepsilon_{ij} \approx \mathcal{F}(\varepsilon_i,\varepsilon_j), \tag{12}$$

where we refer to the function, $\mathcal{F}$, as an "EPS combiner." Such a model is a huge simplification of the underlying contact dynamics, but is useful in practice because it enables real-time evaluation using precomputed Bounce Maps.

Candidate EPS combiners: We consider all combiners satisfying the following design criteria:

Unity:	$\mathcal{F} : [0,1]\times[0,1] \to [0,1]$
Symmetry:	$\mathcal{F}(\varepsilon_i,\varepsilon_j) = \mathcal{F}(\varepsilon_j,\varepsilon_i)$
Monotonicity:	$\partial_i\mathcal{F} \geq 0, \quad \partial_j\mathcal{F} \geq 0$
Identity:	$\varepsilon = \mathcal{F}(\varepsilon,\varepsilon)$,

where the last Identity follows by considering mirror contact problems (see Figure 13). In addition, we only consider *non-parametric* models, so as to not introduce additional object-specific parameters that must be determined. We consider seven candidate models: max, min, means (arithmetic, geometric, harmonic), and "flipped" combiners $1-\mathcal{F}(\bar{\varepsilon}_i,\bar{\varepsilon}_j)$ (where $\bar{\varepsilon} \equiv 1-\varepsilon$) which yield flipped geometric and harmonic means (see equations in Table 1).

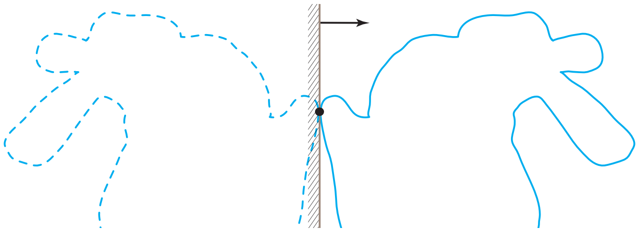


Fig. 13. **Mirrored restitution problems:** Our one-body restitution analysis with a rigid planar surface can be interpreted as a two-body problem for a perfectly symmetric configuration. Both the virtual (Left) and real (Right) objects see one-body EPS1 values of ε, and the simultaneous two-body EPS2 simulation will also result in ε. Consequently, EPS combiners should satisfy the Identity relationship, $\varepsilon = \mathcal{F}(\varepsilon,\varepsilon)$.

EPS2 Dataset: Using the two-body point-contact solver from §5.1, we numerically investigate the candidate EPS combiners for several models. Specifically, we computed self-collisions between an object and a copy of itself. For each object, we generated 200 sample locations that provided an approximately uniform sampling of Bounce Map ε values. For each pair of selected contact points, i and j, we perform a two-body restitution analysis to obtain ε_{ij} (exploiting symmetry); in our dataset, we compute 40000 two-body COR values per object. We plot ε_{ij} as a function of ε_i and ε_j in Figure 14.

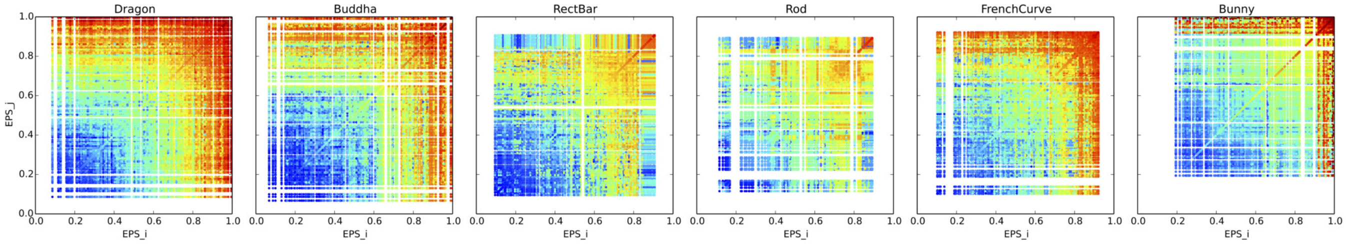


Fig. 14. **Two-body restitution coefficients,** ε_{ij}**,** plotted as a function of ε_i and ε_j values, reveal complicated distributions (ε jet colormap identical to previous plots). Each two-body restitution coefficient was computed by colliding each object with a copy of itself at 40000 (i, j) vertex pairs, with known ε_i and ε_j values spanning the available range.

