



A Robust and Multi-Scale Modal Analysis For Sound Synthesis

Cécile Picard, François Faure, George Drettakis, Paul Kry

► To cite this version:

Cécile Picard, François Faure, George Drettakis, Paul Kry. A Robust and Multi-Scale Modal Analysis For Sound Synthesis. DAFx-09 - 12th International Conference on Digital Audio Effects, Sep 2009, Como, Italy. pp.1-7. inria-00410314

HAL Id: inria-00410314

<https://inria.hal.science/inria-00410314>

Submitted on 20 Aug 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A ROBUST AND MULTI-SCALE MODAL ANALYSIS FOR SOUND SYNTHESIS

Cécile Picard

REVES/INRIA Sophia Antipolis
Sophia Antipolis, France
cecile.picard@sophia.inria.fr

François Faure

EVASION/INRIA/LJK Rhône-Alpes
Grenoble, France
francois.faure@inrialpes.fr

George Drettakis

REVES/INRIA Sophia Antipolis
Sophia Antipolis, France
george.drettakis@sophia.inria.fr

Paul G. Kry

SOCS, McGill University
Montreal, Canada
kry@cs.mcgill.ca

ABSTRACT

This paper presents a new approach to modal synthesis for rendering sounds of virtual objects. We propose a generic method for modal analysis that preserves sound variety across the surface of an object, at different scales of resolution and for a variety of complex geometries. The technique performs automatic voxelization of a surface model and automatic tuning of the parameters of hexahedral finite elements, based on the distribution of material in each cell. The voxelization is performed using a sparse regular grid embedding of the object, which easily permits the construction of plausible lower resolution approximations of the modal model. With our approach, we can compute the audible impulse response of a variety of objects. Our solution is robust and can handle non-manifold geometries that include both volumetric and surface parts, such as those used in games, training simulations, and other interactive virtual environment.

1. INTRODUCTION

Our goal is to realistically model sounding objects for animated real-time virtual environments. To achieve this, we propose a robust and flexible modal analysis approach that efficiently extracts modal parameters for plausible sound synthesis while also focusing on efficient memory usage.

Modal synthesis models the sound of an object as a combination of sinusoids, each of which oscillates independently of the others. Modal synthesis approaches are only accurate for sounds produced by linear phenomena, but they can compute these sounds in real-time. Modal synthesis requires the computation of a partial eigenvalue decomposition of the system matrices, which is relatively expensive. For this reason, modal analysis is performed in a preprocessing step. The eigenvalues and eigenvectors strongly depend on the geometry, material and scale of the sounding object. Therefore, modeling numerous sounding objects can rapidly become prohibitively expensive. In addition, this processing step can be subject to computation problems; in particular, when the geometries are non-manifold.

We propose a new approach to efficiently extract modal parameters for any given geometry, overcoming many of the aforementioned limitations. Our method employs bounding voxels of a given shape at arbitrary resolution for hexahedral finite elements. The advantages of this technique are the automatic voxelization

of a surface model and the automatic tuning of the finite element method (FEM) parameters based on the distribution of material in each cell. A particular advantage of this approach is that we can easily deal with non-manifold geometry which includes both volumetric and surface parts. These kinds of geometries cannot be processed with traditional approaches which use a tetrahedralization of the model (e.g., [1]). Likewise, even with solid watertight geometries, complex details often lead to poorly shaped tetrahedra and numerical instabilities; in contrast, our approach does not suffer from this problem. Our specific contribution is the adaptation of the multi-resolution hexahedral embedding technique to modal analysis for sound synthesis. Most importantly, our solution preserves variety in what we call the *Sound Map*, that is, the changes in sound across the surface of the sounding object.

2. BACKGROUND

2.1. Related Work

The traditional approach to creating soundtracks for interactive physically based animations is to directly play-back pre-recorded samples, for instance, synchronized with the contacts reported from a rigid-body simulation. Due to memory constraints, the number of samples is limited, leading to repetitive audio. Moreover, matching sampled sounds to interactive animation is difficult and often leads to discrepancies between the simulated visuals and their accompanying soundtrack. Finally, this method requires each specific contact interaction to be associated with a corresponding pre-recorded sound, resulting in a time-consuming authoring process.

