Designing Custom-made Metallophone with Concurrent Eigenanalysis*

Nobuyuki Umetani
The University of Tokyo
JST ERATO
7-3-1 Hongo, Bunkyo-ku,
Tokyo, 113-0033 Japan
umetani@ui.is.s.u-tokyo.ac.jp

Jun Mitani
University of Tsukuba
JST ERATO
Tsukuba Ibaraki
305-8573 Japan
mitani@cs.tsukuba.ac.jp

Takeo Igarashi
The University of Tokyo
JST ERATO
7-3-1 Hongo, Bunkyo-ku,
Tokyo, 113-0033 Japan
takeo@acm.org

ABSTRACT

We introduce an interactive interface for the custom design of metallophones. The shape of each plate must be determined in the design process so that the metallophone will produce the proper tone when struck with a mallet. Unfortunately, the relationship between plate shape and tone is complex, which makes it difficult to design plates with arbitrary shapes. Our system addresses this problem by running a concurrent numerical eigenanalysis during interactive geometry editing. It continuously presents a predicted tone to the user with both visual and audio feedback, thus making it possible to design a plate with any desired shape and tone. We developed this system to demonstrate the effectiveness of integrating real-time finite element method analysis into geometric editing to facilitate the design of custom-made musical instruments. An informal study demonstrated the ability of technically unsophisticated user to apply the system to complex metallophone design.

Keywords

Modeling - Modeling Interfaces, Modeling - Geometric Modeling, Modeling - CAD, Methods and Applications - Education, Real-time FEM

1. INTRODUCTION

Each acoustic musical instrument has its own typical shape and appearance. Although the exterior may show subtle differences, the fundamental shape cannot be very different (e.g., metallophone plates are always rectangular). Although these shapes have become sophisticated through years of refinement, the appearance of musical instruments can be repetitive and characterless. Conveying character, or a message, through appearance is a major challenge for the makers of acoustic musical instruments. Because the shapes of acoustic musical instruments and their tones are inseparable, making designing them much more complex, acoustic musical instruments cannot have individual designs like electric guitars.

*The supplemental video can be seen at `http://www.youtube.com/watch?v=7TfMgwuDHS4`

NIME2010, June 15-18, 2010, Sydney, Australia

Figure 1: A custom designed metallophone in the shape of a fish.

We propose a system for designing original acoustic musical instruments with computational assistance. We utilize the finite element method (FEM) to predict tone. Although the standard FEM runs on a fixed initial geometry, our idea is to continuously update the simulation results in response to the designer's shape modifications. Real-time feedback during editing provides the guiding principles for creating a better overall design. It assists the designer in obtaining satisfactory design results without the many trial-and-error iterations that would be necessary with an offline simulation. We believe that responsive feedback can also be useful for educational purposes.

The specific object that illustrates our concept is the metallophone. We define a metallophone as a musical instrument that produces sounds via the vibration of metal plates struck by a mallet. Metallophone plates are usually rectangular because this practical shape makes it possible to analytically predict the instrument's tone [5]. Designing a metallophone with an arbitrary shape is difficult because of the complex relationship between shape and tone. We believe that our approach is well suited to designing a metallophone of desired artistic shape and tone because this kind of highly constrained modeling naturally requires a tight integration of design and analysis.

This paper is organized as follows. We first outline previous work in this area. In the next section, we describe the user interface and implementation details of the prototype system. We then address system performance and user experience. Finally, we discuss the limitations of the current system and the potential for future work.

2. RELATED WORK

FEM analysis of acoustic musical instrument is an established field of research. There have been a number of studies on the use of FEM eigenanalysis for predicting the tones of various musical instruments, such as piano [9], guitar [2], and xylophone [3]. Shoofs et al. [15] used FEM eigenanalysis to determine the optimal shape of carillon bells. However, eigenanalysis have previously been used exclusively as an offline evaluation tool and has not been directly integrated into geometry editing as in our system

Automatic optimization methods can be used to design objects that satisfy physical constraints. For example, Smith et al. [16] applied an optimization approach to the design of truss structures. However, automatic methods present many practical difficulties, such as explicitly specifying constraints and parameter spaces that are too large. The interactive approach offers the advantage of allowing users to use their own preferences and judgment during the design process while considering less tangible factors such as aesthetics.

