

Eigenmode Compression for Modal Sound Models

Timothy R. Langlois

Steven S. An

Kelvin K. Jin

Doug L. James

Cornell University

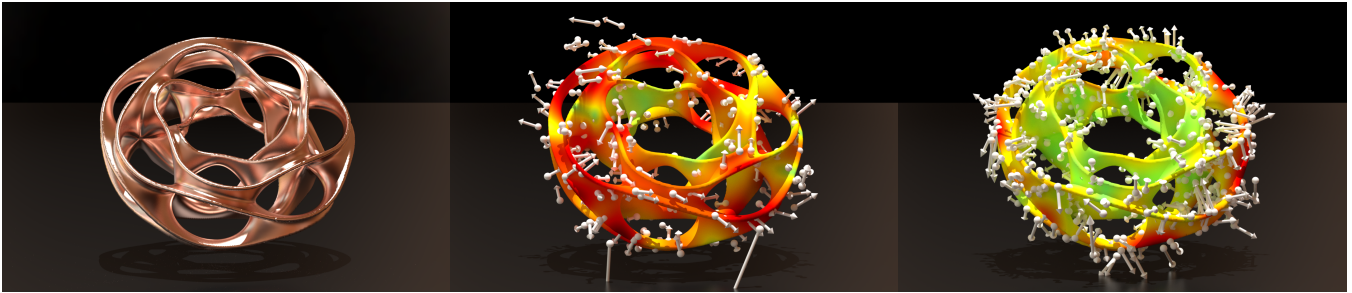


Figure 1: Eigenmode Compression: (Left) This complex Heptoroid model’s displacement eigenmode matrix has 194 audible modes, 81884 vertices, and consumes 186 MB. By approximating each eigenmode with moving least squares (MLS), and nonlinearly optimizing the control points (shown in white), we compressed the entire model down to 3.1 MB—a 60:1 compression ratio—with negligible audible difference. (Middle) mode #17 (2.67 kHz, 276 MLS control points), (Right) mode #53 (5.21 kHz, 610 MLS control points).

Abstract

We propose and evaluate a method for significantly compressing modal sound models, thereby making them far more practical for audiovisual applications. The dense eigenmode matrix, needed to compute the sound model’s response to contact forces, can consume tens to thousands of megabytes depending on mesh resolution and mode count. Our eigenmode compression pipeline is based on non-linear optimization of Moving Least Squares (MLS) approximations. Enhanced compression is achieved by exploiting symmetry both within and between eigenmodes, and by adaptively assigning per-mode error levels based on human perception of the far-field pressure amplitudes. Our method provides smooth eigenmode approximations, and efficient random access. We demonstrate that, in many cases, hundredfold compression ratios can be achieved without audible degradation of the rendered sound.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation; H.5.5 [Information Systems]: Information Interfaces and Presentation—Sound and Music Computing;

Keywords: sound synthesis, modal analysis, eigenmode, contact sounds, compression, physically based modeling, moving least squares, symmetry

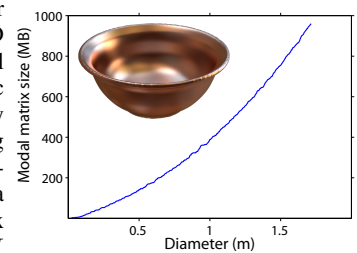
Links: DL PDF WEB

1 Introduction

3D modal sound models are one of the most effective and widely used techniques for rigid-body sound in animation and virtual environments. These methods benefit from the ability to precompute a solid object’s eigenmode matrix and eigenfrequencies, to enable rapid integration of decoupled eigenmode vibrations, which are linearly composed for sound synthesis; since we do not hear object vibrations directly, acoustic transfer models can be precomputed to efficiently estimate sound pressure at the virtual listeners’ positions. Previous decades have seen various advances to improve their speed, real-time performance, quality, measurement, contact dynamics, and acoustic radiation modeling, all to generally make them faster and better.

Unfortunately, despite their success, high-quality 3D modal sound models still suffer from a serious, basic limitation: high memory requirements. Simply storing the precomputed displacement eigenmode matrix U , a large dense N -by- M matrix for $N/3$ vertices and M modes, requires NM floats—tens to hundreds of megabytes even for small objects, dwarfing the size of other sound-model data, such as acoustic transfer. The problem is worse for larger and more complex objects, since they can have larger meshes (larger N) and more audible eigenmodes (larger M). For example, simply scaling up a large bronze bowl can quickly reach 1 GB of eigenmode storage (see inset). To make matters worse, future virtual worlds will involve not just one, but *many distinct* sound objects, each contributing to potentially massive memory overhead. Multi-gigabyte footprints are unacceptable and highly impractical for many applications. Unfortunately, without knowing where the objects are hit, or how they are temporally driven, or where the listener will be, there is no principled way to *a priori* discard eigenmodes without potentially degrading the achievable sound quality.

To address this memory bottleneck without degradation to high-quality modal sound, we propose an eigenmode compression



scheme that exploits the fact that (i) only sparse evaluation of eigenmodes is required, i.e., at contact locations, and (ii) larger relative errors are possible due to human hearing perception and varying radiation efficiency of eigenmodes. Given the eigenmode matrix, our automated compression pipeline builds an optimized Moving Least Squares (MLS) approximation for each eigenmode’s smooth vertex displacement field (see Figure 1). Nonlinear optimization of MLS sample points and weights is used to minimize fitting error for a given number of sample points. To find the bounded-error approximation with smallest sample size, we also search over the number of MLS samples used.

We can achieve further compression by exploiting human perception of mode loudness. The main result we use is loudness dependence: humans’ ability to distinguish amplitude differences decreases with quieter sounds [Fastl and Zwicker 2006], and perceived loudness is frequency dependent. By computing the expected loudness of each eigenmode (using an acoustic transfer model), we assign more error to quieter modes, thereby increasing compression without sacrificing quality. Importantly, assigning larger tolerable errors at very high frequencies (nearing the limits of human hearing) helps counteract the high geometric complexity of these rapidly oscillating modes.

To further increase compression, we exploit the fact that many man-made objects are symmetric, and their eigenmode matrices also possess symmetric structure. There are two types of symmetry we exploit (see Figure 2). First, in the case of intra-modal symmetry, we only need to store a small piece of a mode shape, and can use symmetry transforms to reconstruct the entire eigenmode from this piece. Second, some objects also exhibit inter-modal symmetries, where one mode is a rotation of another mode of the same eigenfrequency. In this case, we only need to store one mode of the pair.

Finally, we present results for numerous objects that demonstrate our method’s ability to achieve high compression ratios without noticeable degradation (see Figure 3 and §7).

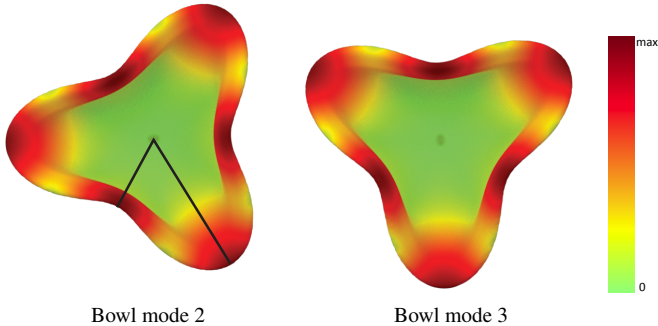


Figure 2: Eigenmode Symmetries: (Left) **Intra-mode or self symmetry** occurs when only a part of a mode (the indicated slice) must be represented, since the whole can be reproduced via symmetry transformations. (Right) **Inter-mode symmetry** occurs when a degenerate eigenmode is a transformed (here rotated) copy of another. Exploiting both symmetries here already gives 12× compression, and we further compress the slice. Figures are colored based on the magnitude of the eigenmodes, using the colormap shown. (Top down view of the Large Bowl model shown)

2 Related Work

Modal sound models are widely used in computer music and graphics, and modal vibration analyses are standard in engineering acoustics [Shabana 1990; Howe 1998]. Modal sound synthesis is a popu-

lar technique for producing plausible sounds of rigid objects [Cook 1995; van den Doel and Pai 1996; Cook 2002], and a high-quality modal sound is a long-standing goal in computer sound [Adrien 1991]. Popularized for use in rigid-body computer animation over a decade ago [van den Doel et al. 2001; O’Brien et al. 2002], many subsequent works make use of precomputed eigenmode matrices of 3D objects, e.g., using finite element analysis [O’Brien et al. 2002], for efficient runtime sound synthesis [James et al. 2006; Raghuvanshi and Lin 2006; Bonneel et al. 2008; Chadwick et al. 2009; Zheng and James 2010; Zheng and James 2011; Chadwick et al. 2012]. Unfortunately all of these approaches can suffer from large memory requirements unless small mesh resolutions, or limited accuracy sound approximations are used.