Model Name	$\mathcal{F}(\varepsilon_i, \varepsilon_j)$	Dragon	Buddha	RectBar	Rod	FrenchCurve	Bunny	<Error>
Arithmetic mean	$(\varepsilon_i + \varepsilon_j)/2$	0.096	**0.106**	**0.114**	**0.108**	0.092	0.174	**0.115**
GeometricFlipped	$1 - \sqrt{\bar{\varepsilon}_i \bar{\varepsilon}_j}$	**0.067**	0.109	0.124	0.119	0.084	0.221	0.121
Geometric mean	$\sqrt{\varepsilon_i \varepsilon_j}$	0.121	0.127	0.117	0.112	0.103	0.166	0.124
HarmonicFlipped	$1 - 2/(1/\bar{\varepsilon}_i + 1/\bar{\varepsilon}_j)$	0.077	0.123	0.136	0.136	**0.090**	0.239	0.133
Harmonic mean	$2/(1/\varepsilon_i + 1/\varepsilon_j)$	0.144	0.148	0.128	0.124	0.115	0.161	0.137
Min	$\min(\varepsilon_i, \varepsilon_j)$	0.208	0.196	0.152	0.158	0.154	**0.152**	0.170
Max	$\max(\varepsilon_i, \varepsilon_j)$	0.128	0.176	0.190	0.198	0.143	0.276	0.185

Table 1. **EPS-Combiner** ℓ_1 **fitting error for two-body experiments** sorted by average ℓ_1 error (far right column). Bold values indicate the lowest error for each column. The arithmetic mean is the best model for this dataset of 240000 (=6*40000) two-body contact simulations.

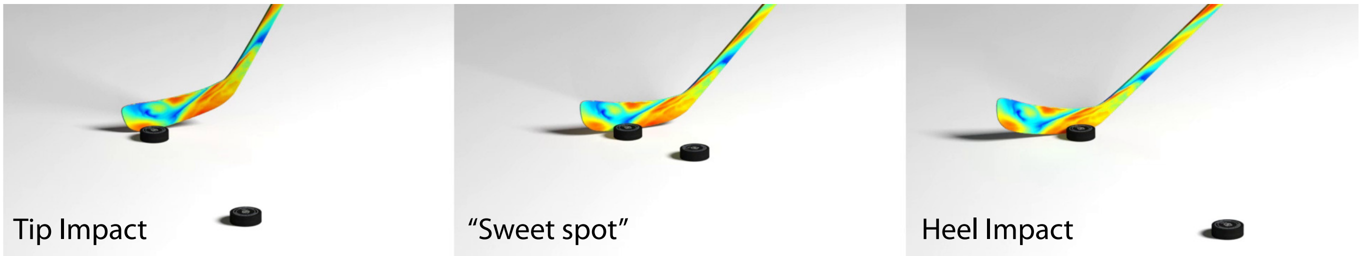


Fig. 15. **Hockey passes** reveal an interesting two-body restitution phenomena: the bouncy blade “tip” and “heel” locations (indicated) cause a fast-moving puck (ε=0.12) to rebound far more than when impacting the less bouncy “sweet spot” at the center of the blade. Pucks are shown at impact locations, and rebound locations at a fixed time later, for each scenario. Two-body restitution values are computed using an arithmetic average EPS-combiner.

And the best combiner is... For each object and candidate combiner model, we compute the average ℓ_1 fitting error over all samples (see Table 1). Interestingly the *arithmetic average* is the best predictor for the nearly quarter million ε_{ij} values considered, with an average ε_{ij} error of about 0.1, or approximately 10% impulse error, which is at the limits of human perception for collisions [O’Sullivan et al. 2003]. An illustrative “hockey” example is shown in Figure 15.

6 EXTENSIONS

Damping and Plasticity: The estimated restitution values essentially represent the maximum values we might expect in the absence of other dissipative factors, such as internal damping, plasticity, and friction. While the last two phenomena are beyond the scope of this paper, we can easily add internal damping to our simulations by modifying the modal oscillator’s internal force accordingly. For example, adding stiffness-proportional Rayleigh damping leads to $\ddot{q}_i = -\omega_i^2(q_i - \beta\dot{q}_i)$. The dependence of ε on β is illustrated in Figure 16, and is itself quite complex. One practical alternative to modeling plasticity and damping effects for graphics practitioners is to precompute the undamped $\varepsilon(\boldsymbol{X}, \boldsymbol{N})$ map, so as to capture interesting spatial variation due to the shape, then simply scale the map values as $\alpha\varepsilon$, where $\alpha \in [0, 1]$, to achieve the desired level of attenuation.