Various techniques have been suggested for making physical simulations interactive. One approach is to use precomputations. James and Pai [6] performed an interactive physical simulation of deformable objects by precomputing the matrices of reference boundary value problems. They also proposed modal analyses of dynamic elastic models for real-time deformation simulations [7]. Another approach is to use approximations for nonlinear elasticity to achieve large deformations quickly and stably [12]. These methods require user input as an external force in a continuously running simulation. In our system, the user directly modifies the definition of the target problem, and the system reruns the simulation with the new setting.

Our goal is to aid the design process by using physical simulations. Our system shares some concepts with first-order analysis [13], which emphasizes the importance of using simulation in the early stages of a design. Masry and Lipson [10] develop a sketch-based three-dimensional (3D) modeling interface capable of FEM analysis that has a similar objective to our own. However, first-order analysis is not very different from conventional computer-aided design and computer-aided engineering systems, in that the analysis is performed *after* some of the design processes have already occurred. In contrast, in our own system the analysis is performed *during* the design process. Several systems have been proposed for designing physical objects such as stuffed animals with the aid of interactive simulations [11]. However, the focus of those systems is mainly the user interface, and they involve relatively simple simulation procedures.

3. USER INTERFACE

This section describes our metallophone design system from the user's point of view. The system basically operates as a two-dimensional (2D) modeling program in which the user interactively edits the shape of a metallophone plate using direct manipulation (Figure 3). FEM eigenfrequency analysis runs concurrently with the user manipulation and presents the predicted tone to the user with visual and audio feedback. The system also provides 3D graphics of the geometric deformation of the plate during vibration. In this way, the user, guided by artistic inspiration, can explore various plate shapes while continuously verifying that the desired tone is produced.

Figure 3 shows a set of modeling operations provided by our system. A metallophone plate is represented as a closed area surrounded by straight lines and arcs, connected by corner vertices. The user drags the corner vertices to move them, and drags the arc to change its radius. The user can also add and delete corner vertices. Our current implementation does not support plates with holes. More complex curves, such as Bézier curves or NURBS, permit the creation of more diverse shapes; this is left to future development.

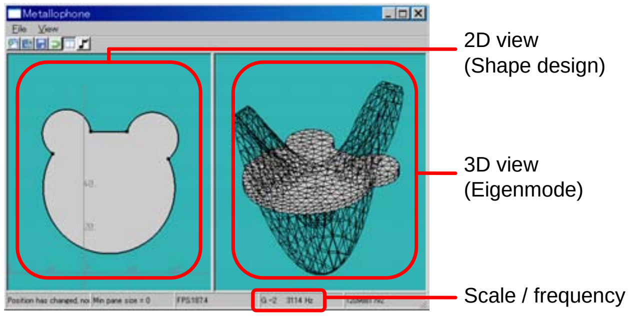

Figure 2: Metallophone design software. The left window is used for the original 2D design, whereas the right window shows the analyzed eigenmode as a 3D graphic. The tone is updated in real time with both audio and visual feedback for the user.

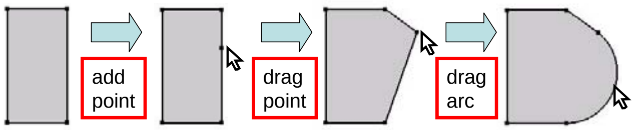

Figure 3: The modeling operations of our software: adding and deleting a point, dragging an arc and a point.