The high-cost of 3D physics-based modal sound models has been avoided in various ways in the past. Modal oscillator models can use simplified forcing models, without the need for spatial eigenmode data or finite element analyses [Cook 2002]. Van den Doel and Pai [1996] exploited modal “gain maps” based on analytical representations of eigenmodes, which avoid storage issues, but are restricted to special classes of resonators, e.g., planar membranes approximated by analytical solutions to the wave equation. Modal models and gain maps can also be estimated empirically from measured sound samples [van den Doel et al. 2001], and automated using robotic measurement [Pai et al. 2001] with adaptive sampling [Richmond 2000]. Arbitrarily high compression can be obtained by using highly simplified response models, e.g., in [van den Doel et al. 2001] the modal “gain map” for a wok is interpolated from just five measured points. In contrast, we attempt to enable high-resolution eigenmode representations for detailed sound models. Many techniques have been introduced to accelerate the

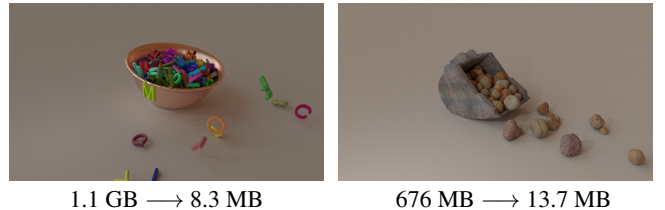


Figure 3: Compression Benefits: *Eigenmode memory requirements for large simulations are drastically reduced without audible differences. (Left) 191 Letters are dropped into a large bowl. (Right) 125 Rocks fall out of a backhoe scoop.*

modal sound synthesis by simplifying sound computations. Van den Doel et al. [2004] use knowledge of human perception and auditory masking for runtime culling of modes which are either not sufficiently excited or audible. This accelerates runtime synthesis, but does not address the runtime memory costs of eigenmode matrices, since all modes must still be available in case they are not culled. Raghuvanshi and Lin [2006] reduce eigenmode memory and synthesis costs by aggressively combining modes whose frequencies differ by less than the human frequency discrimination limit, achieving a space reduction factor of about 15. However, this approach can introduce large errors depending on where the object is hit, how it is driven, e.g., at a mode’s resonance, or where the listener is positioned (when acoustic transfer is used). In contrast, we do not discard any modes (although runtime culling could still be used), and we achieve a much higher compression ratio, without introducing audible errors or subtle effects, e.g., like “beating” of nearby modes which can be perceptually important, such as for bells. Methods for auditory culling and spatial level-of-detail can reduce the number of sound sources at runtime [Tsingos et al. 2004], or aggressively cull modes [Grelaud et al. 2009], or delay or reduce mode simulation costs [Bonneel et al. 2008], but do not provide sound model compression.

Proxy sound models based on ellipsoidal soundbanks [Zheng and James 2010] can avoid the need to build and store object-specific modal models, however the soundbank can itself have an enormous memory footprint, which our method can compress dramatically. Instancing methods are often used to support sound synthesis with many objects, e.g., [Bonneel et al. 2008], simplifying precomputation, but not avoiding the storage of at least one sound model.

High-quality sound pressure can require acoustic transfer functions, which add additional storage requirements of, e.g., multipole expansions [James et al. 2006; Zheng and James 2010]. However, these costs are an order of magnitude less than for the eigenmode matrix in all of our experiments.

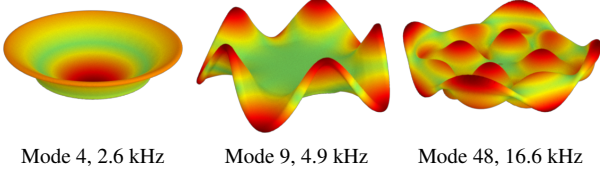


Figure 4: Linear Vibration Modes: Three displacement eigenmodes \mathbf{u}^j of the Dinner Plate model. We approximate the eigenmode values on the surface, which are needed to compute the system’s response to external contact forces \mathbf{f} via dot products, $(\mathbf{u}^j \bullet \mathbf{f})$.

We use MLS approximation, which has seen various applications in computer graphics for surface reconstruction [Fleishman et al. 2005; Sorkine and Cohen-Or 2004], defining implicit surfaces [Alexa et al. 2001; Shen et al. 2004], image deformation [Schaefer et al. 2006], and interpolating scalar fields [Jin et al. 2009]. We use MLS approximations and heavily optimize their control parameters, and exploit symmetries, to compactly represent M eigenmode vector fields per object.

Eigenmode compression is related to compressing collections of mesh displacement fields. Many mesh animation compression schemes use principal components analysis (PCA) or other data reduction techniques to decorrelate the displacement fields [Alexa and Müller 2000], however these do not address how to compress the mode basis itself. Seo et al. [2011] considered the problem of compressing blend shapes for animation, however their method exploits the sparsity of deformation fields which does not apply to global eigenmodes. Second-generation wavelets [Schröder and Sweldens 1995; Kolarov and Lynch 1997] offer another approach to compressing surface displacement field operators, such as for “spiky” Green’s function displacement fields to support fast wavelet summation [James and Pai 2003], and have been used to compress parametrically coherent mesh animations [Guskov and Khodakovsky 2004]. However, they are less ideal for compressing pseudo-band-limited eigenmodes with sparse random access, especially when mode-specific mesh patches are needed to exploit symmetry.

There exists a large body of work in computer graphics on analyzing, detecting, and processing symmetry [Mitra et al. 2012]. One of the major applications of symmetry research is indeed compression [Simari et al. 2006], and many researchers have proposed different methods for analyzing and representing symmetric objects [Lipman et al. 2010; Mitra et al. 2006]. Our approach builds upon the work of Martinet et al. [2006], which we extend to detect geometric and modal symmetries simultaneously.

3 Background

Before explaining our compression approach, we briefly summarize the 3D modal sound model employed (which is identical

to [Zheng and James 2010]), and discuss its memory requirements. We start with the equation of motion for a 3D linear elastodynamic model [Bathe 1996], $\mathbf{M} \ddot{\mathbf{u}}(t) + \mathbf{C} \dot{\mathbf{u}}(t) + \mathbf{K} \mathbf{u}(t) = \mathbf{f}(t)$ where $\mathbf{u} \in \mathbb{R}^N$ is the vector of $N/3$ nodal displacements, and \mathbf{M} , $\mathbf{C} = \alpha \mathbf{M} + \beta \mathbf{K}$, $\mathbf{K} \in \mathbb{R}^{N \times N}$ are the mass, Rayleigh damping, and stiffness matrices, respectively; $\mathbf{f} \in \mathbb{R}^N$ represents any external nodal forces, such as contact forces, which act to excite vibrations. Modal analysis of linear finite element models is standard in graphics and engineering (see [Shabana 1990; Bathe 1996]), and decouples the N equations of motion into independent simple harmonic oscillators. It amounts to first solving the generalized eigenvalue problems, $\mathbf{K} \mathbf{u}^j = \omega_j^2 \mathbf{M} \mathbf{u}^j$, for the eigenvectors \mathbf{u}^j and nonzero eigenvalues ω_j^2 , with $j = 1 \dots M$. Here \mathbf{u}^j represents the j^{th} normal vibration mode’s nodal displacement field, and ω_j is its natural frequency of vibration; M is the number of eigenmodes with frequencies within the audible frequency range of interest (20 Hz – 20 kHz). Some vibration modes are illustrated in Figure 4. The eigenmode matrix of interest is then

