Physical based sound rendering: simple bar element

WENHAN ZHU (COSMOS)

University of Waterloo

April 2, 2018

Abstract

Physically based sound rendering is a field that synthesize sound physically based on material properties such as shape and elasticity. In this paper, the background on this field will be discussed and a simple FEM (Finite Element Method) on bar element will be implemented to show case the most simple case of the simulation. The limitations of such method and more advance research on how they have been approached will also be discussed at the end of this paper.

Introduction

We can create realistically looking CG that is indistinguishable from real world with modern computer graphics. However, the sound production of such CG relies on talented foley artists. In recent years, there have been numerous research done on automating this process to produce realistic sound that corresponds to the animation created by computer graphics. Though out the years there have been different models to address this problem. The details will be address later in the paper.

Related Work

Generating sound from animation was investigated early for musical instruments by Cook [1]. van den Doel et. al [2] [3] developed a method by using vibration modes of an object for generating sound. Then there are sound generation for rigid body simulations by O'Brien [4] with improved techniques. The main method used was FEM with tetrahedral nodes to compute the modal frequency and used linear modal sound synthesis to create the final sound. This paper's work is heavily based on these results but with a simpler method of 1D bar element node.

Modeling of sound

Basic Vibration Model

The most simple vibration model for objects is a spring mass system. It have only 1 degree of freedom and is described by the famous equation

$$m\ddot{x} + kx = 0$$

It has a fundamental frequency $\omega = \sqrt{\frac{k}{m}}$ and can be used to model a single wave as a result. Damping can also be added to the system so that the vibration will eventually stop.

Elastic Vibration Model

Linear modal synthesis [2] is used for modeling the dynamic deformation of a object and is used widely in physically based sound synthesis. Such model have the following form.

$$M\ddot{x} + C\dot{x} + Kx = f$$

M is the mass matrix, C is the damping matrix and K is the stiffness matrix. At lower frequencies, we can use Rayleigh damping to have a simple linear model of C based on M, K. By solving the generalized eigenvalue problem

$$KU = \Lambda MU$$

We can extract each of the frequencies from the results and use it to produce the final sound.

$$y(t) = \sum_{n=1}^{N} a_n e^{-d_n t} \sin 2\pi f_n t$$

More detailed explanation of a simplified version of this model will be discussed in the next section.

Sound generation of 1D bar element

Lets look at a specific model of a 1D bar element. Consider a bar of length l and using 2 nodes at each end. So we will have

$$x_1 = 0, x_2 = l$$

At each node the displacement and force function are $q_i(t), Q_i(t)$ respectfully.

Now, assume a simple shape function for the bar

$$u(x, t) = a_0 + a_1 x$$

We want to convert the differential equation

$$\ddot{u}(x, t) = c^2 u''(x, t)$$

where $\ddot{u}, u''$ are second derivatives relative to time and displacement respectfully.

To a system of ordinary differential equations

$$[m]\ddot{u}(t) + [k]u(t) = \vec{F}(t)$$

that can be easily solved by a computer.

With the previously shape function, we can combine the boundary conditions

$x = 0, u(0, t) = q_1(t)$

to get

$$a_0 = q_1(t)$$

and

$$x = l, u(l, t) = q_2(t)$$

to get

$$a_1 = \frac{q_2(t) - q_1(t)}{l}$$

Now replug the value back to the original shape function we can get

$$u(x, t) = q_1 + \frac{q_2 - q_1}{l}x$$

$$= \begin{bmatrix} 1 - \frac{x}{l} & \frac{x}{l} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

We will rewrite the two parts as

$$u(x, t) = \vec{\phi}^T(x) \cdot \vec{q}(t)$$

Where $\vec{\phi}^T$ is the shape function and $\vec{q}$ is the nodal coordinates.

From the previous equation we can derive the following:

$$\dot{u} = \phi^T \dot{\vec{q}}$$
$$\ddot{u} = \dot{\vec{q}}^T \vec{\phi} \vec{\phi}^T \ddot{\vec{q}}$$
$$u' = \vec{\phi}' \vec{q}$$
$$u'' = \vec{q}^T \vec{\phi}' \vec{\phi}'^T \vec{q}$$

3

Now we can use the conservation of energy in kinetic energy and strain energy to determine the mass matrix and stiffness matrix separately.