The system continuously predicts the tone that the plate will produce when struck with a mallet and presents it to the user during the editing process. Visual feedback is given as a numerical value (Figure 2, bottom) and audio feedback is provided in the form of a sine wave, with the speaker emitting the predicted tone. As the user drags the corner vertex, the tone from the speaker and the numerical value on the screen gradually change. Audio feedback allows the user to continuously monitor the tone while editing, and visual feedback is useful for verifying the exact value. Low tones tend to be produced by large plates (and vice versa), and the user can observe and learn this phenomenon during interactive editing, which facilitates the overall design process.

The system also predicts the geometric deformation of the plate during vibration and visually presents it to the user as a 3D graphic (Figure 2, right). The actual deformation is too small to be visible, so the system exaggerates it in the visualization. This helps the user decide where the plate should be attached to the base, because the plate should be attached at a point of minimal deformation. It is also useful for visually evaluating the quality of the vibrations. Vibrations perpendicular to the plate are desirable because they are excited when the mallet strikes the face vertically, whereas a plane vibration is not likely to occur during a performance, and its tone is not dominant (Figure 4). Thus, the user should strive for a shape that produces perpendicular vibrations.

The system only supports the design of individual plate geometry, and physical construction must be done manually (Figure 5). The user may export the plate geometry to a DXF file and then send it to a wire-electrical discharge machine for the actual cutting of the metal plate. The metal plate can also be cut manually with a band saw. Some errors always occur during this process, and the finished

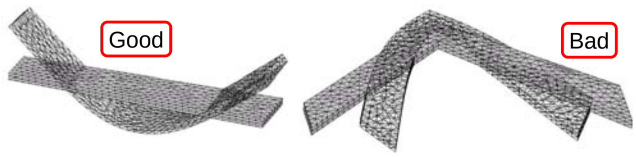

Figure 4: Perpendicular vibration mode is desirable (left), whereas in plane vibration mode is undesirable (right)

metal plate will not produce exactly the same tone as the simulated plate. It is therefore necessary to adjust the tone by rasping the plate. Our system is also useful in this regard, because it is able to predict how the tone will change when particular edges are rasped. Finally, the user attaches the plate to a wooden board at the point suggested by the vibration shape analysis.

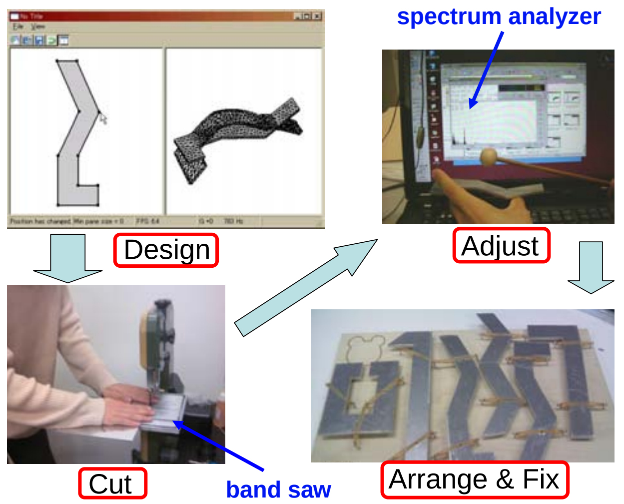

Figure 5: The process of producing a metallophone: first, cutting the metal plate; second, adjusting the tone by rasping; finally, attaching the plate to a wooden board.