$$\mathbf{U} = \begin{bmatrix} | & | & \dots & | \\ \mathbf{u}^1 & \mathbf{u}^2 & & \mathbf{u}^M \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{N \times M}$$

and can be normalized such that $\mathbf{U}^T \mathbf{M} \mathbf{U} = \mathbf{I}$. Letting $\mathbf{u}(t) = \mathbf{U} \mathbf{q}(t)$, where $\mathbf{q} \in \mathbb{R}^M$ are modal amplitudes, the equation of motion for the j^{th} mode becomes

$$\ddot{q}_j(t) + (\alpha + \beta \omega_j^2) \dot{q}_j(t) + \omega_j^2 q_j(t) = \mathbf{u}^j \bullet \mathbf{f}(t) \quad (1)$$

and can be solved rapidly during sound synthesis [van den Doel et al. 2001]. Note that the eigenmodes \mathbf{U} are needed at runtime to project the contact forces \mathbf{f} into the modal subspace, i.e., via $\mathbf{U}^T \mathbf{f}$, and require NM floats of storage. For contact sound synthesis, one immediate source of compression is that only surface \mathbf{U} values are needed, and so, for convenience (but a slight abuse of notation), we will hereafter assume that N refers to surface degrees of freedom.

The other significant storage requirement for high-quality modal sound is an acoustic transfer model to estimate sound pressure. Without loss of generality, the sound at position \mathbf{x} and time t can be approximated as $\sum_{j=1}^M |p_j(\mathbf{x})| q_j(t)$, where $p_j(\mathbf{x})$ is the acoustic pressure due to a unit-amplitude oscillation of mode j . Here it is approximated by the single-point multipole expansion

$$p_j(\mathbf{x}) = \sum_{\ell=0}^{\bar{n}_j} \sum_{m=-\ell}^{\ell} h_{\ell}^{(2)}(k_j r) Y_{\ell}^m(\theta, \phi) c_{\ell,m}^j, \quad (2)$$

where $k_j = \omega_j/c$ is the wavenumber, $h_{\ell}^{(2)}$ is a spherical hankel function of the second kind, and $c_{\ell,m}^j \in \mathbb{C}$ are the $(\bar{n}_j + 1)^2$ multipole expansion coefficients that must be precomputed and stored. For sufficient accuracy, the expansion order is set to $\bar{n} = \max(\frac{1}{4} k_j L, 4)$, where L is the diameter of the object [Zheng and James 2010; Liu 2009]. We note that, for all models we have analyzed, storage requirements for the uncompressed mesh-resolution-dependent eigenmode matrix, \mathbf{U} , are at least an order-of-magnitude larger than for the multipole coefficients, \mathbf{c} .

4 Eigenmode Approximation

We now describe how we fit each eigenmode \mathbf{u}^j with an optimized Moving Least Squares (MLS) approximation, $\tilde{\mathbf{u}}^j$, such that its approximation error ε is below a mode-specific error bound, $\varepsilon(\tilde{\mathbf{u}}^j) \leq \varepsilon_{\text{goal}}^j$; we use a relative ℓ_2 error norm, $\varepsilon(\tilde{\mathbf{u}}) = \|\mathbf{u} - \tilde{\mathbf{u}}\|_2 / \|\mathbf{u}\|_2$. The MLS approximation has a number of desirable features for eigenmode approximation: it provides a smooth approximation suitable

for approximating both low- and high-frequency eigenmodes of varying complexity; we can optimize the number of MLS samples and their parameters (positions and weights) to minimize fitting error while achieving high compression; it is a meshless approximation that accepts flexible input data, and simplifies implementation; it provides random-access eigenmode evaluation at $O(1)$ cost at runtime; it can be adapted to exploit symmetry; and it efficiently interpolates samples across thin volumes. We note that this is an asymmetric compression scheme: high compression is achieved using a computationally expensive MLS-fitting preprocess, but subsequent random-access decompression is fast. In later sections, we also describe how we gain further compression by: (1) when appropriate, we exploit intra- and inter-mode symmetry, and (2) we adjust each mode’s error level ϵ_{goal}^j based how well it radiates, as well as human frequency and amplitude perception.

4.1 Moving Least Squares

We seek to find a set of 3D control points $\mathbf{p} = (\mathbf{p}_i)_{i=1}^n$ with associated displacement values $\mathbf{w} = (\mathbf{w}_i)_{i=1}^n$ from which an MLS approximation can reconstruct the original mode accurately. Once we have them, we can evaluate a scalar component of the mode at \mathbf{x} , by first constructing an m -degree polynomial, $f(\mathbf{p} - \mathbf{x}) = \mathbf{c}^T \mathbf{b}(\mathbf{p} - \mathbf{x}) \in \Pi_m^3$, with $d = \frac{(3+m)!}{3!m!}$ coefficients $\mathbf{c} \in \mathbb{R}^d$, where $\mathbf{b}(\mathbf{p} - \mathbf{x}) \in \mathbb{R}^d$ is a vector of monomial basis functions. Given the control parameters \mathbf{p} and \mathbf{w} , the coefficients \mathbf{c} are computed by minimizing the MLS error,

$$\mathbf{c}^* = \arg \min_{\mathbf{c}} \sum_{i=1}^n [w_i - f(\mathbf{p}_i - \mathbf{x})]^2 \theta(\mathbf{p}_i - \mathbf{x}). \quad (3)$$

Each xyz component of \mathbf{f} is computed separately, by replacing w_i with $\mathbf{w}_{i,x}$, $\mathbf{w}_{i,y}$ or $\mathbf{w}_{i,z}$ in (3); we use a QR factorization to solve the least-squares problem, which involves solving with three (xyz) right-hand sides. Once \mathbf{f} is fitted, the mode approximation at vertex \mathbf{x} is simply $\mathbf{f}(\mathbf{0})$. The weighting function θ controls the influence of each control point; we use the adaptive $\theta(\mathbf{v}) = \exp(-\|\mathbf{v}\|^2/h^2)$ defined in [Pauly et al. 2002], where $h = r/3$, with r the radius of the enclosing sphere of the k nearest neighbors of \mathbf{x} . This weight function allows the approximation to adapt to varying control point densities, and also improves performance since it is essentially zero for control points with $\|\mathbf{p}_j - \mathbf{x}\| > r$.

Storage and evaluation speed: Unless otherwise stated, we use cubic approximation, $m = 3$, and $k = 28$ nearest neighbors when choosing and optimizing control points. Storage of the MLS representation for an eigenmode with n_j control points requires $6n_j$ floats (3 for each \mathbf{p}_i and \mathbf{w}_i). The time required to approximate a mode value using the MLS representation is very fast, and still enables real time performance. In our implementation, evaluation takes about $40\mu s$, and involves finding the k nearest control points (using a kd-tree), and solving the 3 linear systems of size $k \times d$. Detailed timing statistics are given in §7.

Control point initialization using adaptive MLS: The key step in MLS compression is to choose where to place our n control points. We initialize their placement using the greedy selection method of [Sorkine and Cohen-Or 2004]. Since we need at least d control points to ensure the linear system is not under-determined, we first randomly pick d mesh vertices as control points, setting their weights \mathbf{w} to vertex displacements \mathbf{u} . We then iteratively improve the approximation by adding the vertex with the highest eigenmode error as the next control point, until the error reaches ϵ_{goal} or the desired number of control points n is attained. This approach could quickly reach ϵ_{goal} with a modest number of sample points ($\ll N$) for all examples we tested.

