Deep-Modal: Real-Time Impact Sound Synthesis for Arbitrary Shapes

Xutong Jin
jinxutong@pku.edu.cn
Peking University

Sheng Li*
lisheng@pku.edu.cn
Peking University

Tianshu Qu
qutianshu@pku.edu.cn
Peking University

Dinesh Manocha
dmanocha@umd.edu
University of Maryland

Guoping Wang
wgp@pku.edu.cn
Peking University

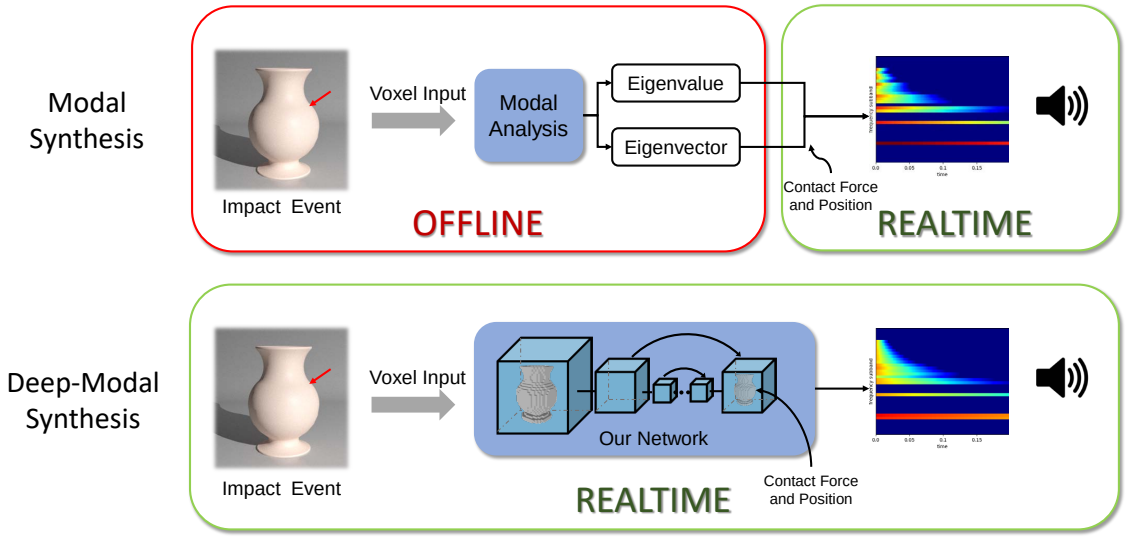

Figure 1: Modal sound vs. our Deep-Modal sound: For a 3D object of arbitrary shape, prior synthesis algorithms (top) perform expensive modal analysis preprocessing. Our novel learning-based synthesis algorithm, Deep-Modal, can generate impact sounds in real-time without processing on modal analysis.

ABSTRACT

Model sound synthesis is a physically-based sound synthesis method used to generate audio content in games and virtual worlds. We present a novel learning-based impact sound synthesis algorithm called Deep-Modal. Our approach can handle sound synthesis for common arbitrary objects, especially dynamic generated objects, in real-time. We present a new compact strategy to represent the mode data, corresponding to frequency and amplitude, as fixed-length vectors. This is combined with a new network architecture that can convert shape features of 3D objects into mode data. Our network is based on an encoder-decoder architecture with the contact positions of objects and external forces embedded. Our method can synthesize interactive sounds related to objects of various shapes at any contact position, as well as objects of different materials and sizes. The synthesis process only takes 0.01s on a GTX 1080 Ti GPU. We show the effectiveness of Deep-Modal through extensive evaluation using different metrics, including recall and precision of prediction, sound spectrogram, and a user study.

CCS CONCEPTS

• **Applied computing** → **Sound and music computing**.