Friction: It is straightforward to incorporate a Coulomb friction model into the point-plane contact problem, and thus compute $\varepsilon(\boldsymbol{X}, \boldsymbol{N}, \mu)$ values using an appropriate solver, e.g., [Kaufman et al. 2008]. Since normal impulses will excite tangential velocity components in the modal vibrations, adding friction can result in increased dissipation, and tend to produce smaller restitution values. Unfortunately, since μ is a pairwise property of two contacting surfaces, it leads to complications similar to those of efficiently computing EPS2. This problem is likely to be more complicated since object features that lead to surface friction are less well understood. We leave this for future work.

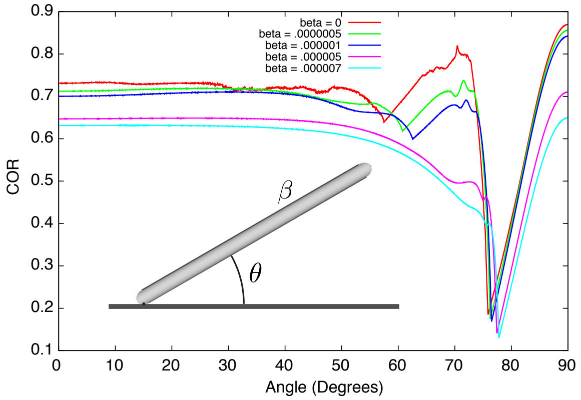


Fig. 16. **Restitution versus stiffness-proportional damping:** Using the classic rod impact example from [Stoianovici and Hurmuzlu 1996], we demonstrate the dependence of ε on internal elastic damping by varying the stiffness-proportional damping, β. For extremely large values of β we see noticeable drops in ε. Notice increasing damping generally tends to reduce ε, there are some points where it actually increases. ($m = 90$ modes)

Symmetry: Preserving symmetry is an important desideratum for rigid-body simulation (SYM from [Smith et al. 2012]). Tabulated bounce maps for symmetric objects may or may not be symmetric due to surface or volume meshing differences, interpolation error, etc. However, we can preserve symmetry and reduce the precomputation costs by exploiting symmetry in our preprocess and runtime lookup. Prior to restitution analysis, we compute the object's symmetry groups (n-way, mirror, cylindrical) using the moment-function of [Martinet et al. 2006], then, similar to [Langlois et al. 2014], we identify a minimal symmetry patch of the surface. We then sample the representative patch using appropriate smooth or nonsmooth sampling methods. At runtime, Bounce Map evaluation is done by first transforming $(\boldsymbol{X}, \boldsymbol{N})$ to a representative patch location $(\boldsymbol{X}', \boldsymbol{N}')$, and then returning the interpolated value $\varepsilon(\boldsymbol{X}', \boldsymbol{N}')$. In this way, we can reduce analysis costs, and enforce joint ε/object symmetry.

7 RESULTS

Model and sampling statistics are given in Table 2 for all examples. Bounce Maps for various geometric models were shown previously in the paper for the letters (Fig 1), spring (Fig 4), and the Buddha and Dragon (Fig 10). Additional Bounce Maps are shown for rod-like objects in Figure 17, and in Figure 19 for various objects. Smoothness properties, and spatial and angular structure of $\varepsilon(\boldsymbol{X}, \boldsymbol{N})$ were demonstrated earlier in Figures 3 and 9. One-body restitution ground-impact examples are shown in the supplemental video for a ruler (with α attenuation) and a baseball bat, and demonstrate dramatic variations due to bounce mapping. Nontrivial two-body EPS-combiner examples are shown for (a) ruler-table impacts in Figure 24, (b) interesting puck rebounds from a hockey stick depending on impact location (see Figure 15), and (c) dropping BOUNCE letters in Figure 2. Please see our accompanying video for animated results and additional footage.

Implementation: In our C++ implementation, the fast restitution analysis method is pleasantly parallel and allows rapid evaluation of ε samples. Since each $\varepsilon(\boldsymbol{X}, \boldsymbol{N})$ computation only requires $O(m)$ data to compute the object's dynamics independently at $O(m)$ flops per timestep, it each be trivially parallelized. We computed millions of ε values on a single workstation (dual Intel Xeon E5-2690V3 2.6GHz

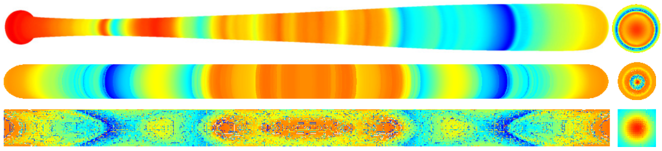


Fig. 17. **Bats and bars:** (Left) We observe that long rod-like objects tend to have more bounce (orange/red) near their middle, and characteristic "dead zones" (in blue) toward their ends. The baseball bat has some notable responses, including a very bouncy end cap (see video for demonstration). In addition, while baseball bat and ball contact interactions can involve complicated conforming contact, it is interesting to see that the COR minima (dark blue) in the much-simplified single-body bounce-map analysis roughly land in the region of a bat's actual "sweet spot." (Right) Strong angular variations can be observed on the end caps of each rod, with the highest value at the center (corresponding to $\theta = 90^o$ for the smooth rod).

