Stanford University, Fall 2015, CS 448Z: Physically Based Animation and Sound

Homework #2: Real-time Rigid-Body Dynamics with Modal Sound

Due: Thursday Nov 12, 2015

Prof. Doug L. James
Stanford University
October 23, 2015

Abstract

The goal of this assignment is for you to implement an interactive audiovisual simulation of a rigid-body system which includes sounds generated using modal sound synthesis. You will be provided with the geometric models, and precomputed modal sound data (eigenmodes and frequencies).

1 Introduction

Acceleration noise is an important component in rigid-body contact sound, and is the dominant source of sound for small objects [Chadwick et al. 2012b; Chadwick et al. 2012a]. However, for larger objects, significant vibrations occur at audible frequencies. Modal vibration analysis and modal sound synthesis provide good approximations of these sounds (see the appendix in [Zheng and James 2010] for more information). In this assignment, you will build a simple but compelling audiovisual simulation of a rigid body, by implementing a modal sound synthesis pipeline. In addition you can augment it with a simplified acceleration noise model to enhance contacts involving small objects.

2 Part I: Rigid-body Simulation

In this assignment, you will extend your code from the previous assignment on acceleration noise to support more general rigid body dynamics. For simplicity we will initially only consider the case of a rigid object colliding with a planar ground plane. Please implement the following:

1. *Rotational dynamics* should be implemented using quaternions. Feel free to use an existing quaternion library implementation, e.g., `Eigen::Quaternion`.
2. *Inertia matrix* can be approximated by assuming that the mass of the object lies on the surface mesh, and therefore you simply need to sum over each triangle's contribution, e.g., assuming that its mass lies at its centroid. Alternately the inertia matrix can be computing accurately for a uniform material density by converting the volume integral to a surface integral (see Mirtich).
3. *Collision detection* for vertex-plane interactions has already been implemented in HW1, and can be reused to determine precise collision times within a timestep.
4. *Contact impulses* can be computed for a user-specific restitution coefficient between the plane and object vertices. The impulse can be applied directly to the rigid-body dynamics model, and cached for later application to the modal sound model.
5. *Mouse interaction* should be supported so that you can apply impulses to the object and/or push it around in the virtual environment.

3 Part II: Synthesizing Modal Sound

Given the starter code and data for this assignment, you will be able to load geometry (OBJ mesh), and corresponding mass-normalized eigenmodes and frequencies[1] resulting from a volumetric modal analysis. You will be provided with 3 modal models: a sphere, rod, and a slab.

Please implement the following:

1. *Time-stepping modal dynamics:* As described in class, this data will allow you to timestep each mode's SHO using an IIR filter (e.g., see [James and Pai 2002]), or a traditional time-stepping scheme (such as symplectic Euler, or the midpoint/Newmark method).
2. *Apply contact impulses from the rigid-body simulator* to each modal SHO to excite their dynamics. Using a smoothed contact force profile one some contact time duraction τ will enable you to model more realistic contact forcing and sound response.
3. *Generate sound* by listening directly to the sum of the modal coordinates $s(t) = \sum_i q_i(t)$ at each time sample, t. You can ignore retarded time offsets (delays) for this assignment.
4. *Tune damping parameters:* Different materials have different internal damping behaviors which affect the decay of modal vibrations. To produce realistic sound, you will need to adjust the Rayleigh damping parameters, α and β. One recommendation is to set $\alpha = 0$, then tune β as best you can, then adjust α only to dampen low frequencies as much as necessary.