KEYWORDS

impact sound; sound synthesis; frequency; amplitude; neural networks; impact; shape feature; dynamic object

ACM Reference Format:
Xutong Jin, Sheng Li, Tianshu Qu, Dinesh Manocha, and Guoping Wang. 2020. Deep-Modal: Real-Time Impact Sound Synthesis for Arbitrary Shapes. In *Proceedings of the 28th ACM International Conference on Multimedia (MM*

*corresponding author

ACMMM '20, 12-16 October 2020 , Seattle, WA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413572

'20), October 12–16, 2020, Seattle, WA, USA. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3413572

1 INTRODUCTION

In virtual reality and multimedia applications, modal synthesis is widely used for physically-based audio content generation [18, 24, 37, 38]. These modal synthesis methods calculate characteristic vibration modes according to the shape and material properties of an object instead of using any pre-recorded audio samples. This process is also called *modal analysis*. The key to modal analysis is producing a generalized eigenvalue decomposition of large sparse mass and stiffness matrices, which is computationally expensive (time cost varies from seconds to hours depending on the scale of the finite elements) and should be performed in the offline stage, as illustrated at the top of Figure 1. Pre-processed modal data of a 3D object (including the eigenvalue and the eigenvector) are stored for each object and used to synthesize sound at run-time, whenever a contact event occurs at the contact position with a specific force.

Traditionally, the precomputed modal data depends strongly on the shape, size, and material of a 3D object. Each object in a scene generally requires computing expensive, object-specific modal data. Subsequently, many researchers have proposed approximate techniques that apply the modal data of one object to objects with different sizes or materials, but the same geometric shape [26, 48]. However, these modal analysis based methods are incapable of dealing with new objects with arbitrary shapes such as fracture or deformable objects in runtime applications. Therefore, it is preferable for real-time algorithms to synthesize sounds for such objects without any preprocessing and these methods require further investigation.

Main Contributions: we overcome these major limitations of prior modal synthesis methods and present a novel learning-based approach that can handle arbitrary objects with different shapes, sizes, and materials in real-time without preprocessing on modal analysis. We present a sound synthesis network, Deep-Modal, that can convert the shape features of 3D objects into mode data (a compact representation) to facilitate the sound synthesis and audio content generation in real-time (see the bottom of Figure 1). Overall, our main contributions include:

- We present a novel learning-based method that can synthesize impact sound with high efficiency. To the best of our knowledge, this is the first approach that can synthesize interactive sound of an object with different shapes in real-time without preprocessing on the modal analysis of an object.
- We design a novel Deep-Modal network that converts the shape features of 3D objects into compact mode data that can be used to synthesize a sound signal.
- Our network is a lightweight network and can synthesize reliable results close to the ground truth. Any common objects with different shapes, sizes, or materials can be available to synthesize the impact sound at any contact position with an external force. Our method can handle unseen objects in highly dynamic scenes in real-time. We highlight the performance on complex benchmarks.

2 RELATED WORK

We briefly review the related aspects of modal synthesis and deep learning models for sound synthesis.

2.1 Modal Synthesis

For offline applications, wave-based methods can produce high-quality sound [41]. For interactive virtual environments, modal synthesis [18, 24, 38] has been widely used for rigid-body objects. Modal analysis methods compute the characteristic vibration modes of a 3D object using eigendecomposition-based preprocessing. Based on modal sound synthesis, richer sound effects such as knocking, sliding, and friction on 3D objects can be simulated [37]. Other techniques are based on precomputed acoustic transfer [11] to generate more realistic sounds. The modal analysis techniques have also been combined with sound propagation [2, 25, 28]. Acceleration noise synthesis [4] was proposed to synthesize sounds produced when an object experiences abrupt rigid-body acceleration. More accurate damping models [30, 31] and contact models [49] have been proposed. Many techniques have also been proposed to accelerate the performance of modal sound synthesis. These methods include using the parallel computing capabilities of the GPU to accelerate the speed of the run-time stage [46] and using an approximation to reduce the computational complexity [3].

However, the computation of modal analysis strongly depends on the shape, size, and material parameters of a 3D object, so the generated modal data is object-specific. Zheng et al. [48] proposed a method to simulate sound models for all scaled versions of a rigid body with the same shape. To simulate fracture sounds, they used fragmented pre-processed modal data to synthesize the sound of fragments with different sizes, and they approximated all the fragments' shapes using ellipses to avoid the shape dependence of the sound modal. Z. Ren et al. [26] point out the correlation between eigenvalue and material coefficients, meaning that a modal data can be applied to objects with different materials. However, these modal analysis-based preprocessing methods are still constrained by geometric shape dependence, i.e. one set of modal data is only applicable to one geometric shape for the 3D object. In contrast to these methods, our approach is general and applicable to arbitrary 3D objects with no constraints on shape, size, or material properties.