Work by Adrien [2] describes how effective digital sound synthesis can be used to reconstruct the richness of natural sounds. There has been much work in Computer music [3, 4] and computer graphics [1, 5, 6] exploring methods for generating sound based on physical simulation. Most approaches target sounds emitted by vibrating solids. Physically based sounds require significantly more computation power than recorded sounds. Thus, brute-force sound simulation cannot be used for real-time sound synthesis. For interactive simulations, a widely used solution is to apply vibrational parameters obtained through modal analysis. Modal data can be obtained from simulations [1, 5] or extracted from recorded sounds of real objects [6]. The technique presented in this paper is more closely related to the work of O'Brien et al. [1], which extends

modal analysis to objects that are neither simple shapes nor available to be measured.

The computation time required by current methods to preprocess the modal analysis prevents it from being used for real-time rendering. Work of Maxwell and Bindel [7] address interactive sound synthesis and how the change of the shape of a finite element model affects the sound emission. Concerning the performance of mode-based computations, a fast sound synthesis approach that exploits the inherent sparsity of modal sounds in the frequency domain has recently been introduced by Bonneel et al. [8]. Our technique tackles computational efficiency by proposing a multi-scale resolution approach of the modal analysis, managing the amount of modal data according to the memory requirements.

2.2. Modal Analysis

Modal sound synthesis is a physically based synthesis approach that consists of solving the governing equations of motion of a sounding system. The solution of these equations becomes complicated when the size of the system is large or when the forcing functions of the system are non-periodic. A system of n degrees-of-freedom is governed by a set of n coupled ordinary differential equations of second order. By expressing the deformation of the object as linear combinations of normal modes, the equations of motion are uncoupled and the solution for object vibration can be easily computed. In order to decouple the damped system into single degree-of-freedom oscillators, we assume Rayleigh damping (see, for instance, [9]). Using the finite element method, we obtain the general form of the system for eigendecomposition, from which the modal parameters, i.e., frequencies, dampings, and corresponding gains are extracted. For our approach, the calculations for modal parameters are similar to the ones presented in the paper of O'Brien et al. [1] and we refer the reader to this work for additional information.

3. METHOD

In the case of small elastic deformations, rigid motion of an object does not interact with the objects's vibrations. On the other hand, we assume that small-amplitude elastic deformations will not significantly affect the rigid-body collisions between objects. For these reasons, the rigid-body behavior of the objects can be modelled in the same way as animation without audio generation.

3.1. Deformation Model

Our implementation uses the *Sofa Framework*¹ for rigid-body simulation. This choice was motivated by the ease with which it could be extended for our purpose. The main feature of SOFA compared with other libraries is its high flexibility while maintaining efficiency. SOFA is an open-source C++ library for physical simulation. It can be used as an external library in another program, or using one of the associated GUI applications. It allows the use of multiple interacting geometrical models of the same object, typically, a mechanical model with mass and constitutive laws and a collision model with simple geometry. A visual model with detailed geometry and rendering parameters is also integrated, where each model can be designed independently of the others. During run-time, consistency is maintained using mappings. Additionally,

¹Simulation Open Framework Architecture;
<http://www.sofa-framework.org/>

SOFA scenes are modeled using a data structure similar to hierarchical scene graphs which allows the physical objects to be split easily into collections of independent components, each describing one feature of the model. Moreover, simulation algorithms are also modeled as components in the scene graph, providing us with the same flexibility for algorithms as for models.

Elastic deformations are used to generate the audio signal. Before performing modal decomposition, we must first select a deformable modeling method that can be used to generate the stiffness and the mass matrices of the mechanical system. A variety of methods could be used, including particle systems [5] or finite differences methods. The tetrahedral finite element method has also been used [1]. However, tetrahedral meshes are computationally expensive for complex geometries, and can be difficult to tune. As an example, in the tetrahedral mesh generator *Tetgen*², the mesh element quality criterion is based on the minimum radius-edge ratio, which limits the ratio between the radius of the circumsphere of the tetrahedron and the shortest edge length.