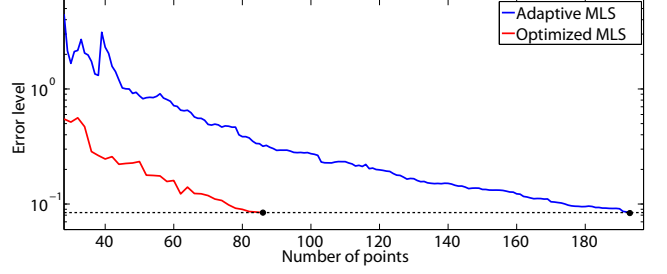


Figure 5: MLS error convergence versus n : Adaptive MLS provides fair compression at the target error, $\epsilon_{\text{goal}} = 0.084$, but our optimized MLS fit requires even fewer control points (lower n).

4.2 Control Point Optimization

By further optimizing the n control points and weights, we can significantly improve compression over adaptive MLS (see Figure 5). Since the MLS approximation, $\tilde{\mathbf{u}}$, is a function of the controls \mathbf{p} , $\mathbf{w} \in \mathbb{R}^{3n}$, we optimize their values using the nonlinear least-squares optimization of eigenmode error at vertices \mathcal{V} ,

$$\min_{\mathbf{p}, \mathbf{w}} \|\tilde{\mathbf{u}}(\mathbf{p}, \mathbf{w}) - \mathbf{u}\|_2^2 = \min_{\mathbf{p}, \mathbf{w}} \sum_{i \in \mathcal{V}} \|\tilde{\mathbf{u}}_i - \mathbf{u}_i\|_2^2. \quad (4)$$

We perform this nonlinear least-squares optimization using the Levenberg-Marquardt (LM) algorithm; we use the *Ceres Solver* implementation [Agarwal et al.] which uses automatic differentiation to compute the Jacobian $\mathbf{J} = \nabla \tilde{\mathbf{u}}(\mathbf{p}, \mathbf{w})$.

Multi-level optimization: Since the compressed models have very few control parameters, $6n$, the bottleneck of each LM iteration is not its linear solve, but rather the Jacobian computation which has cost dependent on the number of vertices, $|\mathcal{V}| \gg 6n$. To reduce this cost, we use a 3-stage multi-level approach: we first run LM to convergence using only $1/16$ of the vertices, $\mathcal{V}_{1/16}$, then improve this initial guess by running LM to convergence using $1/4$ of the vertices, $\mathcal{V}_{1/4}$, and finally we use all of the vertices, \mathcal{V} , as in (4). This dramatically reduces the number of expensive all-vertex Jacobian evaluations, since there are much fewer expensive LM iterations due to the improved starting guess. We use non-nested vertex sets, $\mathcal{V}_{1/16}$ and $\mathcal{V}_{1/4}$, where each is half randomly sampled, and half importance sampled from $\|\mathbf{u}_i\|$ to avoid missing localized modes.

Minimizing n : Highest compression for mode j means finding the fewest control points, n_j , with bounded error, $\epsilon(\tilde{\mathbf{u}}^j) \leq \epsilon_{\text{goal}}^j$. Unfortunately the optimization process only adjusts the control points, (\mathbf{p}, \mathbf{w}) , but cannot add or remove them. We use a simple binary search to determine the smallest n_j with $\epsilon(\tilde{\mathbf{u}}^j) \leq \epsilon_{\text{goal}}^j$, by repeatedly bisecting the minimum and maximum number of controls and optimizing them—the minimum n_j is initialized to $l_{\min} = d$, and the maximum l_{\max} is taken as the number of greedily chosen adaptive MLS points. To save time on large examples, we “early exit” the expensive \mathcal{V} -level LM optimization when the error falls below ϵ_{goal}^j .

5 Exploiting Symmetry

Many objects, especially manufactured ones, are symmetric. If their modes also exhibit symmetries, we can increase the amount of compression by only storing parts of some modes, and using symmetry to reconstruct the rest. We first analyze the geometry to detect any cylindrical, n -way rotational, and mirror symmetries using the method of [Martinet et al. 2006]. If approximate eigenmode symmetries exist, then we proceed to exploit them as follows.

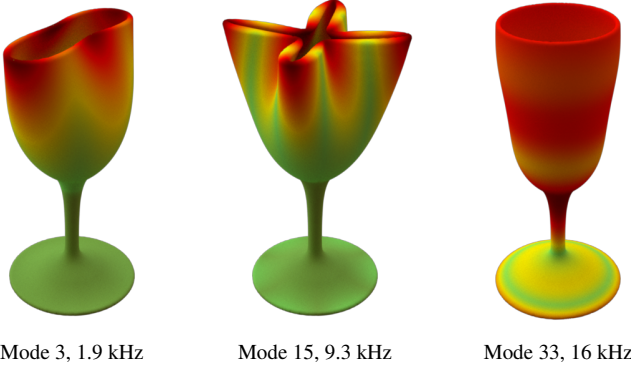


Figure 6: Intra-mode symmetry examples: (Left) mirror symmetry; (Middle) 4-way rotational symmetry, plus several mirror symmetries; (Right) cylindrical symmetry.

5.1 Intra-mode symmetry

The first symmetry we exploit is intra-mode or self symmetry (see Figures 2 and 6). We slightly modify the geometric-symmetry method of [Martinet et al. 2006] to detect object and eigenmode symmetries simultaneously. Instead of a purely geometric generalized moment function, we compute the *generalized geometry-eigenmode moment function of order $2p$* ,

$$\mathcal{M}^{2p}(\hat{\mathbf{v}}) = \int_{\mathbf{s} \in S} \|\mathbf{s} \times \hat{\mathbf{v}}\|^{2p} \|\mathbf{u}(\mathbf{s})\|^{2p} d\mathbf{s}, \quad (5)$$

where \mathbf{s} is a vector from the surface’s center of mass to a point on the surface, S . It follows that their real-valued spherical harmonic representation is given by

$$\mathcal{M}^{2p}(\hat{\mathbf{v}}) = \sum_{l=0}^p \sum_{m=-2l}^{2l} C_{2l,m}^{2p} Y_{2l}^m(\hat{\mathbf{v}}), \quad (6)$$

$$C_{2l,m}^{2p} = S_p^l \int_{\mathbf{s} \in S} \|\mathbf{s}\|^{2p} \|\mathbf{u}(\mathbf{s})\|^{2p} D_{2l}^{0,m}(R_{\mathbf{s}}) d\mathbf{s}, \quad (7)$$

where formulae for S_p^l and $D_{2l}^{0,m}(R_{\mathbf{s}})$ are given in [Martinet et al. 2006]. By searching among roots of $\nabla \mathcal{M}^{2p}(\hat{\mathbf{v}}) = 0$, we find candidate symmetry axes, and then classify the symmetries of the eigenmode magnitudes as either cylindrical, n-way rotation, or mirror symmetries as described in [Martinet et al. 2006]. In our implementation, we use order $2p = 8$ moment functions.

Patch Extraction: After detecting symmetries using our generalized moment function, we find a patch of the object to approximate with MLS for storage. First, the symmetries are sorted in order of their expected compression contribution: cylindrical symmetries, n-way rotational symmetries, and finally mirror symmetries. Then the symmetries are checked in order. If a symmetry works for the current set of vertices, only a patch of the vertices are MLS-compressed and stored, and the process continues with this patch. For cylindrical symmetries, we store a 5° slice of the object’s surface. For mirror symmetries we store one side of the mirror. For n-way rotational symmetries, we only need to store a $\frac{360^\circ}{n}$ slice. However, often there are orthogonal mirror symmetries to the n-way rotational symmetries which can provide additional savings. Therefore, when choosing which slice of an n-way symmetry to store, we search the remaining symmetries for any orthogonal mirrors, and position our slice to take advantage of one of them, if found (see Figure 7).