2.2 Deep Learning Methods

There has been a lot of work exploring three-dimensional geometric features learning, including perspective-based methods [12, 33], which generate a corresponding image through some specified perspectives and regard the combination of 2D image features as a 3D geometry feature; voxel-based methods [15, 22, 44]; and point cloud-based methods [21, 23, 43].

Recently, there has also been a considerable amount of work using neural networks to deal with audio, producing methods such as WaveNet [39], Tacotron [42], and Tacotron 2 [29], which use neural networks to convert text features into audio features to perform end-to-end speech synthesis. In particular, some works focus on audio-visual connected deep learning. A recurrent neural network was used to predict audio features from video [19], and the impact sound of an object being struck can be used to improve the accuracy of object classification using ISNN [32]. Ephrat et al. [6] built

a network-based model for speaker-independent speech separation by converting the audio-visual features. The neural network is also employed for physics-based sound-related simulation, as proposed in [7, 20, 34, 35] for sound propagation and in [47] for predicting the shape of the colliding object through the simulated sound.

Our learning-based algorithm is also motivated by these methods, building the audio-visual connection on the problem of modal sound synthesis and designing a lightweight network.

3 FOUNDATION

We first introduce the foundations of modal sound synthesis. We begin with the linear deformation equation for a 3D linear elastic dynamics model [26]:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f}, \tag{1}$$

where $\mathbf{x}$ is nodal displacements and $\mathbf{M}, \mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K}, \mathbf{K}$ represent the mass, Rayleigh damping, and stiffness matrices, respectively. $\mathbf{f}$ represents the external force, which stimulates the vibrations.

Through generalized eigenvalue decomposition $\mathbf{KU} = \mathbf{\Lambda MU}$, the system can be decoupled into the following form [26]:

$$\ddot{\mathbf{q}} + (\alpha\mathbf{I} + \beta\mathbf{\Lambda})\dot{\mathbf{q}} + \mathbf{\Lambda}\mathbf{q} = \mathbf{U}^T\mathbf{f}, \tag{2}$$

where $\mathbf{\Lambda}$ is a diagonal matrix and $\mathbf{q}$ satisfies $\mathbf{x} = \mathbf{U}\mathbf{q}$. The solution to Equation 2 is a bank of damped sinusoidal waves. Each wave represents a mode. The i'th mode is:

$$q_i = a_i e^{-c_i t} \sin\left(2\pi\omega_i t + \theta_i\right), \tag{3}$$

where ω_i is the frequency of the mode (damped natural frequency), c_i is the damping coefficient, a_i is the excited amplitude, and θ_i is the initial phase. We currently directly use the vibration modes for sound synthesis [18, 24, 36, 37] because we treat the sound source as a point without considering the effect of the acoustic transfer[11].

As the object begins at rest and is then struck at $t = 0$, we can assume θ_i to be zero. Solving Equation 2, we get

$$c_i = \frac{1}{2}\left(\alpha + \beta\lambda_i\right), \tag{4}$$

$$\omega_i = \frac{1}{2\pi}\sqrt{\lambda_i - \left(\frac{\alpha + \beta\lambda_i}{2}\right)^2}, \tag{5}$$

where λ_i represents the i-th mode eigenvalue. Therefore, the damping coefficient c_i can be solved when material properties (α, β) and frequency (ω_i) are known. We only focus on the frequency and amplitude of the modes.

As the number of modes varies with different objects, it is difficult to represent modes with fixed-length vectors. KleinPAT [40] accelerates acoustic transfer precomputation by packing modes into several partitions, which need a time-domain algorithm and are not applicable to modal analysis. We fix the lengths of mode vectors that is similar to the strategy described in [24]. We pack modes into sub-bands that are evenly distributed on the Mel-scale due to their perceptual similarity. Modes in a sub-band are packed into a compact mode by summing up the amplitudes. The frequency of the compact mode is the central frequency of the sub-band.