4. IMPLEMENTATION

4.1 FEM model for eigenmode analysis

In our current implementation, a metallophone plate is modeled as a thin 3D linear solid plate extruded from a 2D shape designed by the user. We discretize the solid into tetrahedral elements by first triangulating the 2D shape, then extruding each triangle to obtain a 3D prism, and finally dividing each prism into tetrahedra. We then apply FEM eigenvalue analysis to compute its tone, or eigenfrequency. The deformation caused by the vibration is computed as the eigenmode corresponding to the eigenvalue. Note that the simulation parameters must be calibrated for each specific material. We use aluminum plate with a thickness of 4mm. The simulation accuracy is very sensitive to the mesh density, because we use linear tetrahedral elements to represent a bend in a 3D plate. To alleviate this problem, known as "shear-locking" [17], we must prevent excessive distortions of the tetrahedral elements. We divide the plate into two layers, each filled with tetrahedral elements (Figure 6), such that the longest edge is always shorter than two times the shortest edge. Thick plate elements might work very well here in place of tetrahedral elements, but we have chosen to use tetrahedral elements because they are simple to implement and yield sufficiently accurate results. The use of thick plate elements will be a subject of future research.

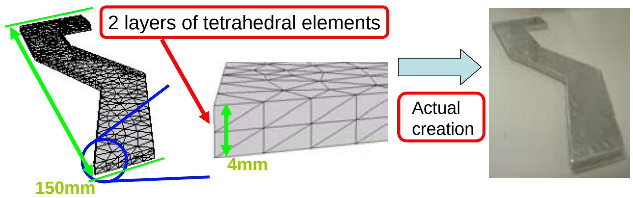

Figure 6: The mesh used in FEM discretization. We use double-layered tetrahedral elements extruded from a triangular 2D mesh.

4.2 Mesh Deformation

To continuously simulate a tone during a user's shape editing, the system adjusts the mesh to fit the shape. To adjust the mesh to fit a user-specified shape, the system first places the nodes on the boundary and then applies smoothing to the nodes inside the boundary. We use Laplacian smoothing if the mesh distortion is small (Figure 7). When the change in the boundary is large and the mesh is distorted, we employ Delaunay smoothing to suppress the mesh distortion.

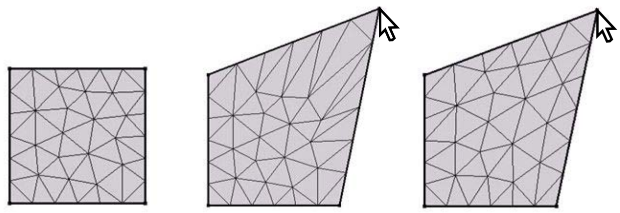

Figure 7: The smoothing of mesh using Laplacian smoothing

4.3 Eigenmode analysis of the metallophones

We assume that the metallophone is floating in a gravity-free space without any external forces or fixed boundaries (which makes the stiffness matrix singular). Generalized eigenanalysis with a singular stiffness matrix has been investigated in the field of aerospace engineering [4, 8]. A typical method of obtaining an eigenvalue and its eigenmode consists of first shifting the stiffness matrix with the mass matrix, then calculating the distribution of the eigenvalues using the Lanczos or Jacobi method to obtain the lowest nonzero eigenvalue, and finally calculating the eigenmode by shifted inverse iteration [14]. However, the extensive calculation of the distribution of all of the eigenvalues is unnecessary for our purpose. Hence, we compute only the smallest nonzero eigenvalue, which enables real-time performance. We also take advantage of the problem setting. First, we already know that the kernel of the coefficient matrix consists of translations and rotations. Second, we can reuse the solution from the previous configuration as a good initial guess for the inverse iteration.

Here is a detailed derivation of the algorithm we use. The eigenvalue problem is formulated using the FEM discretization as

$$\bar{\mathbf{M}}\ddot{\boldsymbol{u}} + \mathbf{K}\boldsymbol{u} \quad = \quad \mathbf{0} \qquad (1)$$

where $\boldsymbol{u}$ is the nodal displacement vector, $\bar{\mathbf{M}}$ is the lumped mass matrix, and $\mathbf{K}$ is the positive semi-definite stiffness matrix. By splitting the displacement $\boldsymbol{u}$ into the product of a spatially varying amplitude $\boldsymbol{\phi}$ and a harmonic oscillation with angular velocity ω, $\boldsymbol{u}(\boldsymbol{x}, t) = \boldsymbol{\phi}(\boldsymbol{x})e^{i\omega t}$, and substituting this into Eq.(1), we obtain

$$\mathbf{K}\boldsymbol{\phi} = \lambda\bar{\mathbf{M}}\boldsymbol{\phi} \tag{2}$$

where the eigenfrequency $f = \omega/(2\pi) = \sqrt{\lambda}/(2\pi)$. Our goal is to calculate the smallest nonzero eigenvalue λ and its corresponding eigenvector $\boldsymbol{\phi}$.