Displacement transformation: After finding a direction-invariant symmetry, $\|\mathbf{u}(\mathbf{x})\| \approx \|\mathbf{u}(\mathbf{R}\mathbf{x})\|$ for some \mathbf{R} symmetry transformation, it still remains to determine any orthogonal transformation

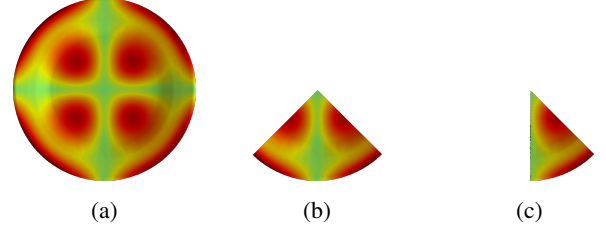


Figure 7: Intra-mode symmetry example: Starting with a full mode (a), we detect symmetries and only save a small patch of the mode. In (b), a 4-way rotational symmetry is used, and in (c), a mirror symmetry is exploited.

needed to match the eigenmode patch’s vector displacements. We use a least-squares solve on vertex data to estimate any (orthogonal) displacement transformation, \mathbf{T} (with $\|\mathbf{T}\|_2 \approx 1$) such that $\mathbf{u}(\mathbf{x}) = \mathbf{T} \mathbf{u}(\mathbf{R}\mathbf{x})$.

Symmetry tolerances: Given the approximate nature of discrete eigenmode symmetry (due to meshing, MLS interpolation, numerical eigenanalysis, etc.) we use a tolerance when confirming eigenmode symmetry; in our results, we use 0.02.

5.2 Inter-mode symmetry

Beyond symmetry within a single mode, an interesting characteristic of cylindrically and n-way rotationally symmetric objects is that they can have degenerate eigenmodes, i.e., modes with near-equal eigen-frequencies, which form rotationally congruent pairs (see Figures 2 and 8). If we can detect a congruent pair (j, j') , we only need to store one of them along with the relative rotation which maps one to the other. We detect these pairs by summarizing the angular structure of the modes in a low-dimensional Fourier basis to find a candidate rotation, and then perform a rigorous verification of the candidate. Furthermore, we observe that congruent pairs are usually close to each other in frequency, so instead of doing this for all pairs (j, j') , we only do it for pairs such that $j' = j + 1$ (assuming modes are numbered in order of increasing frequency).

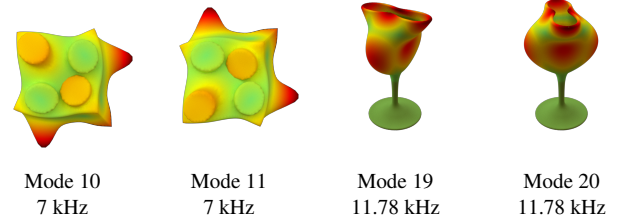


Figure 8: Inter-mode symmetry: Pairs of rotationally congruent eigenmodes (shown here for Lego and Wine Glass models) just need to store one of the modes and a relative rotation.

For a given pair of modes (j, j') , we first focus on the problem of finding a best rotation angle $\phi_{j,j'}$ about a known symmetry axis. For mode j , we compute Fourier-like moments,

$$a_m^j = \int_S \|\mathbf{u}^j(\mathbf{x})\| e^{im\phi(\mathbf{x})} dS_{\mathbf{x}}, \quad m = -\bar{m} \dots \bar{m}$$

that describe the mode’s amplitude variation about the rotation axis,

$$A^j(\phi) = \sum_{m=-\bar{m}}^{\bar{m}} a_m^j e^{-im\phi}.$$

In our implementation we use $\bar{m} = 30$. Much like the generalized moment function \mathcal{M} , this representation allows us to efficiently search for candidate rotation angles that map \mathbf{u}^j to $\mathbf{u}^{j'}$. For two congruent modes (j, j') that have $A^j(\phi) = A^{j'}(\phi + \phi_{j,j'})$, it follows that $a_m^{j'} = a_m^j e^{im\phi_{j,j'}}$. Therefore, we can estimate the rotation angle by considering the near-zero minima of

$$E(\phi) = \sum_{m=-\bar{m}}^{\bar{m}} \left| a_m^j - a_m^{j'} e^{-im\phi} \right|^2.$$

Given the global minima of $E(\phi)$, we verify that the modes are congruent by evaluating the relative global error,

$$\int_S \left\| \mathbf{u}^j(\mathbf{x}) - \mathbf{R}_{\phi_{j,j'}}^T \mathbf{u}^{j'}(\mathbf{R}_{\phi_{j,j'}} \mathbf{x}) \right\| dS_{\mathbf{x}} / \int_S \left\| \mathbf{u}^j(\mathbf{x}) \right\| dS_{\mathbf{x}}, \quad (8)$$

where the look-up at the rotated position, $\mathbf{u}^{j'}(\mathbf{R}_{\phi_{j,j'}} \mathbf{x})$, is approximated using MLS on the mode's vertex data (using $m=0, k=3$). In this way, we consider all geometric cylindrical and n-way rotation symmetry axes, estimate the best rotation angle for each, and take the symmetry with the lowest feasible global error.

Symmetric Acoustic Transfer: One additional benefit of detecting congruent eigenmodes, is that their acoustic transfer functions are also congruent fields. Therefore it is only necessary to compute acoustic transfer multipole coefficients for one of the modes, and the other one is obtained by rotating the multipole expansion coefficients using standard spherical harmonic techniques.

5.3 Runtime Evaluation

Given an external force $\mathbf{f}_i \in \mathbb{R}^3$ acting on node i at vertex position \mathbf{x} , we evaluate each eigenmode's value $\tilde{\mathbf{u}}^j(\mathbf{x})$, and evaluate the dot product $\tilde{\mathbf{u}}^j(\mathbf{x}) \bullet \mathbf{f}_i$ required to force each mode j . To evaluate a $\tilde{\mathbf{u}}(\mathbf{x})$ value (dropping the j superscript) for a mode with symmetry, we transform \mathbf{x} to its symmetric image location, \mathbf{x}' , perform the MLS lookup $\tilde{\mathbf{u}}'(\mathbf{x}')$, then apply T to get the final $\tilde{\mathbf{u}}$.

Mode Caching: Simulations will often apply contact forces to the same vertex repeatedly, e.g., during sliding or resting contact. Consequently, repeated reconstruction of identical mode values can ensue, which is wasteful. We therefore allow a small cache of recently reconstructed vertex mode values, and reuse them if they are repeatedly queried. For simulations with persistent and/or resting contacts we often observe a cache hit rate $> 99\%$, whereas simulations dominated by rolling have very poor cache hit rates.

6 Perceptually Based Error Allocation

Choosing good MLS fitting error tolerances, ϵ_{goal}^j , are critical, as they control the balance between compression and quality. While one could simply set all ϵ_{goal}^j to a conservatively small value which guarantees no audible difference will be heard (e.g., 2.3% would ensure an imperceptible 0.2 dB difference), much higher compression can be obtained (especially at high frequencies) without sacrificing quality by setting per-mode errors based on human perception of each mode's sound. We incorporate three factors: (1) vastly different eigenmode loudness due to radiation efficiency, (2) frequency-dependent hearing sensitivity, and (3) a perceptual model of just noticeable differences (JND) in sound amplitudes.

We reason about the relative loudness of mode j by considering its far-field radiated power for a random excitation; we estimate a relative average pressure as proportional to $\|\mathbf{u}^j\| \|\mathbf{c}^j\|_F / (\omega^j)^2$ (see Appendix A for a derivation). Since the average far-field pressure is only proportional to this value, we shift the dB pressure values by p_{dB}^{shift} so that the maximum modal pressure occurs at 60dB. After calculating the average pressure for a mode, we weight it with the ISO-226 equal-loudness curve (see Figure 9), which normalizes sound pressure levels at different frequencies to more accurately represent human hearing. Since this ISO standard is only defined up to 12.5 kHz, we extended it using data from [Ashihara 2007]: we fit a cubic spline to interpolate the ISO data, and added three additional (frequency, SPL) points: (16 kHz, 41.8 dB), (18 kHz, 64 dB), (20 kHz, 89.9 dB).