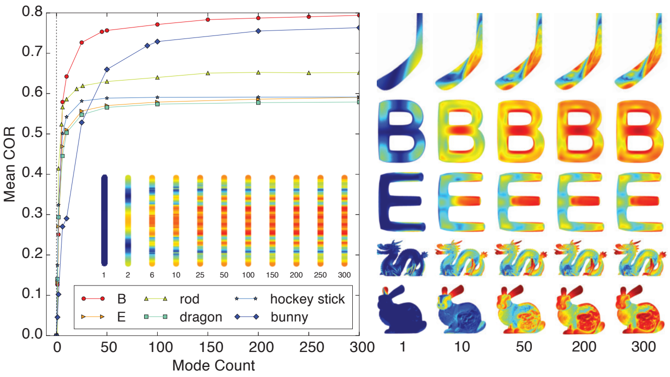


Fig. 18. **Convergence of restitution analysis versus the number of modes,** m is shown for different models. Observe that restitution values for low-mode-count models tend to underestimate ε.

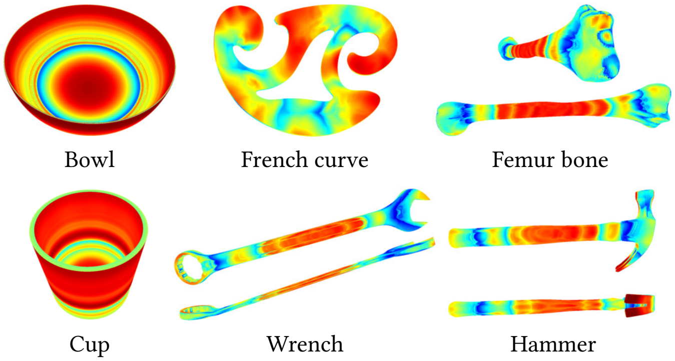


Fig. 19. **Bounce maps for smooth geometry:** The sharp rims of the symmetric bowl and cup models have additional nonsmooth edge samples (not shown). The bone, wrench and hammer examples have similarities to other rod examples (see Figure 17).

	Surface Mesh		Tetrahedral Volume Mesh		ε Samples			ε Statistics		
Name	Vertices	Triangles	Vertices	Tetrahedra	Vertex	Edge	Face	min(ε)	max(ε)	avg(ε)
Bunny	112912	225820	30464	130503	0	0	225820	0.149891	1	0.650141
French Curve	87826	175652	58184	247490	11084	127956	175652	0.057375	0.91785	0.579018
Dragon	156836	313672	97004	415903	0	0	313672	0.000424	1	0.573
Flat Spring	50642	101280	208673	855037	0	0	101280	0	1	0.671125
Buddha	99934	199896	365644	1596938	0	0	199896	0.081336	0.994896	0.533583
Baseball Bat	127746	255488	9665	37600	0	0	255488	0.069281	0.956061	0.636968
Rod	151170	302336	49342	205353	0	0	302336	0.109042	0.8551	0.576428
Rectangular Bar	24802	49600	44321	240000	2696	19200	49600	0.11027	0.849087	0.560266
Bowl	247990	495976	118740	423879	0	0	495976	0.185017	1	0.755548
Bowling Pin	8962	17920	9390	39666	0	0	17920	0.176574	0.911379	0.62008
Femur Bone	79066	158128	2903	10836	0	0	158128	0.054912	0.912503	0.593628
Cup	59341	118678	89962	374780	4933	22760	118678	0.318098	0.967689	0.730247
Wrench	193492	386984	8317	30718	11060	1078738	386984	0.039464	0.920994	0.613731
Hammer	80384	160760	82904	350381	15321	414051	160760	0.000001	1	0.502898
"B"	9357	18718	19276	82366	0	0	18718	0.352972	0.944619	0.745858
"O"	8303	16606	16870	71923	0	0	16606	0.258659	0.927492	0.776712
"U"	7229	14454	14596	61374	0	0	14454	0.093109	0.909523	0.579912
"N"	8414	16824	17443	74712	0	0	16824	0.088812	0.949161	0.568093
"C"	7096	14188	14371	61134	0	0	14188	0.107234	0.940146	0.567126
"E"	8406	16808	16839	70868	0	0	16808	0.12469	0.955651	0.573654
Ruler	10974	21944	13509	41408	0	0	21944	0.124216	0.912349	0.553389
Table	28996	57988	48842	188592	1293	2404	3773	0	1	0.559098
Hockey Stick	56506	113008	106610	439458	57307	141332	113008	0.028524	0.982618	0.587974