Our method is inspired from work by Nesme et al. [10]. It uses hexahedral finite element for computing the mass and stiffness matrices of the mechanical system. The technique can be summarized as follows. An automatic high-resolution voxelization of the geometric object is first built. The voxelization initially concerns the surface of the geometric model, while the interior is automatically filled when the geometry represents a solid object. The voxels are then recursively merged up to an arbitrary coarser mechanical resolution. The merged voxels are used as hexahedral finite elements embedding the detailed geometrical shape. At each level, the mass and stiffness of a merged voxel are deduced from its eight children, using a weighted average that takes into account the distribution of material. With this method, we can handle objects with geometries that simultaneously include volumetric and surface parts; thin or flat features will occupy voxels and will thus result in the creation of mechanical elements that approximate their shape (see Section 4.1).

We extend the method for microscopic deformations that allows sound rendering. Thus, in order to compute the modal parameters, we compute the assembled mass and the assembled stiffness matrices for the object by summing the contribution of each cell. Then, we solve the decoupled system to extract the modal parameters as explained in Section 2. Our preprocessing step that performs modal analysis can be summarized as follows.

ALGORITHM 1. Algorithm for modal parameters extraction.

1. *Compute mass and stiffness at desired mechanical level*
2. *Assemble the mass and the stiffness matrices*
3. *Modal analysis: solve the eigenproblem*
4. *Store eigenvalues and eigenvectors for sound synthesis*

The model approximates the motion of the embedded mesh vertices. That is, the visual model with detailed geometry does not match the mechanical model on which the modal analysis is performed. The motion of the embedding uses a trilinear interpolation of the mechanical degrees of freedom (DOFs), so we can nevertheless compute the motion of any point on the surface given the mode shapes.

3.2. Sound Generation

When rendering the sound with a modal synthesis approach, we do not solve the emission problem, but instead we consider the sound

²<http://tetgen.berlios.de/>

to be simply a sum of damped sinusoids. The activation of this model depends on where the object is hit. If we hit the object at a vibration node of a mode, then that mode will not vibrate, but others will. This is what we refer to as the *Sound Map*, which could also be called a sound excitation map as it indicates how the different modes are excited when the object is struck at different locations.

The sound resulting from an impact on a specific location on the surface is calculated as a sum of n damped oscillators:

$$s(t) = \sum_n^1 a_i \sin(w_i t) e^{-d_i t} \quad (1)$$

where w_i , d_i , and a_i are respectively the frequency, the decay rate and the gain of the mode i .

In our method, we synthesize the sounds via a reson filter (see, for example, Van den Doel et al. [6]). This choice is made based on the effectiveness for real-time audio processing. No radiation properties are considered; our study focuses specifically on effective modal synthesis. However, radiation can be computed in a number of ways [11, 12]. As the motions of objects are computed with modal analysis, surfaces can be easily analyzed to determine how the motion will induce acoustic pressure waves in the surrounding medium. Finally, our study does not consider contact-position dependent damping or changes in boundary constraints, as might happen during moments of excitation. Instead we use a uniform damping value for the sounding object.

3.3. Position Dependent Sound Rendering

To properly render impact sounds of an object, the method must preserve the sound variety when hitting the surface at different locations. For example, consider the metal bowl, modeled by a triangle mesh with 274 vertices, shown in Figure 1.

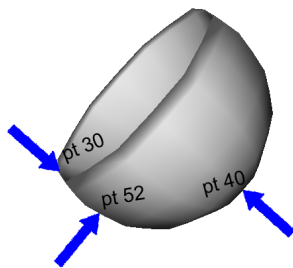


Figure 1: A sounding metal bowl: sound synthesis is performed for excitation on specific locations on the surface: points 30, 40 and 52.

Figure 1 specifies where the bowl is hit. We take 3 different locations, i.e., top, side and bottom, on the surface of the object where the object is impacted. The excitation force is modelled as a dirac, such as a regular impact. The material of the bowl is aluminium, with the parameters 69×10^9 for Young Modulus, 0.33 for Poisson coefficient, and 2700 kg/m^3 for the volumic mass. The Rayleigh damping parameters for stiffness and mass are set to 3×10^{-7} and 10. The use of a constant damping ratio is a simplification that still produces good results.