Cholesky factorization is employed to represent the lumped mass matrix as $\bar{\mathbf{M}} = \mathbf{L}\mathbf{L}^{\mathbf{T}}$, and both sides of Eq.(2) are multiplied on the left by $\mathbf{L}^{-1}$ to obtain the standard eigenvalue problem

$$\mathbf{A}\boldsymbol{\psi} = \lambda\boldsymbol{\psi} \tag{3}$$

where $\mathbf{A} = \mathbf{L}^{-1}\mathbf{K}\mathbf{L}^{-\mathbf{T}}$ and $\boldsymbol{\psi} = \mathbf{L}^{\mathbf{T}}\boldsymbol{\phi}$. We solve this by inverse iteration. The standard iteration procedure is modified by adding a step that removes all zero eigenvectors of $\mathbf{A}$ from the current solution. Because of our problem setting, we already know that the zero eigenvectors of $\mathbf{K}$ are $\boldsymbol{\phi}_0^i$ $(i = 1 \dots 6)$, the translations along the three coordinate axes and the rotations around them. We then apply modified Gram-Schmidt orthogonalization to $\mathbf{L}^{\mathbf{T}}\boldsymbol{\phi}_0^i$ $(i = 1 \dots 6)$ to obtain the orthonormal basis vectors $\boldsymbol{\psi}_0^i$ $(i = 1 \dots 6)$ that span the kernel of $\mathbf{A}$, and define a projection $\mathcal{P}$ that maps a vector $\boldsymbol{v}$ to the complement space of the kernel of $\mathbf{A}$ by $\mathcal{P}(\boldsymbol{v}) = \boldsymbol{v} - \sum \boldsymbol{\psi}_0^i \left(\boldsymbol{\psi}_0^i \cdot \boldsymbol{v}\right)$. In each step of the iteration, we apply this projection to the solution vector and normalize it. We add a small positive number ε to the diagonals of $\mathbf{A}$ to improve the numerical conditions. Once the shifted eigenvalue λ_1' and its corresponding eigenvector $\boldsymbol{\psi}_1$ of $\mathbf{A}$ are computed, we finally obtain the smallest nonzero eigenvalue $\lambda_1 = \lambda_1' - \varepsilon$ and the eigenvector $\boldsymbol{\phi}_1 = \mathbf{L}^{-\mathbf{T}}\boldsymbol{\psi}_1$. Reusing of the solution from the previous configuration significantly improves the convergence of the inverse iterations. The processing returns to the main thread in each iteration step to avoid freezing to user input.

5. RESULTS

5.1 System Performance

Table 1 summearizes the performance of our system. Because the performance depends on user operations, we measured the average number of frames per second during the continuous editing of shapes. In our current implementation, the number of elements is proportional to the area of the plate, because we have limited the edge length to avoid element distortions, as described in Section 4. This causes a slight slowdown when the shape is large, but we consider it acceptable.

Size (mm)	#Tetrahedra	FPS	Frequency (Hz)
100 × 30	2196	10.7	1931
150 × 30	3192	4.2	860
200 × 30	4524	3.3	494

Table 1: The performance of our system during interactive mnipulation, tested with a 2.5GHz CPU and 2.0GB RAM. The first column shows the size of the rectangular metallophone plates, the second shows the number of tetrahedral elements, the third shows the frame per second, the fourth shows the frequencies (Hz).