For the mass matrix, if we assume the bar has constant density ρ, constant elasticity E and area A we have

$$T = \frac{1}{2}\rho A \int_0^L \dot{u}^2(x,t)dx$$
$$u = \frac{1}{2}\int_v EAu'^2(x,t)$$

By expanding the equation we can derive the mass and stiffness matrix

$$[m] = \frac{\rho Al}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$[k] = \frac{EA}{l}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Now we can use to calculate the fundamental frequencies ω (which is the square root of the eigenvalue of the system) of a given bar element by the following equation.

$$\det[[k] - \omega^2[m]] = 0$$

The previously discussed system is flexible enough to calculate the result for a arbitrary number of nodes for the finite element method.

Now we arrive with all the calculated frequencies. We can derive the final wave file of the sum of them. To calculate the excitement for each frequency by using the delta function. We have the following

$$a_n = \frac{\psi_n}{c\alpha_n\omega_n}$$

where ψ_n is the eigenvector corresponding to the fundamental frequencies and α_n is it's norm.

So now we can add up each element and create the final sound.

Creating the simulation

Simulation Environment

The simulation is created in *Python* [5]. Using *sounddevice* [6] and *pyglet* [7] for sound and graphics packages accordingly. The computation is done with *numpy* [8] and *scipy* [9]. The simulation does not use a physics engine, the only contact is abstracted as a delta function on a point as discussed in the previous section. The model of each of the objects in the simulation will be computed as simulation startup and when the simulation is running, we only need to calculate the excitement for each element.

Simulation pipeline

1. Look for user input

2. Calculate contact location based on user input

3. Calculate sound based of contact location and object

4. Output sound

The code for simulation and a few audio examples using the technique described above is included in the hand in file.

Results and discussion

With the model we are using, we can produce sound that represents a vibrating bar. With different location hit on the bar it can produce different sounds. It show cases a very simple case for physically base sound rendering based on linear model methods.

Such model requires previous knowledge about the material properties such as density, elasticity modulus. In order to have more accurate sound base on this model, we need a lot of nodes to do the calculation. With many nodes the computation time will be high and it requires pre-computing time. With enough optimization and some pre-computed sound bank it is possible to produce sound at reasonable interactive rates. [10]

Due to the natural of such approach, it has limitation to the kind of sound it can produce. In real life not every sound produced is based on linear models. There have been research by Chadwick et. al [11] that explored areas that produces non-linear sounds such as acceleration noise and thin shell sounds.

The method discussed only focuses on the vibration on the model. Currently it is only considering it as a single sound source and output it without the consideration of acoustic transfer. Research

has been done address this problem using Helmholtz solutions. [12]

Conclusion

In the recent 2 decades, there are many research done in the area of physically based sound rendering. We can now accurately model many sound effects in the real world. However, due to the natural of the area, some areas are not very physically defined such as flutes. There are still a lot to explore in the field and I find it very interesting overall.

References

[1] P. R. Cook. Integration of physial modeling for synthesis and animation. *Proceedings of the International Computer Music Conference*, pages 525–528, 1995.

[2] Dinesh K. Pai Kees van den Doel. Synthesis of shape dependent sounds with physical modeling. *Proceedings of the Ineternational Conference on Auditory Display*, 1996.

[3] Dinesh K. Pai Kees van den Doel. The sounds of physical shapes. *Presence 7*, 1998.

[4] Christine M. Gatchalian James F. O'Brien, Chen Shen. Synthesizing sounds from rigid-body simulations. *SIGGRAPH*, 2002.

[5] Python. https://www.python.org/.

[6] sounddevice. https:://python-sounddevice.readthedocs.io/en/0.3.10/.

[7] pyglet. https://bitbucket.org/pyglet/pyglet/wiki/Home.

[8] numpy. http://www.numpy.org/.

[9] scipy. https://www.scipy.org/.

[10] Ming C. Lin Zhimin Ren, Hengchin Yeh. Example-guided physically based modal sound synthesis. *SIGGRAPH*, 2013.

[11] Jeffrey N. Chadwick, Changxi Zheng, and Doug L. James. Precomputed acceleration noise for improved rigid-body sound. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012)*, 31(4), August 2012.

[12] Dinesh K. Pai Doug L. James, Jernej Barbic. Precomputed acoustic transfer: Output-sensitive, accurate sound generation for geometrically complex vibration sources. *SIGGRAPH*, 2006.