We compare our approach to modal analysis performed first using tetrahedralization with *Tetgen*³ with 822 modes. Our method

³Tetrahedral Mesh Generator: <http://tetgen.berlios.de/>

uses hexahedral finite elements and is applied with a grid of $2 \times 2 \times 2$ cells, leading to 81 modes. However, to adapt the stiffness of a cell according to its content, the mesh is refined more precisely than desired for the animation. The information is propagated from fine cells to coarser cells. For this example, the elements of the $2 \times 2 \times 2$ coarse grid resolution approximates mechanical properties propagated from a fine grid of $4 \times 4 \times 4$ cells.

The frequency content of the sound resulting from impact at the 3 locations on the surface is shown in Figure 2.

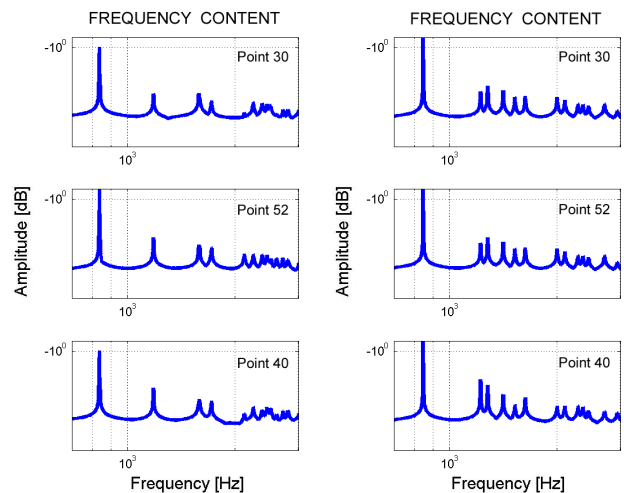


Figure 2: Sound synthesis with a modal approach using classical tetrahedralization with 822 modes (left) and our method with a $2 \times 2 \times 2$ hexahedral FEM resolution, leading to 81 modes (right): power spectrum of the sounds emitted when impacting at the 3 different locations shown in Figure 1, (from top to bottom) points 30, 52 and 40.

In Figure 2, each power spectrum is normalized with the maximum amplitude in order to factor out the magnitude of the impact. The eigenvalues that correspond to vibration modes will be nonzero, but for each free body in the system there will be six zero eigenvalues for the body's six rigid-body freedoms. Only the modes with nonzero eigenvalue are kept. Thus, 816 modes are finally used for sound rendering with the tetrahedralization method and 75 with our hexahedral FEM method.

The movie provided⁴ compares the sounds synthesized with the tetrahedral FEM and the hexahedral FEM approaches. While Figure 2 highlights the visual differences in the frequency content, we notice in listening to the synthesized sounds that those generated by our method are quite similar to those created with the standard tetrahedralization, even when significantly fewer vibration modes are used (i.e., 75 in contrast to 816).

4. ROBUSTNESS AND MULTI-SCALE RESULTS

Computing modes for complex geometries can become prohibitively expensive especially when numerous sounding objects have to be processed. As an example, the actual cost of computing the partial eigenvalue decomposition using a tetrahedralization in the

⁴Additional material:
<http://www-sop.inria.fr/reves/Cecile.Picard>

case of a bowl with 274 vertices and generating 2426 tetrahedras is 5 minutes with an Intel Core Duo with 2.33 GHz and 2 GB of memory. The number of tetrahedras determine the dimension of the system to solve. To avoid this expense, we provide a method that greatly simplifies the modal parameter extraction even for non-manifold geometries that include both volumetric and surface parts. Our technique consists of using multi-resolution hexahedral embeddings.

4.1. Robustness

Most approaches for tetrahedral mesh generation have limitations. In particular, an important requirement imposed by the application of deformable FEM is that tetrahedra must have appropriate shapes, for instance, not too flat or sharp. By far the most popular of the tetrahedral meshing techniques are those utilizing the Delaunay criterion [13]. When the Delaunay criterion is not satisfied, modal analysis using standard tetrahedralization is impossible. In comparison with tetrahedralization methods, our technique can handle complex geometries and adequately performs modal analysis. Figure 3 gives an example of problematic geometry for tetrahedralization because of the presence of very thin parts, specifically the blades that protrude from either side.