Table 2. **Model statistics** including surface triangle mesh used for contact sampling; volumetric mesh used for modal analysis; number of ε samples on faces, edges and vertices; and basic ε statistics. All examples were processed with mass density $\rho = 2000\ kg/m^3$, Young's modulus $E = 7.0 \times 10^{10}$, Poisson's ratio $\nu = 0.3$, and $m = 45$ modes.

12-core processors). Unless stated otherwise, we interpolated vibration modes of the volumetric model onto uniform triangle meshes, then evaluated ε values at face centroids, and (if nonsmooth) at vertex and edges; processed models used at least $m = 45$ modes (see the discussion of convergence against mode count below), unless otherwise stated.

Performance of fast restitution analysis: Quoting for the bunny model ($m = 45$ modes) we observe 254s / 10000 samples in serial, and 12s / 10000 samples in parallel (48 threads). For any specific contact sample, the number of timesteps taken (with $\Delta t = 0.1\mu s$) varied over a large range, from 400 to 14000 timesteps, and thus the contact duration varied from $40\mu s$ to $1.4ms$. We observe that long contact times are also correlated with high ε values (see Figure 20). Nonsmooth models can take longer due to angular sampling, and were not processed in the majority of our examples. Runtime lookup costs for Bounce Map values are negligible for smooth models, whereas nonsmooth surfaces depend on the cost of local interpolation and the density of sampling.

Comparison to SH96: Our simulation produces qualitatively similar results to [Stoianovici and Hurmuzlu 1996] for a 20 mm steel rod without (Fig 3) and with damping (Fig 16). However, their particular setup had slightly different settings, such as rod-slab friction, which may make the curves differ slightly.

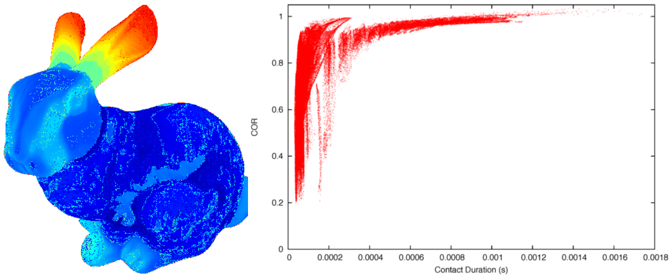


Fig. 20. **Contact duration, τ,** (Left) ranged from $40\mu s$ to $1.4ms$ for this steel bunny (plotted here as $\log_{10}(\tau)$). (Right) We observe that high restitution values are correlated with longer contact durations (225k samples shown), and tend to occur at more compliant features of the model. For example, the bunny's ears are very compliant and also have the highest ε values, suggesting that their long contact periods allow vibrational energy to be converted back into rigid-body motion effectively.

Convergence of the restitution analysis: Convergence as a function of modes, m, has been explored for different models. All models converged quickly (see Figure 18). We found that using $m \approx 50$ results in a good trade-off between preprocessing speed ($\sim O(m)$) and bounce map accuracy (e.g. error in mean ε between using $m = 50$ and $m = 300$ are 0.037 for the letter 'B', 0.022 for the rod, and 0.103 for the bunny).

Convergence (Conservative vs breaking contact impulses): Our restitution analysis generates inelastic breaking impulses (λ) and will thus generate $\varepsilon = 0$ values for $m = 0$ modes, then generally increase with increasing m. However, to model an energy-conserving system, one could use an elastic impulse (2λ) instead, which will produce $\varepsilon = 1$ for $m = 0$ modes, then generally decrease as m increases. We compare these two approaches in Figure 23, and note that breaking impulses are preferrable.

Parameter Dependence: We demonstrate the weak dependence of estimated restitution values on several parameters used in the preprocess: (a) the elasticity material parameters ρ and E (see Figure 21), and (b) the initial normal impact velocity, v_n^- (see Figure 22). We acknowledge that changes in Poisson's ratio ν can lead to changes in mode shape, that can affect restitution values.