We use a vector to represent the amplitude of all compact modes. As the frequencies of the modes are discrete, there are some sub-bands with no modes inside, meaning that the corresponding compact modes are empty. We use another binary vector as a mask for compact modes: 1 indicates the compact mode is nonempty and 0 indicates empty. These two vectors have fixed lengths, and we use them to represent a mode.

Simply regressing on the amplitude in each Mel sub-band will lead to over-smoothed results, which is also indicated in the work of speech synthesis [29]. We use the form of the binary mask to reduce over-smoothness as the amplitude in empty compact mode can be discarded by the binary mask. We set the amplitude in an empty compact mode as the amplitude of the nearest non-empty one. This is a trick to make network regression easier without affecting correctness. Figure 2 shows how we convert the origin mode data to amplitude and mask, which is used in our method.

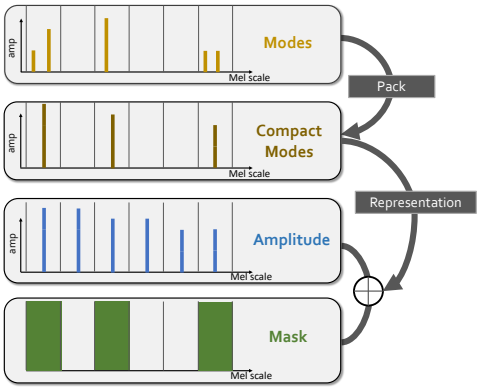

Figure 2: Illustration of compact mode data: A compressed form of conversion from the origin mode to amplitude and mask for our learning-based approach.

4 DEEP-MODAL: OUR LEARNING METHOD

In principle, we formalized the physics-based rigid-body sound synthesis process as a supervised learning problem. Our goal is to correctly predict the audio signal according to the shape, material, size, external force, and contact position of the object. With approximation, the effect of the material and size on the sound can be separated and can then be treated in independent post-processing [26, 48]. In addition, the effect of external forces can be separated as described in subsection 4.1. Following this strategy, we design our network, which focuses on correctly predicting the impact sound when the geometric shape and contact position are varied and the size, material, and external force are fixed. Finally, we add separate post-processing to deal with the variation in material, size, and external force. The sound synthesis process can be formally described as mapping $\mathbf{g}$ from input tuple $\mathbf{x} = \{\mathbf{m}, \mathbf{p}\}$ to estimate $\mathbf{e}$ of ground-truth sound $\mathbf{s}$. The input tuple includes 3D model $\mathbf{m}$ and contact position $\mathbf{p}$. We use a convolutional neural network parameterized by a set of weights $\boldsymbol{\theta}$ to represent $\mathbf{g}$. We use a large dataset to obtain the optimal parameters $\widehat{\boldsymbol{\theta}}$ through supervised learning. The dataset contains N pairs of input tuples and corresponding sounds, $\mathcal{D}_N = \left\{\left(\mathbf{x}^1, \mathbf{s}^1\right), \ldots, \left(\mathbf{x}^N, \mathbf{s}^N\right)\right\}$. Our supervised learning is to minimize the average distance between the prediction of the network $\mathbf{e} = \mathbf{g}\left(\mathbf{x}; \boldsymbol{\theta}\right)$ and ground-truth sound $\mathbf{s}$:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{N}\sum_{n=1}^{N} \ell\left(\mathbf{s}^n, \mathbf{g}\left(\mathbf{x}^n; \boldsymbol{\theta}\right)\right), \tag{6}$$