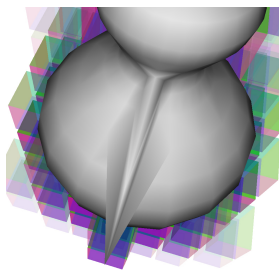


Figure 3: An example of a complex geometry that can be handled with our method. The thin blade causes problems with traditional tetrahedralization methods.

We suppose the object is made of aluminium (see Section 3.3 for the material parameters). We apply a coarse grid of $7 \times 7 \times 7$ cells for modal analysis. The coarse level encloses the mechanical properties of a fine grid of $14 \times 14 \times 14$ cells. Figure 5 shows the power spectrum of the sounds resulting from impacts, modelled as a dirac, on 5 different locations. Each power spectrum is normalized with the maximum amplitude of the spectrum in order to factor out the magnitude of the impact.

Figure 5 shows that the *Sound Map* is preserved; we can observe that the different modes have varying amplitude depending on the location of excitation. It is interesting to examine the quality of the sound rendered when hitting the wings. Because this part is lightweight compared to the rest of the object, the amplitude of higher frequencies is more pronounced than at other locations.

4.2. A Multi-scale Approach

To study the influence of the number of hexahedral finite elements on the sound rendering, we model a sounding object with different resolutions of hexahedral finite elements. We have created a squirrel model with 999 vertices which we use as our test sounding object. Its material is pine wood, which has parameters 12×10^9 for Young Modulus, 0.3 for Poisson coefficient, 750 kg/m^3 for

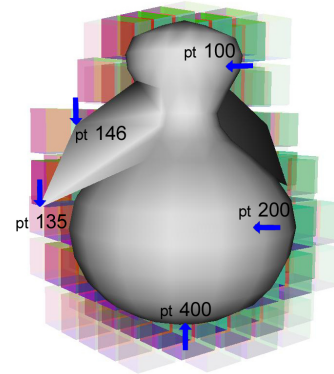


Figure 4: Test impacts for sound generation are simulated on 5 different locations on the surface of the complex geometry: points 100, 135, 146, 200 and 400.

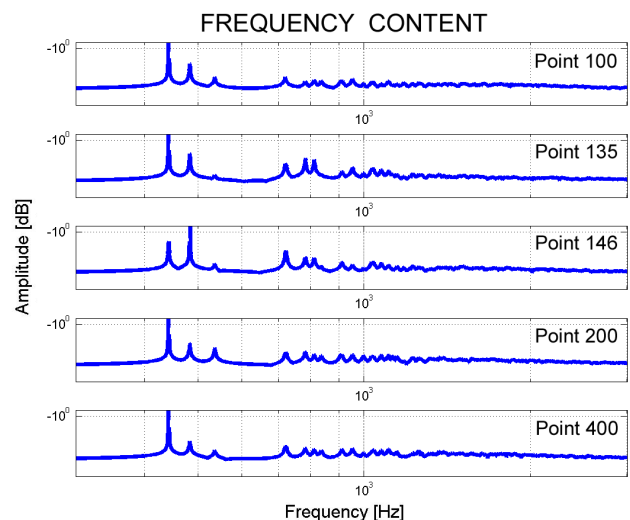


Figure 5: The power spectrum of the sounds resulting from impacts at the 5 different locations shown in Figure 4: (from top to bottom) points 100, 135, 146, 200 and 400. Note that the audible response is different based on where the object is hit.

the volumetric mass. Rayleigh damping parameters for stiffness and mass are set to 8×10^{-6} and 50 respectively. Sound synthesis is performed for 3 different locations of excitation, see Figure 6 (top left). The coarse grid resolution for finite elements is set to $2 \times 2 \times 2$, $3 \times 3 \times 3$, $4 \times 4 \times 4$, $8 \times 8 \times 8$ and $9 \times 9 \times 9$ cells. In this example, the finer grid resolution is one level up to the one of coarse grid, that is, a coarse grid of $2 \times 2 \times 2$ cells has a fine level of $4 \times 4 \times 4$ cells.