5.2 User Experience

We asked a professional artist to design a metallophone using our software. The artist was allowed to work on the task freely and without time limitations and was provided instructions when required. We then used an electoric discharge machine to manufacture an actual metallophone based on the artist's design. Figure 8 shows the shapes designed to correspond to the musical scale notes from C (523 Hz) to B (987 Hz). After the design of each piaces, we trimed and assembled the metallophone (Figure 1)

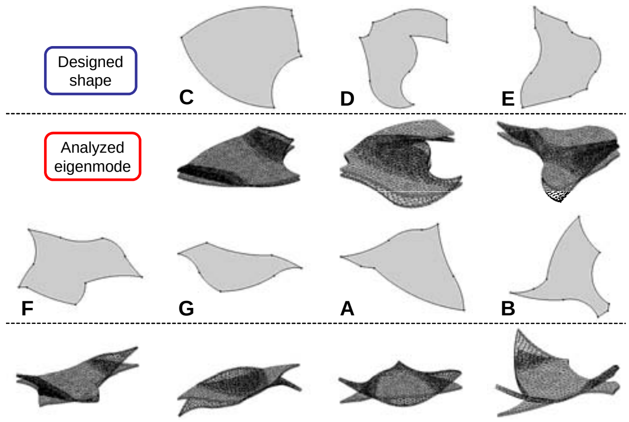

Figure 8: Metallophone shapes designed by an artist. The upper row shows the designed 2D shapes, whereas the lower row shows their analyzed eigenmodes in 3D.

The top three rows in Table 2 show that the frequencies of most of the pieces conformed well to one another for the target, the simulation, and the actual metallophone. To further improve the quality, we manually adjusted the tones of the actual metallophone pieces by trimming their edges (except for the piece corresponding to F). As mentioned in Section 3, our metallophone design software was also useful for this adjustment process, because it was able to predict the tone changes caused by edge trimming. The bottom row in Table 2 lists the frequencies of the actual metallophone after manual adjustment. Although inharmonic overtones make the timbre unclear, we found that the sounds produced by an actual metallophone were of acceptable quality for a hobbyist.

Scale	C	D	E	F	G	A	B
Targeted	523	587	659	698	783	880	987
Simulated	525	588	661	699	786	880	989
Measured	506	604	621	698	787	860	993
Adjusted	523	587	659	698	783	881	987

Table 2: Target, simulated, measured, and adjusted frequencies (Hz) of the metallophone for each note in the scale, illustrating the accuracy of our analysis.

We interviewed the artist afterward to obtain subjective feedback. The artist reported that the design took roughly 5 hours to complete, with most of the time being devoted to the C and D pieces. This was mainly because these lower tones required larger areas than the others, which greatly slowed the response of the analysis. One of the difficulties the artist encountered during the design process was needing to maintain an overall balance in shape among the pieces while keeping their tones true to the intended tone. This was a major design constraint. Another difficulty was that sometimes a small modification of the shape resulted in a large change tone, necessitating high responsiveness of the analysis. Finally, the artist noted that the relationship between the shape and tone of the metallophone was still difficult to understand, even after participating in this study. Nevertheless, the artist at least learned that larger pieces tend to produce lower tones, having previously believed the opposite to be true.

6. LIMITATIONS AND FUTURE WORK

Our current system only computes the smallest nonzero eigenvalue and its eigenmode. This is sufficient for predicting the basic tone, but the computing higher order eigenvalues and eigenmodes is necessary to more accurately predict the resulting timbre. We plan to address this problem in the future. Another interesting possibility is to design a special metallophone plate that produces different tones when struck with a mallet in different positions.

The present implementation uses volumetric tetrahedral elements to model the metallophone plate. However, modeling with locking-free thick plate elements [1] might be more efficient for this purpose. The simulation for an overtone is computationally more demanding, and thick plate elements might be necessary in this case. One advantage of volumetric elements is that they are able to support truly 3D form factors, such as those in a sculpture; this is another possible direction for future research.