In addition, in order to model different materials and/or different uniform scalings from those provided in the starter code/data, you may need to use the following handy scalings:

1. *Stiffness scaling:* The shear modulus parameter, G, or elastic modulus, E, are related by
$$G = \frac{E}{2(1+\nu)},$$
(see `https://en.wikipedia.org/wiki/Shear_modulus`) and linearly scale the stiffness matrix K and thus determine how strong the forces are for a given deformation of the linear elastic model, and affect the frequency scaling. Specifically, given a mode's frequency ω computed for a given shear modulus, G, it's adjusted frequency for a new modulus G' is
$$\omega' = \sqrt{\frac{G'}{G}}\omega,$$
so that quadrupling G will double frequencies. Similarly, since $G \propto E$,
$$\omega' = \sqrt{\frac{E'}{E}}\omega,$$

---

[1] The starter code actually gives the eigenvalues, $\lambda = \omega^2$, and not the frequencies directly.

is the expression for elastic moduli. The starter code examples use the following parameters:

- *Sphere and Rod:* $E = 7e10, \nu = 0.3$
- *Ground slab:* $E = 2e11, \nu = 0.4$

You can change your materials by adjusting ω appropriately.

2. *Density scaling:* The provided modal data assumes $\rho = 1$ which will need to be adjusted for realistic materials. You change the mass density of the material by scaling the frequencies since $\omega^2 \propto 1/\rho$, and therefore changing density from 1 to ρ shifts frequencies to
$$\omega' = \frac{1}{\sqrt{\rho}}\omega.$$
It also changes $\boldsymbol{U}$ due to the mass normalization, $\boldsymbol{U}^T\boldsymbol{M}\boldsymbol{U} = \boldsymbol{I}$. Therefore $\boldsymbol{U}$ will need to be scaled such that $\boldsymbol{U} \propto \frac{1}{\sqrt{\rho}}$, i.e.,
$$\boldsymbol{U}' = \frac{1}{\sqrt{\rho}}\boldsymbol{U}.$$

3. *Size scaling:* The objects were precomputed for a fixed size, however, you may wish to uniformly scale the object to make it bigger or smaller. You can do this by using the scaling formulae presented in Equation 9 of [Zheng and James 2010] to change ω and thus Rayleigh damping, and the mode matrix due to mass normalization.

4 One Notable Extension

Once you have implemented the core system, you should try to implement *one* extension and make a "milestone" video as your final result. Tyings you might consider are as follows:

- *Complete sphere-sphere simulator:* By combining the sphere modal sound model with the acceleration noise model from the first assignment you can implement a consistent sphere-sphere contact sound simulation.
- *Nonspherical object-object collisions:* To support object-object collisions, you will need to perform collision detection for general non-spherical objects. Bounding volume hierarchies (BVHs) based on spheres or bounding boxes, are particularly effective for rigid objects, and will allow you to detect collisions more efficiently. Alternately you approximate collisions using proxy geometry, such as using a sphere-based approximation of the shape, e.g., inflating each vertex to collectively cover the surface.
- *Contact damping:* Getting a high quality sound can require careful modeling of contact interactions. One possibility is to include contact damping which damps modal vibrations when it is in contact. One possibility is the contact damping model used in [Zheng and James 2011], but other simpler dampers can be used.

5 What to Submit

You should submit (1) your code, (2) a short 1-page write-up on your findings, (3) video captures of (i) a notable simulation example, and (ii) your extension results–these videos will be shared with others in the class. Upload your final submission as a zip file to Canvas (`http://canvas.stanford.edu`).

References

CHADWICK, J. N., ZHENG, C., AND JAMES, D. L. 2012. Faster acceleration noise for multibody animations using precomputed soundbanks. *ACM/Eurographics Symposium on Computer Animation*. Project page: `http://tinyurl.com/np675hm`.

CHADWICK, J. N., ZHENG, C., AND JAMES, D. L. 2012. Precomputed acceleration noise for improved rigid-body sound. *ACM Trans. Graph. 31*, 4 (July), 103:1–103:9. Project page: `http://tinyurl.com/nh7k88m`.

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

ZHENG, C., AND JAMES, D. L. 2010. Rigid-body fracture sound with precomputed soundbanks. 69.

ZHENG, C., AND JAMES, D. L. 2011. Toward high-quality modal contact sound. *ACM Transactions on Graphics (TOG) 30*, 4, 38.