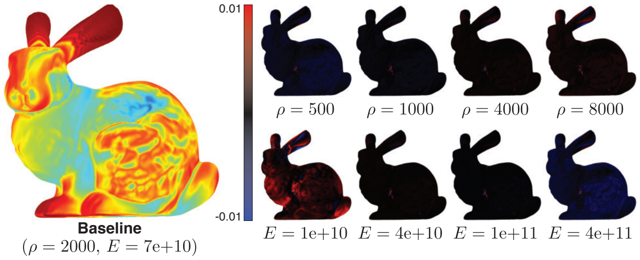


Fig. 21. **Weak dependence on material parameters** is demonstrated for mass density, ρ, and Young's modulus, E. We observe less than 1% change in ε while varying the parameters over an order of magnitude around the default "steel"-like material. More extreme variations, e.g., very low E or very high ρ, could lead to violation of the rigidity assumption. (#modes=50)

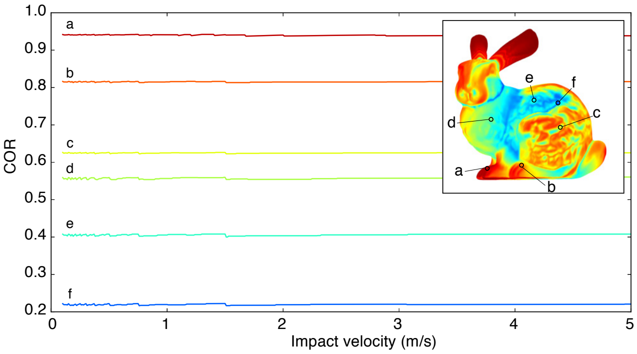


Fig. 22. **Weak dependence on impact velocity, v_n^-:** In Algorithm 1, recall that we set the pre-impact normal velocity to be $v_n^- = -1\ \frac{m}{s}$. However, we demonstrate that restitution values generated by our model have very weak dependence on v_n^- over a wide range of relevant speeds, as shown plotted here for six contact locations on the bunny object. (#modes=45)

8 CONCLUSION

We have proposed an efficient method for sampling the coefficient of restitution ε over the surface of an object, and enabling cheap runtime evaluation for impact events in commodity rigid-body simulators. The key insight is that significant variation in ε is due to vibration loss during rapid but complex micro-collision events, and that these processes can be efficiently simulated in parallel to map the surface's bounciness. We believe that such techniques provide a simple and practical way to improve the realism of rigid-body animations in computer graphics. Beyond graphics, the development of more accurate restitution models will be of wide use in scientific and engineering contact analysis.

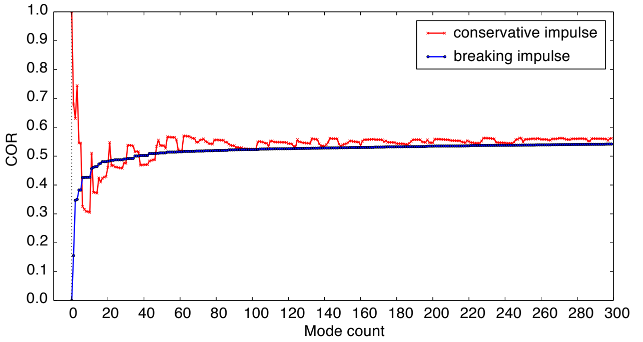


Fig. 23. **Convergence of restitution analysis using elastic vs inelastic impulses:** Both methods approach similar ε values as $m \to \infty$, but start from $\varepsilon = 1$ and 0 (when $m = 0$), respectively. However, the conservative case has noisier convergence, and requires much smaller time-steps for comparable accuracy, e.g., 100x-1000x smaller, and so we use breaking impulses.

Limitations and Future Work: Classical rigid-body impact with restitution is an enormous simplification of the highly complex interactions that occur between impacting stiff bodies. Despite our contributions, our problem is also an over simplification of a truly messy problem. Our analysis is based purely on elastic vibrations, possibly with damping, and can recover interesting spatial variations. However, the accurate estimation of restitution for a specific model would require detailed modeling of the material properties, and friction. We have proposed a single-body preprocess to approximate ε and thereby enable a practical precomputation-based solution, however it is inherently a complex two-body impact problem, and single-body EPS combiners provide only an approximate solution. Friction is important in vibrational impact, and further complicates efficient preprocessing methods. Plasticity (and material nonlinearity) is an important source of energy loss during hard impacts, as is dependence on normal velocity magnitude. We have assumed that v_n^- arises only due to translational velocity, however nonzero angular velocity contributions do introduce a slight additional variation. We have considered stiff materials, such as steel, in our precomputation, however the restitution model and analysis become less applicable for softer objects, e.g., due to large deformations. Bounce maps have highly unpredictable spatial and angular structure, and it is difficult to know *a priori* what resolution is sufficient for the functions. We have used a simple uniform mesh sampling scheme, but more attention should be provided to adaptive sampling schemes. The method is highly parallel with a low memory footprint, and is an ideal candidate for GPU acceleration. Finally, recent work in granular matter has explored the