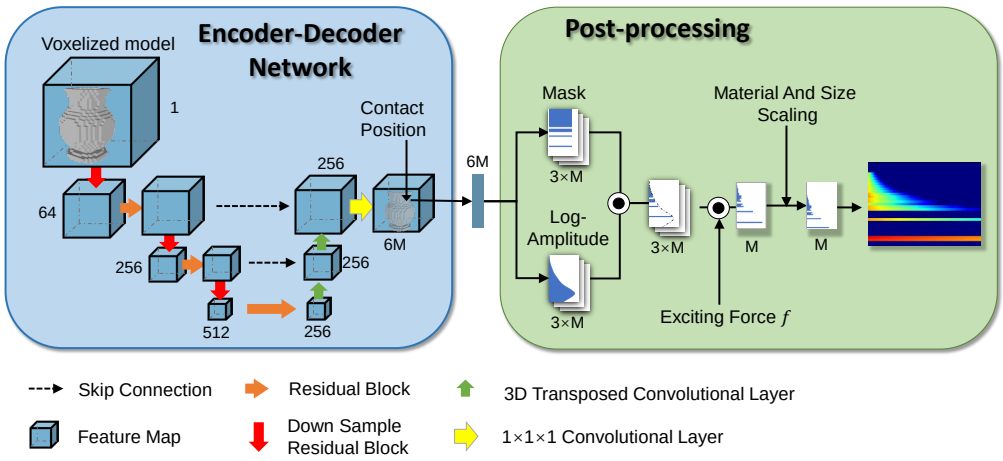

Figure 3: Architecture of our Deep-Modal sound synthesis. Our Deep-Modal network takes voxels as inputs and outputs a sound feature map. A contact position-related sound feature is selected from the sound feature map. The sound feature is viewed as a mask and log-amplitude, which are combined with an exciting force into compact modes. The compact modes are post-processed to fit the input material and size. At runtime, we use these audio features to synthesize sound.

where ℓ is the loss function used to measure the distance between two impact sounds. Next, we will introduce our method in detail.

4.1 Representation

Input Model. Generally, a mesh is used to represent a 3D object for interactive applications. A conversion from mesh to volumetric data using the finite element partition is necessary for modal sound synthesis. The typical representation of the volume data includes tetrahedral cells and hexahedral cells. The hexahedral cells can also be regarded as a binary voxel representation used for the 3D convolutional neural network, as described in [44]. The binary voxel representation treats a 3D object as a probability distribution of binary variables in the voxel grid. We therefore employ this hexahedral form of representation, which can bridge modal sound synthesis and deep learning, and each 3D object is represented using a binary tensor: 1 indicates a voxel is inside the object, while 0 indicates a voxel is outside of the object (i.e. in space).

Sound feature. We first demonstrate how the effect of an external force can be separated. In the modal synthesis method described in section 3, any external force can be decomposed into the resultant force from three orthogonals, i.e. a linear combination of unit forces with the same contact position: $\mathbf{f} = k_1\mathbf{f_1} + k_2\mathbf{f_2} + k_3\mathbf{f_3}$. By solving Equation 2, the amplitudes of modes $\mathbf{a}$ excited by $\mathbf{f}$ can be decomposed into a linear combination of the amplitudes excited by the unit force:

$$\begin{aligned}\mathbf{a} &= \mathbf{U}^T\mathbf{f} \oslash \boldsymbol{\omega} \\ &= \mathbf{U}^T\left(\mathbf{k_1f_1} + \mathbf{k_2f_2} + \mathbf{k_3f_3}\right) \oslash \boldsymbol{\omega} \\ &= \left(\mathbf{U}^T\mathbf{k_1f_1} \oslash \boldsymbol{\omega}\right) + \left(\mathbf{U}^T\mathbf{k_2f_2} \oslash \boldsymbol{\omega}\right) + \left(\mathbf{U}^T\mathbf{k_3f_3} \oslash \boldsymbol{\omega}\right) \\ &= k_1\left(\mathbf{U}^T\mathbf{f_1} \oslash \boldsymbol{\omega}\right) + k_2\left(\mathbf{U}^T\mathbf{f_2} \oslash \boldsymbol{\omega}\right) + k_3\left(\mathbf{U}^T\mathbf{f_3} \oslash \boldsymbol{\omega}\right) \\ &= k_1\mathbf{a_1} + k_2\mathbf{a_2} + k_3\mathbf{a_2},\end{aligned} \tag{7}$$

where $\boldsymbol{\omega}$ represents the frequencies of modes and the symbol $\oslash$ denotes element-wise division. $\mathbf{a_1}, \mathbf{a_2}, \mathbf{a_3}$ represent the amplitudes of modes excited by unit force $\mathbf{f_1}, \mathbf{f_2}, \mathbf{f_3}$. Therefore, the effect of the external force is separated. As the linear relationship of $\mathbf{a}, \mathbf{a_1}, \mathbf{a_2}, \mathbf{a_3}$ remains after the compression of modes described in section 3, we define our force-unrelated sound feature as three compact modes excited by three unit forces. It should be noted that we regard sub-bands with unexcited modes as having no modes inside, so the binary mask of these compact modes can be different. We also transformed amplitude to log-amplitude, which is consistent with decibel.