The final representative modal pressure amplitude combining these factors is

$$p_{dB}^j = 20 \log \left(\frac{\|\mathbf{u}^j\| \|\mathbf{c}^j\|_F}{(\omega^j)^2} \right) - \text{ISO-226}(\omega^j) + p_{dB}^{shift}. \quad (9)$$

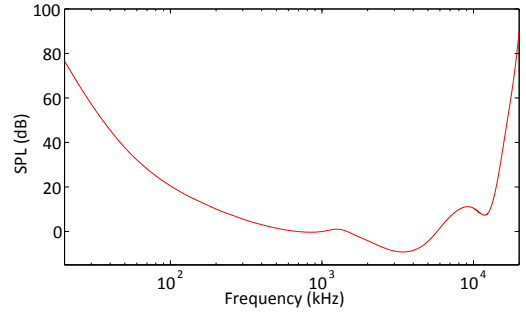


Figure 9: Extended ISO-226 Frequency Weighting

We then use this weighted average pressure to set a compression error level so that the rendered differences are expected to be near the JND. Studies in psychoacoustics have determined that humans' ability to distinguish amplitude differences in sounds varies with the sound's loudness, with larger differences tolerated at lower amplitudes [Fastl and Zwicker 2006] (see Figure 10). Given an acceptable pressure amplitude difference Δp (in dB) for a mode, the corresponding error is $\epsilon_{\text{goal}} = 10^{\Delta p/20} - 1$. Based on JND amplitude differences Δp for a single tone at different amplitudes p_{dB} ([Fastl and Zwicker 2006], p.180), we model the acceptable error level using the power-law approximation, $\epsilon = 1.5 (p_{dB})^{-0.9}$ (shown in Figure 10). The effect of perceptual weighting on MLS compression is shown in Figure 11. Graphs of relative pressure estimates (9) and the resulting allocated per-mode error levels ϵ_{goal}^j are shown in Figure 12 for several objects.

Figure 10: Amplitude Just Noticeable Differences are given for a 1 kHz tone at different amplitudes, along with the power-law approximation we use.

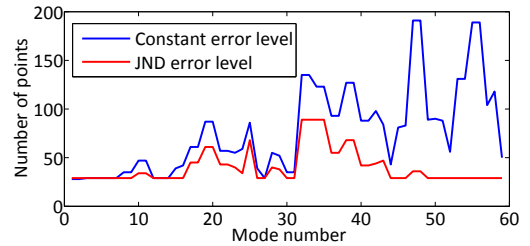
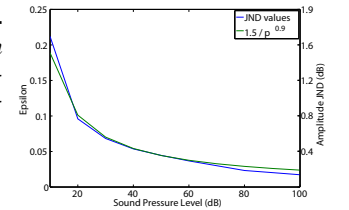


Figure 11: Benefits of perceptual weighting: Perceptually based error allocation allows fewer points to be used for quieter modes, compared to setting a constant $\epsilon_{\text{goal}} = 0.037$ (0.32 dB)—the smallest error our model will assign. (Results are for the Dinner Plate.)

7 Results

We present compression statistics in Table 1 for numerous models. The most complex nonsymmetric and symmetric models are the Heptoroid (see Figure 1) and the Large Bowl, respectively, and

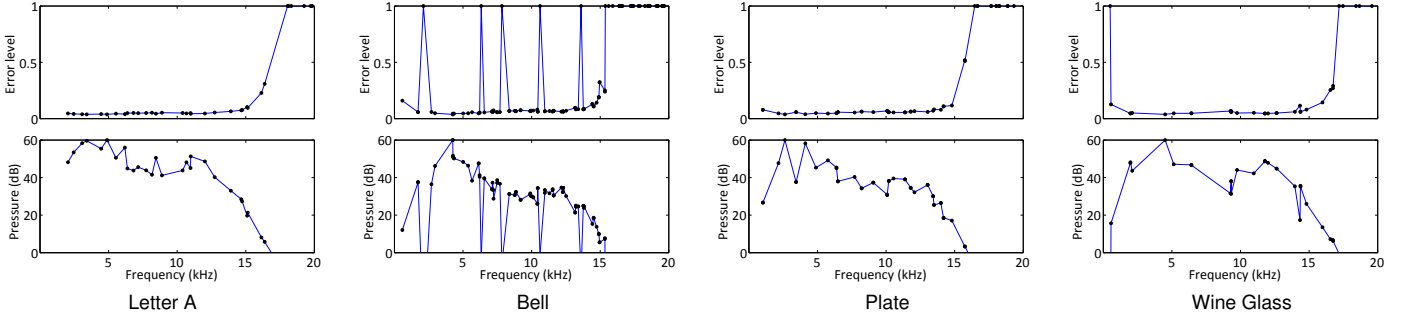


Figure 12: Perceptually based errors (Top) demonstrate that larger compression errors $\varepsilon_{\text{goal}}^j$ are used for modes which are either quieter or occur at frequencies of less perceptual importance; (Bottom) representative far-field modal pressures (normalized to 60dB) used in our error model, with ISO-226 weighting applied. Note that quieter modes are allocated larger MLS error goals, whereas the loudest modes are allocated the smallest errors. We clamp the maximum error to 1.

| Model | Geometry | | | MLS | | | | Transfer | | Symmetry | |
|----------------|----------|------|--------|------------------------|--------------------|---------------------|-------------|---------------------|-------------------------|-----------------|----------------------|
| | vertices | M | D (cm) | Uncompressed size (kB) | Adaptive size (kB) | Optimized size (kB) | Optimized R | Multipole size (kB) | Sym Multipole size (kB) | Symmetry-only R | # of congruent pairs |
| Large Bowl | 70276 | 1194 | 200 | 983311 | 14835 | 7366 | 133 : 1 | 215715 | 163762 | 4.2 : 1 | 286 |
| Backhoe Bucket | 99253 | 494 | 96 | 574582 | 19734 | 9989 | 57 : 1 | 16353 | 16353 | 1.1 : 1 | 0 |
| Heptoroid | 81884 | 194 | 30 | 186158 | 6497 | 3064 | 60 : 1 | 470 | 470 | 1.0 : 1 | 0 |
| Bell | 41974 | 134 | 30 | 65912 | 408 | 200 | 329 : 1 | 2026 | 1668 | 1.6 : 1 | 23 |
| Lego (huge) | 7706 | 121 | 10 | 10927 | 319 | 183 | 59 : 1 | 111 | 88 | 3.2 : 1 | 26 |
| Dinner Plate | 50214 | 59 | 23 | 34718 | 74 | 39 | 897 : 1 | 182 | 135 | 4.6 : 1 | 17 |
| Wine Glass | 51434 | 46 | 19 | 27726 | 41 | 26 | 1061 : 1 | 108 | 69 | 5.7 : 1 | 16 |
| Number 8 | 7843 | 37 | 30 | 3401 | 90 | 38 | 89 : 1 | 199 | 199 | 1.6 : 1 | 0 |
| Letter A | 7534 | 34 | 30 | 3002 | 88 | 42 | 71 : 1 | 167 | 167 | 2.0 : 1 | 0 |
| Ellipsoid DB | 6916 | 3037 | 40 | 251488 | 3602 | 2515 | 100 : 1 | 30463 | 30463 | 5.1 : 1 | 0 |
| Rocks | 3057 | 2727 | 26 | 101338 | 8444 | 3720 | 27 : 1 | 5606 | 5606 | 1 : 1 | 0 |

Table 1: Compression Statistics including compression ratios (R), transfer sizes, symmetry information, and the diameter (D) are given for various models. For transfer sizes, we report the total size and the size if only one of each congruent pair is stored. For the ellipsoid database of 71 ellipsoids, and the collection of 38 rock models, we report the total sizes and compression ratios, as well as the average number of vertices per model.

both cases observe excellent compression. For brevity, selected representative results are provided for collections of processed objects: plastic letters of the alphabet and plastic numerals. Please see the accompanying video for demonstrations, and comparisons of sounds generated using the uncompressed (before) and compressed (after) models. We provide comparisons for (i) before/after point-like impact sounds, and (ii) before/after sounds produced by various animations involving rigid-body contact dynamics. In both cases, compression introduces very negligible (if any) audible sound differences, as desired.