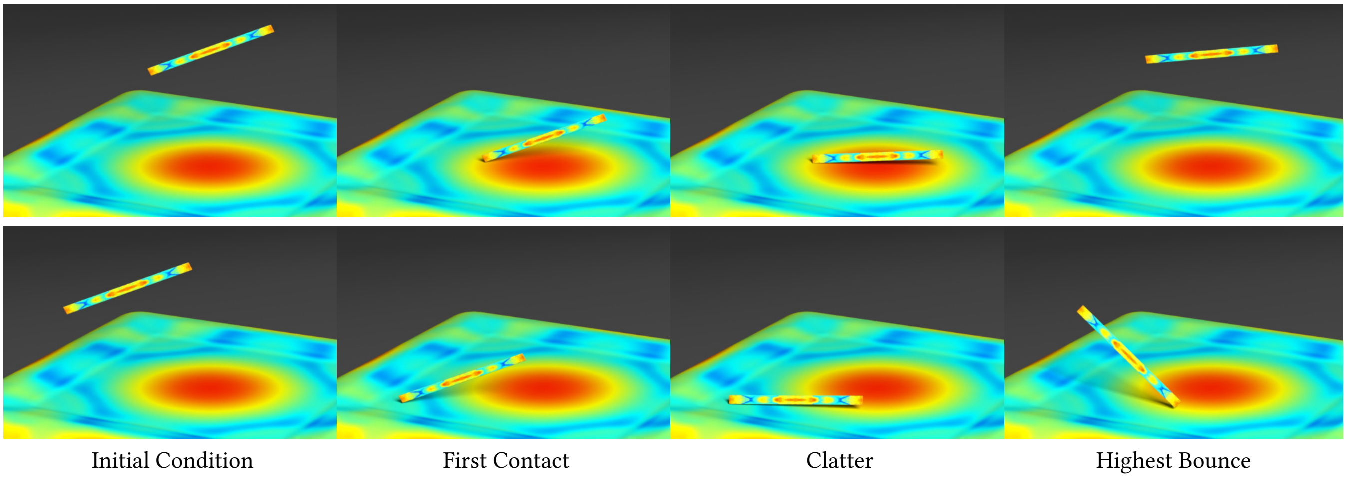


Fig. 24. **Two-body impact animation** (left to right) between bounce-mapped ruler and table models, computed using EPS-combiner restitution coefficients. Two very different responses occur for ruler impacts near (Top) the very bouncy table center versus (Bottom) an off-center location. The highest rebound height (far-right frames) is clearly greater for the table center case.

tangled connection between normal COR, tangential COR, and friction [Doménech-Carbó 2014], which remain to be explored in the setting of vibrational restitution analysis.

ACKNOWLEDGMENTS

DKP acknowledges funding from NSERC and the Canada Research Chairs Program, and DLJ thanks Adobe Research for donations. We thank the anonymous reviewers for constructive feedback, Timothy Langlois for software assistance, and early inspiration from Chris Ullrich.

REFERENCES

David Baraff. 1997. An introduction to physically based modeling: Rigid body simulation II–Nonpenetration constraints. *SIGGRAPH course notes* (1997).

Ronen Barzel, John R Hughes, and Daniel N Wood. 1996. Plausible motion simulation for computer graphics animation. In *Computer Animation and Simulation 96*. Springer Vienna, 183–197.

Jan Bender, Kenny Erleben, and Jeff Trinkle. 2014. Interactive simulation of rigid body dynamics in computer graphics. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 246–270.

Bernard Brogliato. 2012. *Nonsmooth mechanics: models, dynamics and control. Second edition.* Springer Science & Business Media.

Anindya Chatterjee and Andy Ruina. 1998. A new algebraic rigid-body collision law based on impulse space considerations. *Journal of Applied Mechanics* 65, 4 (1998), 939–951.