Results show that the frequency content of sounds depend on the location of excitation and on the resolution of the hexahedral finite elements. The higher resolution models have a wider range of frequencies because of the supplementary degrees of freedom. We also observe a frequency shift as the FEM resolution increases. Note that a $2 \times 2 \times 2$ grid represents an extremely coarse embedding, and consequently it is not surprising that the frequency content is different at higher resolution. Nevertheless, there are still

some strong similarities at the dominant frequencies. Above all, an important feature is the convergence in frequency content as the FEM increases. According to Figure 6, a grid of $4 \times 4 \times 4$ cells may be sufficient to properly render the sound quality of the object.

4.3. Limitations

The *Sound Map* is influenced by the resolution of the hexahedral finite elements. This is related to the way stiffnesses and masses of different elements are altered based on their contents. As a consequence, a $2 \times 2 \times 2$ hexahedral FEM resolution would show much less expressive variation than higher FEM resolution. This is shown in the movie⁵. One approach to improving this would be to use better approximations of the mass and stiffness of coarse elements [14].

Nevertheless, based on the quality of the resulting sounds, and given that increased resolution for the finite elements implies higher memory and computational requirements for modal data, finite elements resolution can be adapted to the number of sounding objects in the virtual scene.

5. DISCUSSION

Table 1 gives the computation time and the memory usage of the modal data when computing the modal analysis with different FEM resolution on the squirrel model. In this example, the finer grid resolution is one level up to the one of coarse grid, that is, a coarse grid of $4 \times 4 \times 4$ cells has a fine level of $8 \times 8 \times 8$ cells. These are com-

Coarse Grid Resolution (cells)	Computation Time (seconds)	Memory Usage (MB)
$7 \times 7 \times 7$	115.11	11.309
$6 \times 6 \times 6$	39.14	5.698
$5 \times 5 \times 5$	12.98	2.663
$4 \times 4 \times 4$	5.35	1.042

Table 1: *Computation time and memory usage for different grid resolutions.*

putation times of our unoptimized initial implementation on a 2.33 GHz Intel Core Duo.

Despite the fact that audio is considered a very important aspect in virtual environments, it is still considered to be of lower importance than graphics. We believe that physically modeled audio brings a significant added value in terms of realism and the increased sense of immersion.

The use of physics engines is becoming much more widespread for animated interactive virtual environments; the interface between these engines and audio has often been one of the obstacles for the adoption of physically based sound synthesis in simulations. This is often due to the lack of appropriate design choices in the two interfaces that prevent them from working together effectively.

Our method is built on a physically based animation engine, *Sofa Framework*. As a consequence, problems of coherence between physics simulation and audio are avoided by using exactly the same model for simulation and sound modeling.

⁵Additional material:
<http://www-sop.inria.fr/reves/Cecile.Picard>

6. CONCLUSION

We propose a new approach to modal analysis using automatic voxelization of a surface model and automatic tuning of the finite elements parameters, based on the distribution of material in each cell. Our goal is to perform sound rendering in the context of an animated real-time virtual environment, which has specific requirements, such as real-time processing and efficient memory usage.

We have shown that in simple cases our method gives similar results as traditional modal analysis with tetrahedralization for simple cases. For more complex cases, our approach provides plausible results. In particular, sound variety along the object surface, the *Sound Map*, is well preserved.

Our technique can handle complex non-manifold geometries that include both volumetric and surface parts, which can not be handled by previous techniques. We are thus able to compute the audio response of numerous and diverse sounding objects, such as those used games, training simulations, and other interactive virtual environment.

Our solution allows a multi-scale solution because the number of hexahedral finite elements only loosely depends on the geometry of the sounding object.

Finally, since our method is built on a physics animation engine, the *Sofa Framework*, problems of coherence between simulation and audio can be easily addressed, which is of great interest in the context of interactive environment.

In addition, due to the fast computation time, we are hopeful that real-time modal analysis will soon be possible on the fly, with sound results that are approximate but still realistic for virtual environments.