Therefore, the ground-truth sound can be denoted as:

$$\mathbf{s} = \{(\mathbf{s}_{1,b}, \mathbf{s}_{1,a}), (\mathbf{s}_{2,b}, \mathbf{s}_{2,a}), (\mathbf{s}_{3,b}, \mathbf{s}_{3,a})\},$$

and the prediction sound of the network can be denoted as:

$$\mathbf{e} = \{(\mathbf{e}_{1,b}, \mathbf{e}_{1,a}), (\mathbf{e}_{2,b}, \mathbf{e}_{2,a}), (\mathbf{e}_{3,b}, \mathbf{e}_{3,a})\},$$

where $\mathbf{s}_{i,b}, \mathbf{s}_{i,a}$ represent the mask and log-amplitudes of compact modes excited by i-th unit force and each can be represented as a vector of length M; M is the number of all Mel sub-bands. Therefore, we use a vector of length $6M$ to represent the feature of a sound. The element in $\mathbf{e}$ is the estimation of corresponding element in $\mathbf{s}$.

We employed loss function ℓ for sound feature as a combination of mean squared error ℓ_1 for log-amplitude and binary cross-entropy loss ℓ_2 for the mask:

$$\ell(\mathbf{s}, \mathbf{e}) = \frac{1}{3}\sum_{i=1}^{3}\left(\lambda_1\ell_1(\mathbf{s}_{i,a}, \mathbf{e}_{i,a}) + \lambda_2\ell_2(\mathbf{s}_{i,b}, \mathbf{e}_{i,b})\right), \tag{8}$$

where λ_1, λ_2 are tunable parameters representing the contribution of log-amplitudes and masks.

Position p The specific position is usually not processed as the input directly for position-related predictions in the neural network; however, this position is generally used to make a retrieval from the network output that includes all possible positions, such as in image segmentation [14, 27] and human pose estimation [17, 45]. Inspired by these works, we employ a similar strategy to predict position-related sounds.

Our neural network first predicts the audio features of all possible positions. These audio features form a feature map called a *sound feature map*. Each component in this sound feature map has an apple-to-apple correspondence with each voxel of a voxelized model. Those contact positions located within a voxel are treated as the same by approximation. For contact positions within a voxel, the impact sound feature can be obtained by selecting the corresponding component from the sound feature map. Each component in the sound feature map is a vector with a length of $6M$ according to our sound feature representation described above. For each training object, the loss function ℓ in Equation 8 is the sum over all possible contact position voxels to update the network weights. We down-sample the possible impact locations to a lower resolution than the input voxel model so that we can obtain higher speed at the cost of accuracy.

4.2 Network Architecture

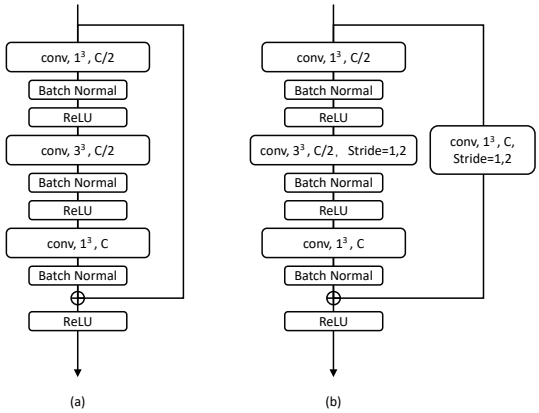

Figure 4: Residual block used in our network. (a) standard residual block; (b) downsampling residual block when stride=2 or skip connection when stride=1.