Stephen Chenney and D. A. Forsyth. 2000. Sampling Plausible Solutions to Multi-Body Constraint Problems. In *Proceedings of SIGGRAPH 2000*. 219–228.

Antonio Doménech-Carbó. 2014. On the tangential restitution problem: independent friction–restitution modeling. *Granular Matter* 16, 4 (2014), 573–582.

W. Goldsmith. 1960. *Impact: The Theory and Physical Behaviour of Colliding Solids.* Edward Arnold.

Doug L. James and Dinesh K. Pai. 2004. BD-Tree: Output-sensitive Collision Detection for Reduced Deformable Models. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 393–398. https://doi.org/10.1145/1015706.1015735

Jingyi Jin, Michael Garland, and Edgar A. Ramos. 2009. MLS-based scalar fields over triangle meshes and their application in mesh processing. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games (I3D '09)*. ACM, New York, NY, USA, 145–153. https://doi.org/10.1145/1507149.1507173

Danny M Kaufman, Shinjiro Sueda, Doug L James, and Dinesh K Pai. 2008. Staggered projections for frictional contact in multibody systems. In *ACM Transactions on Graphics (TOG)*, Vol. 27. ACM, 164.

Timothy R. Langlois, Steven S. An, Kelvin K. Jin, and Doug L. James. 2014. Eigenmode Compression for Modal Sound Models. *ACM Trans. Graph.* 33, 4, Article 40 (July 2014), 9 pages. https://doi.org/10.1145/2601097.2601177

Aurélien Martinet, Cyril Soler, Nicolas Holzschuch, and François X. Sillion. 2006. Accurate Detection of Symmetries in 3D Shapes. *ACM Trans. Graph.* 25, 2 (April 2006), 439–464. https://doi.org/10.1145/1138450.1138462

B. Mirtich and J. F. Canny. 1995. Impulse-based dynamic simulation of rigid bodies. In *Symposium on Interactive 3D Graphics*.

Carol O'Sullivan, John Dingliana, Thanh Giang, and Mary K Kaiser. 2003. Evaluating the visual fidelity of physically based animations. In *ACM Transactions on Graphics (TOG)*, Vol. 22. ACM, 527–536.

P. Painlevé. 1895. Sur les Lois du Frottement de Glissement. *Comptes Rendus de l'Academie des Sciences* 121 (1895).

Friedrich Pfeiffer and Christoph Glocker. 2008. *Multibody Dynamics with Unilateral Contacts.* John Wiley & Sons.

Jovan Popović, Steven M. Seitz, Michael Erdmann, Zoran Popović, and Andrew P. Witkin. 2000. Interactive Manipulation of Rigid Body Simulations. In *Proceedings of SIGGRAPH 2000*. 209–218.

Ahmed A Shabana. 2012. *Theory of Vibration: An Introduction.* Springer Science & Business Media.

Ahmed A Shabana. 2013. *Dynamics of multibody systems.* Cambridge university press.

Breannan Smith, Danny M. Kaufman, Etienne Vouga, Rasmus Tamstorf, and Eitan Grinspun. 2012. Reflections on Simultaneous Impact. *ACM Trans. Graph.* 31, 4, Article 106 (July 2012), 12 pages. https://doi.org/10.1145/2185520.2185602

David E Stewart. 2011. *Dynamics with Inequalities: impacts and hard constraints.* SIAM.

Dan Stoianovici and Yildirim Hurmuzlu. 1996. A Critical Study of the Applicability of Rigid-Body Collision Theory. *Journal of Applied Mechanics* 63, 2 (1996), 307.

William James Stronge. 2004. *Impact mechanics.* Cambridge University Press.

Christopher D. Twigg and Doug L. James. 2007. Many-Worlds Browsing for Control of Multibody Dynamics. *ACM Transactions on Graphics* 26, 3 (July 2007), 14:1–14:8.

C Ullrich and Dinesh K Pai. 1998. Contact response maps for real time dynamic simulation. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, Vol. 3. IEEE, 1950–1957.

C Ullrich and Dinesh K Pai. 1999. Green's function contact maps for accurate real time collisions. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, Vol. 3. IEEE, 1849–1855.

Yu Wang and Matthew T. Mason. 1987. Modeling Impact Dynamics for Robotic Operations. In *Proceedings of the IEEE International Conference on Robotics and Auutomation.* 678–685.

Changxi Zheng and Doug L. James. 2011. Toward High-Quality Modal Contact Sound. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2011)* 30, 4 (Aug. 2011). http://www.cs.cornell.edu/projects/Sound/mc