7. ACKNOWLEDGMENTS

This work was partly funded by Eden Games⁶, an *ATARI* Game Studio in Lyon, France. We would like to thank Nicolas Tsingos for his input on an early draft.

8. REFERENCES

- [1] James F. O'Brien, Chen Shen, and Christine M. Gatchalian, "Synthesizing sounds from rigid-body simulations," in *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, New York, NY, USA, 2002, pp. 175–181.
- [2] Jean-Marie Adrien, "The missing link: modal synthesis," *Representations of musical signals*, pp. 269–298, 1991.
- [3] Perry R. Cook, *Real Sound Synthesis for Interactive Applications*, A. K. Peters, 2002.
- [4] Francisco Iovino, René Caussé, and Richard Dudas, "Recent work around modalys and modal synthesis," in *ICMC: International Computer Music Conference*, Thessaloniki Hellas, Greece, September 1997, pp. 356–359.
- [5] Nikunj Raghuvanshi and Ming C. Lin, "Interactive sound synthesis for large scale environments," in *SI3D'06: Proceedings of the 2006 symposium on Interactive 3D Graphics and Games*, 2006, pp. 101–108.

⁶Video games studio; <http://www.eden-games.com/>

- [6] Kees van den Doel, Paul G. Kry, and Dinesh. K. Pai, "Foley automatic: physically-based sound effects for interactive simulation and animation," in *Proc. SIGGRAPH '01*, New York, NY, USA, 2001, pp. 537–544.
- [7] C. B. Maxwell and D. Bindel, "Modal parameter tracking for shape-changing geometric objects," in *DAFx '07: Proceedings of the 10th International Conference on Digital Audio Effects*, 2007.
- [8] Nicolas Bonneel, George Drettakis, Nicolas Tsingos, Isabelle Viaud-Delmon, and Doug James, "Fast modal sounds with scalable frequency-domain synthesis," in *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, New York, NY, USA, 2008, pp. 24:1–24:9.
- [9] Klaus-Juergen Bathe, *Finite element procedures in engineering analysis*, Prentice-Hall, New Jersey, 1982.
- [10] Matthieu Nesme, Yohan Payan, and François Faure, "Animating shapes at arbitrary resolution with non-uniform stiffness," in *Eurographics Workshop in Virtual Reality Interaction and Physical Simulation (VRIPHYS)*, Madrid, Nov 2006, Eurographics.
- [11] James F. O'Brien, Perry R. Cook, and Georg Essl, "Synthesizing sounds from physically based motion," in *Proceedings of ACM SIGGRAPH 2001*, Aug. 2001, pp. 529–536.
- [12] Doug L. James, Jernej Barbic, and Dinesh K. Pai, "Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 987–995, July 2006.
- [13] Jonathan Richard Shewchuk, "A condition guaranteeing the existence of higher-dimensional constrained delaunay triangulations," in *SCG '98: Proceedings of the fourteenth annual symposium on Computational geometry*, New York, NY, USA, 1998, pp. 76–85.
- [14] Matthieu Nesme, Paul G. Kry, Lenka Jeřábková, and François Faure, "Preserving topology and elasticity for embedded deformable models," in *ACM Transactions on Graphics (Proc. of SIGGRAPH)*. ACM, August 2009.
- [15] Kees van den Doel and Dinesh. K. Pai, "Synthesis of shape dependent sounds with physical modeling," in *Proceedings of the International Conference on Auditory Display (ICAD96)*, S. P. Frysinger and G. Kramer, Eds., Palo Alto, CA, U.S., 1996, International Community for Auditory Display, International Community for Auditory Display.
- [16] "SOFA: Simulation Open Framework Architecture," Available at <http://www.sofa-framework.org/>.
- [17] "Tetgen: Tetrahedral Mesh Generator," Available at <http://tetgen.berlios.de/>.
- [18] "Eden Games: Video games studio," Available at <http://http://www.eden-games.com/>.