For all models, the surface mesh we use is the surface of the tetrahedral mesh used for the eigenmode analysis. The tetrahedral mesh was chosen to conform to the original geometry well. The number of surface vertices directly determines the uncompressed size of the eigenmode matrix, and while coarser meshes could be used, they will give coarser approximations of the eigenmodes.

Implementation details: Modal analysis and acoustic transfer computations are done using the implementation described in [Zheng and James 2010]. Rigid-body dynamics are simulated using a solver based on [Guendelman et al. 2003]. All timing statistics are run on a machine with four Intel Xeon 7560 processors at 2.27GHz.

Ellipsoidal proxy soundbank: We compressed ellipsoid sound models from the ellipsoidal soundbank of [Zheng and James 2010]. Each ellipsoid was processed at the maximum size it could be used at (we chose to scale each ellipsoid so the maximum bounding box dimension was 40cm). For each ellipsoid mode with index m , the final compression level was calculated as $\varepsilon_{\text{goal}}^m = \min_{j \geq m} \varepsilon_{\text{goal}}^j$. This thresholding ensures that whenever the ellipsoid proxies are scaled down, the error level for each mode never increases. Despite these

conservative error levels, by exploiting symmetry, we achieved a high compression ratio ($100\times$) without introducing perceptible error (see Table 1). Surprisingly, at these compression rates, all compressed eigenmodes of the soundbank consume nearly as little memory as a single uncompressed model, greatly improving practical uses.

Symmetry: Applying intra- and inter-mode symmetry-based compression alone we observed up to $5.7\times$ compression (see Table 1). Models with a large number of inter-mode congruent pairs, such as cylindrical models, e.g., 286/1194 modes (24%) for Large Bowl, were a particularly effective source of compression.

Comparison (Varying MLS polynomial degree, m): While we tend to use cubic $m = 3$ degree MLS approximations due to their good compression, lower degrees are also possible when faster reconstruction is desired. A comparison of compression and reconstruction costs for varying degree, m , is shown in Table 2.

| | m | Opt. size (kB) | Adaptive size (kB) | Opt. time (m) | Adaptive time (s) | Avg. vertex MLS time (μ s) |
|-------------|-----|----------------|--------------------|---------------|-------------------|---------------------------------|
| Wine Glass | 1 | 37 | 71 | 40 | 56 | 2.8 |
| | 2 | 27 | 48 | 23 | 31 | 7.8 |
| | 3 | 26 | 43 | 45 | 91 | 33 |
| Lego (huge) | 1 | 416 | 493 | 153 | 139 | 3.7 |
| | 2 | 226 | 383 | 308 | 87 | 8.8 |
| | 3 | 183 | 327 | 388 | 96 | 34 |

Table 2: Dependence of compression and reconstruction costs on polynomial degree, m .

8 Conclusion

We have presented a method for significantly compressing eigenmode matrices used in 3D modal sound synthesis using moving least squares approximations, and by exploiting symmetry. A perceptually based error model is proposed which allows the use of larger errors for modes at perceptually quieter frequencies. Overall, the method is able to achieve high compression rates, at the cost of a small runtime evaluation cost at sparse contact points.

Limitations and Future Work: While compressed modes result in very small differences in the rendered sounds, uncompressed modes should be used for the best quality. Building complex 3D modal sound models is expensive, and compression further increases pre-process times. Faster compression schemes (with better than “adaptive MLS” performance) would be useful. Preprocesses which directly estimate compressed representations would be helpful. Some applications may wish to use greater compression, which can be done at the cost of introducing noticeable sound differences. Runtime evaluation costs are reduced for lower MLS degree, m , which may be better for real-time applications. Compression factors are larger (smaller) if higher (lower) resolution meshes are used. Our perceptual weighting model is based on radiative power, which requires acoustic transfer models for best results. Our approximation model is based on average-case (relative ℓ_2) error of the sound input-output model, however large pointwise relative errors can still occur depending on the specific location of the contact (input) and the listener (output). Also, frequency-dependent hearing sensitivities vary between listeners, and with amplitude, and users may or may not be able to hear differences. Future perceptual models might exploit inter-mode masking effects, which can be significant. Symmetry handling is approximate, and future works could better exploit it when large perceptual errors are admissible, and when approximate symmetries exist. Mode compression can result in acoustic transfer data dominating memory footprints for large objects, and future work should address the need for perceptually based transfer models.

Acknowledgments

We would like to acknowledge collaborations with Daniel Schroeder, Changxi Zheng, Jeffrey Chadwick, Shoshana Yun, and Kavita Bala. We thank the anonymous reviewers for their constructive feedback. Mitsuba was used for all renderings. We acknowledge funding and support from the National Science Foundation (HCC-0905506, DGE-1144153), Intel, the John Simon Guggenheim Memorial Foundation, software from Advanced CAE Research (FastBEM) and Autodesk, and donations from Pixar. This research was conducted in conjunction with the Intel Science and Technology Center–Visual Computing. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or others.

A Derivation of relative mode importance

We now derive the expression for the relative average far-field pressure as proportional to $\|\mathbf{u}^j\| \|\mathbf{c}^j\|_F / (\omega^j)^2$ used when allocating relative mode errors. Given the pressure is $O(|p||q|)$, we estimate $|q|$ using the modal response to an impulse $\mathbf{f} \delta(t)$,

$$q(t) = \frac{1}{\omega} e^{-\frac{\alpha+\beta}{2}\omega^2 t} \sin(\omega t) (\mathbf{u} \bullet \mathbf{f}) \implies |q| \leq \frac{\|\mathbf{u} \bullet \mathbf{f}\|}{\omega} \stackrel{\|\mathbf{f}\|=1}{\leq} \frac{\|\mathbf{u}\|}{\omega}.$$

We estimate $O(|p|)$ as proportional to the square-root of (far-field) radiated power. Consider a unit-amplitude vibration of a single mode, with acoustic transfer $p(\mathbf{x})$ of (2). The time-averaged radiated power through a surrounding sphere S_R of radius R , is propor-

tional to $P_R = \int_{S_R} |p|^2 dS = \int_{S_R} |p(\mathbf{x})|^2 \sin \theta R^2 d\theta d\phi$ [Howe 1998]. For the far-field power ($R \rightarrow \infty$) only leading-order $1/r$ contributions from p contribute. Since the Hankel functions have $h_n^{(2)}(kr) \sim i^{n+1} e^{-ikr} / kr$ as $kr \rightarrow \infty$, and using the orthogonality of the spherical harmonics, we find that

$$P_\infty = \frac{1}{k^2} \sum_{n=0}^{\tilde{n}} \sum_{m=-n}^n |c_n^m|^2 = \frac{1}{k^2} \|\mathbf{c}\|_F^2.$$

We then estimate the mode-relative amplitude of the space- and time-averaged far-field transfer as $|p| = \sqrt{P_\infty} \sim \|\mathbf{c}\|_F / k$. It follows that the relative modal importance is given by $\|\mathbf{u}^j\| \|\mathbf{c}^j\|_F / (\omega^j)^2$.