Interactive eigenanalysis could potentially be extended to the design of other musical instruments. We are planning to develop a system for designing a wind instrument, such as an ocarina. In this case, we would need to perform eigenanalysis on a sound wave equation. Finally, we believe that interactive shape design with concurrent FEM analysis will be useful for more general design problems, and we plan to test it in other domains.

7. ACKNOWLEDGMENTS

This project was partially supported by 2008 IPA Exploratory Software Project (Project Manager: Ikuo Takeuchi), Information-Technology Promotion Agency, Japan. We would like to thank Yuki Tsujita for designing the fish shaped metallophone, Greg Saul for his help in the construction of the metallophone, JST ERATO project members for valuable suggestions and help, and anonymous reviewers for their constructive feedback.

8. ADDITIONAL AUTHORS

Kenshi Takayama (The University of Tokyo, JST ERATO, email: `kenshi@ui.is.s.tokyo.ac.jp`)

9. REFERENCES

[1] K. Bathe and E. Dvorkin. A formulation of general shell elements- the use of mixed interpolation of tensorial components. *International Journal for Numerical Methods in Engineering*, 22:697–722, 1986.

[2] E. Bécache, A. Chaigne, G. Derveaux, and P. Joly. Numerical simulation of a guitar. *Computers and Structures*, 83(2-3):107–126, 2005.

[3] J. Bretos and C. Santamara. Finite element analysis and experimental measurements of natural eigenmodes and random responses of wooden bars used in musical instruments. *Applied Acoustics*, 56(3):141–156, 1999.

[4] H. Cox. Vibration of missiles: Matric formulation of the problem involving free and harmonically forced flexural vibrations. *Aircraft Engineering and Aerospace Technology*, 33, 1961.

[5] N. H. Fletcher and T. D. Rossing. The physics of musical instruments. 6 1998.

[6] D. L. James and D. K. Pai. Artdefo: accurate real time deformable objects. In *Proceedings of SIGGRAPH '99*, pages 65–72, 1999.

[7] D. L. James and D. K. Pai. Dyrt: dynamic response textures for real time deformation simulation with graphics hardware. *ACM Trans. Graph.*, 21(3):582–585, 2002.

[8] A. Jennings. Natural Vibrations of a Free Structure. *Aircraft Engng*, 34:81–83, 1962.

[9] J. Kindel and I. Wang. Vibrations of a piano soundboard: Modal analysis and finite element analysis. *The Journal of the Acoustical Society of America*, 81:S61, 1987.

[10] M. Masry and H. Lipson. A sketch-based interface for iterative design and analysis of 3d objects. In *Proceedings of Eurographics workshop on Sketch-Based Interfaces*, pages 109–118, Dublin, Ireland, Aug 2005.

[11] Y. Mori and T. Igarashi. Plushie: an interactive design system for plush toys. *ACM Trans. Graph.*, 26(3):45, 2007.

[12] M. Müller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler. Stable real-time deformations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 49–54, 2002.

[13] H. Nishigaki, S. Nishiwaki, T. Amago, and N. Kikuchi. First order analysis for automotive body structure design. In *ASME DETC*, 2000.

[14] Y. Saad. *Numerical Methods for Large Eigenvalue Problems (Algorithms and Architectures for Advanced Scientific Computing)*. Manchester Univ Pr, 4 1995.

[15] A. Schoofs, F. Van Asperen, P. Maas, and A. Lehr. A Carillon of Major-Third Bells. I. Computation of Bell Profiles using Structural Optimization. *Music Perception*, 4(3):255–266, 1987.

[16] J. Smith, J. Hodgins, I. Oppenheim, and A. Witkin. Creating models of truss structures with optimization. *ACM Trans. Graph.*, 21(3):295–301, 2002.

[17] O. Zienkiewicz, R. Taylor, and J. Zhu. *The finite element method: its basis and fundamentals*. Butterworth-Heinemann, 2005.