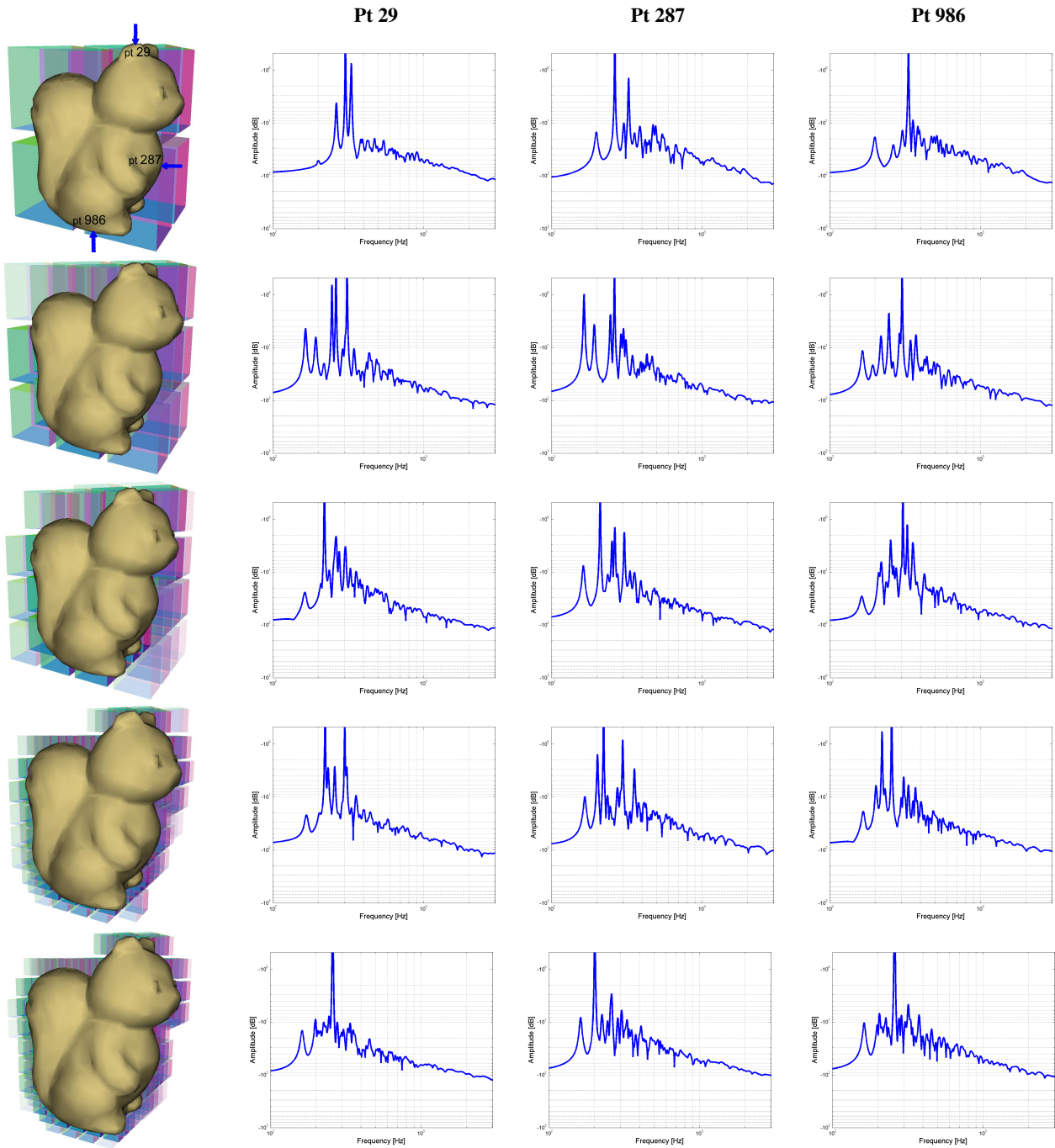


Figure 6: A squirrel in pine wood is sounding when impacting on 3 different locations: points 29, 287 and 986 (from left to right). Frequency content of the resulted sounds with 5 different resolutions for the hexahedral finite elements: (from top to bottom), $2 \times 2 \times 2$, $3 \times 3 \times 3$, $4 \times 4 \times 4$, $8 \times 8 \times 8$ and $9 \times 9 \times 9$ cells.