References

- ADRIEN, J.-M. 1991. Representations of musical signals. MIT Press, Cambridge, MA, USA, ch. The Missing Link: Modal Synthesis, 269–298.
- AGARWAL, S., MIERLE, K., AND OTHERS. Ceres solver. <https://code.google.com/p/ceres-solver/>.
- ALEXA, M., AND MÜLLER, W. 2000. Representing animations by principal components. In *Computer Graphics Forum*, vol. 19, Wiley Online Library, 411–418.
- ALEXA, M., BEHR, J., COHEN-OR, D., FLEISHMAN, S., LEVIN, D., AND SILVA, C. T. 2001. Point set surfaces. In *Proceedings of the Conference on Visualization '01*, IEEE Computer Society, Washington, DC, USA, VIS '01, 21–28.
- ASHIHARA, K. 2007. Hearing thresholds for pure tones above 16khz. *The Journal of the Acoustical Society of America* 122, 3.
- BATHE, K.-J. 1996. *Finite Element Procedures*, second ed. Prentice Hall.
- BONNEEL, N., DRETTAKIS, G., TSINGOS, N., VIAUD-DELMON, I., AND JAMES, D. 2008. Fast Modal Sounds with Scalable Frequency-Domain Synthesis. *ACM Transactions on Graphics* 27, 3 (Aug.), 24:1–24:9.
- CHADWICK, J. N., AN, S. S., AND JAMES, D. L. 2009. Harmonic shells: A practical nonlinear sound model for near-rigid thin shells. *ACM Trans. Graph.* 28, 5 (Dec.), 119:1–119:10.
- CHADWICK, J. N., ZHENG, C., AND JAMES, D. L. 2012. Precomputed acceleration noise for improved rigid-body sound. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012)* 31, 4 (Aug.).
- COOK, P. R. 1995. Integration of physical modeling for synthesis and animation. In *Proceedings of the International Computer Music Conference*, 525–528.
- COOK, P. 2002. *Real Sound Synthesis for Interactive Applications*. A.K. Peters.
- FASTL, H., AND ZWICKER, E. 2006. *Psychoacoustics: Facts and models*, vol. 22. Springer.
- FLEISHMAN, S., COHEN-OR, D., AND SILVA, C. T. 2005. Robust moving least-squares fitting with sharp features. *ACM Trans. Graph.* 24, 3 (July), 544–552.
- GRELAUD, D., BONNEEL, N., WIMMER, M., ASSELOT, M., AND DRETTAKIS, G. 2009. Efficient and practical audio-visual rendering for games using crossmodal perception. In *13D '09: Proceedings of the 2009 symposium on Interactive 3D graphics and games*, ACM, New York, NY, USA, 177–182.
- GUENDELMAN, E., BRIDSON, R., AND FEDKIW, R. 2003. Non-convex rigid bodies with stacking. *ACM Trans. Graph.* 22, 3 (July), 871–878.

- GUSKOV, I., AND KHODAKOVSKY, A. 2004. Wavelet compression of parametrically coherent mesh sequences. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '04, 183–192.
- HOWE, M. S. 1998. *Acoustics of fluid-structure interactions*. Cambridge University Press.
- JAMES, D. L., AND PAI, D. K. 2003. Multiresolution green's function methods for interactive simulation of large-scale elastostatic objects. *ACM Trans. Graph.* 22, 1 (Jan.), 47–82.
- JAMES, D. L., BARBIČ, J., AND PAI, D. K. 2006. Precomputed Acoustic Transfer: Output-sensitive, accurate sound generation for geometrically complex vibration sources. *ACM Transactions on Graphics* 25, 3 (July), 987–995.
- JIN, J., GARLAND, M., AND RAMOS, E. A. 2009. MIs-based scalar fields over triangle meshes and their application in mesh processing. In *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games*, ACM, New York, NY, USA, I3D '09, 145–153.
- KOLAROV, K., AND LYNCH, W. 1997. Wavelet compression for 3d and higher-dimensional objects. In *Proc. of SPIE Conference on Applications of Digital Image Processing XX, Volume 3164*.
- LIPMAN, Y., CHEN, X., DAUBECHIES, I., AND FUNKHOUSER, T. 2010. Symmetry factored embedding and distance. *ACM Trans. Graph.* 29, 4 (July), 103:1–103:12.
- LIU, Y. J. 2009. *Fast Multipole Boundary Element Method: Theory and Applications in Engineering*. Cambridge University Press, Cambridge.
- MARTINET, A., SOLER, C., HOLZSCHUCH, N., AND SILLION, F. X. 2006. Accurate detection of symmetries in 3d shapes. *ACM Trans. Graph.* 25, 2 (Apr.), 439–464.
- MITRA, N. J., GUIBAS, L. J., AND PAULY, M. 2006. Partial and approximate symmetry detection for 3d geometry. *ACM Trans. Graph.* 25, 3 (July), 560–568.
- MITRA, N. J., PAULY, M., WAND, M., AND CEYLAN, D. 2012. Symmetry in 3d geometry: Extraction and applications. In *EUROGRAPHICS State-of-the-art Report*.
- O'BRIEN, J. F., SHEN, C., AND GATCHALIAN, C. M. 2002. Synthesizing sounds from rigid-body simulations. In *ACM SIGGRAPH Symposium on Computer Animation*, 175–181.
- PAI, D. K., DOEL, K. V. D., JAMES, D. L., LANG, J., LLOYD, J. E., RICHMOND, J. L., AND YAU, S. H. 2001. Scanning physical interaction behavior of 3d objects. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '01, 87–96.
- PAULY, M., GROSS, M., AND KOBELT, L. P. 2002. Efficient simplification of point-sampled surfaces. In *Proceedings of the conference on Visualization '02*, IEEE Computer Society, Washington, DC, USA, VIS '02, 163–170.
- RAGHUVANSHI, N., AND LIN, M. C. 2006. Interactive sound synthesis for large scale environments. In *Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games*, ACM, New York, NY, USA, I3D '06, 101–108.
- RICHMOND, J. L. 2000. *Automatic measurement and modeling of contact sounds*. Master's thesis, University of British Columbia.
- SCHAEFER, S., MCPHAIL, T., AND WARREN, J. 2006. Image deformation using moving least squares. *ACM Trans. Graph.* 25, 3 (July), 533–540.
- SCHRÖDER, P., AND SWELDENS, W. 1995. Spherical wavelets: Efficiently representing functions on the sphere. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '95, 161–172.
- SEO, J., IRVING, G., LEWIS, J. P., AND NOH, J. 2011. Compression and direct manipulation of complex blendshape models. *ACM Trans. Graph.* 30, 6 (Dec.), 164:1–164:10.
- SHABANA, A. A. 1990. *Theory of Vibration, Volume II: Discrete and Continuous Systems*. Springer-Verlag, New York, NY.
- SHEN, C., O'BRIEN, J. F., AND SHEWCHUK, J. R. 2004. Interpolating and approximating implicit surfaces from polygon soup. *ACM Trans. Graph.* 23, 3 (Aug.), 896–904.
- SIMARI, P., KALOGERAKIS, E., AND SINGH, K. 2006. Folding meshes: Hierarchical mesh segmentation based on planar symmetry. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SGP '06, 111–119.
- SORKINE, O., AND COHEN-OR, D. 2004. Least-squares meshes. In *Proceedings of Shape Modeling International*, IEEE Computer Society Press, 191–199.
- TSINGOS, N., GALLO, E., AND DRETTAKIS, G. 2004. Perceptual audio rendering of complex virtual environments. *ACM Trans. Graph.* 23, 3 (Aug.), 249–258.
- VAN DEN DOEL, K., AND PAI, D. K. 1996. Synthesis of shape dependent sounds with physical modeling. In *Intl Conf. on Auditory Display*.
- VAN DEN DOEL, K., KRY, P. G., AND PAI, D. K. 2001. FoleyAutomatic: Physically Based Sound Effects for Interactive Simulation and Animation. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, 537–544.
- VAN DEN DOEL, K., KNOTT, D., AND PAI, D. K. 2004. Interactive simulation of complex audiovisual scenes. *Presence: Teleoper. Virtual Environ.* 13, 1, 99–111.
- ZHENG, C., AND JAMES, D. L. 2010. Rigid-body fracture sound with precomputed soundbanks. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2010)* 29, 3 (July).
- ZHENG, C., AND JAMES, D. L. 2011. Toward high-quality modal contact sound. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2011)* 30, 4 (Aug.).