To predict position-related feature maps, an encoder-decoder structure is generally used [14, 17, 27, 45]. Our network is also designed based on such a structure, as shown in Figure 3. We employ the residual block [9] instead of the traditional convolutional neural network to construct a deeper network without optimization instabilities. The residual block contains three convolutional layers and a skip connection as shown in Figure 4. In our network architecture, the encoder part of the network is composed of a stack of downsampling residual blocks (Figure 4(b)) and residual blocks (Figure 4(a)), and the decoder part is composed of a stack of 3D transposed convolutional layers. The encoder part is responsible for extracting spatial features of different scales, and the decoder part can convert the spatial features into audio features related to each position. Skip connections are residual blocks (Figure 4(b)) used to merge spatial features of different scales from the encoder. The input of the network is a voxel representation of the 3D object, and the output is a sound feature map. The number of output feature channels is $6M$ according to our sound feature representation.

4.3 Post-processing

To synthesize impact sounds for objects of any material and size, a post-processing stage is employed to integrate different material parameters and sizes under approximation.

Network prediction sound feature is a vector of length $6M$. This vector can be regarded as 6 vectors of length M according to our sound feature representation $\mathbf{e}$, as described in subsection 4.1.

Suppose the external force is $\mathbf{f} = k_1\mathbf{f_1} + k_2\mathbf{f_2} + k_3\mathbf{f_3}$. Amplitudes $\mathbf{a}$ of compact modes after excitation can be obtained from $\mathbf{e}$ (value should be first restored from log-amplitude to amplitude) and $\mathbf{f}$:

$$a_i = \sum_{j=1}^{3} k_j H(e_{j,b,i} - t) e_{j,a,i}, \tag{9}$$

where the Heaviside function $H(\cdot)$ returns 1 if the argument is positive and 0 otherwise. t is a tunable threshold. $e_{j,b,i}, e_{j,a,i}$ represent the i-th element of mask $\mathbf{e}_{j,b}$ and amplitude $\mathbf{e}_{j,a}$, respectively. When the mask value $e_{j,b,i}$ is greater than t, the i-th sub-band is considered to contain an excited mode and its corresponding amplitude should be retained; otherwise it should be discarded.

The damped natural frequency of a sub-band $\boldsymbol{\omega}$ is the central frequency of this sub-band. We convert it to an undamped natural frequency before post-processing. To utilize mode data of fixed material and size to synthesize sounds for objects of different materials and sizes, we need to calibrate $\mathbf{a}$, $\boldsymbol{\omega}$ as follow.

Stiffness matrix $\mathbf{K}$ is linearly related to stiffness, and mass matrix $\mathbf{M}$ is linearly related to the reciprocal of density. However, the effect of Poisson's ratio is more complicated. Inspired by a method to estimate the feature of mode proposed by Z. Ren et al. [26] , we only scale $\mathbf{a}$, $\boldsymbol{\omega}$ for stiffness and density. If the stiffness changes from E to $\sigma_1 E$ and the density changes from ρ to $\sigma_2\rho$, then the undamped natural frequency $\boldsymbol{\omega}$ and amplitude $\mathbf{a}$ will be scaled as:

$$\boldsymbol{\omega} \leftarrow \sigma_1^{1/2}\sigma_2^{-1/2}\boldsymbol{\omega}, \qquad \mathbf{a} \leftarrow \sigma_2^{-1/2}\mathbf{a}. \tag{10}$$

As in the size scaling strategy of Zheng and James [48], if the size changes from l to $\sigma_3 l$, then the undamped natural frequency $\boldsymbol{\omega}$ and amplitude $\mathbf{a}$ will be scaled as:

$$\boldsymbol{\omega} \leftarrow \sigma_3^{-1}\boldsymbol{\omega}, \qquad \mathbf{a} \leftarrow \sigma_3^{-3/2}\mathbf{a}. \tag{11}$$

In summary, the scaling formula for changed materials and size is:

$$\boldsymbol{\omega} \leftarrow \sigma_1^{1/2}\sigma_2^{-1/2}\sigma_3^{-1}\boldsymbol{\omega}, \qquad \mathbf{a} \leftarrow \sigma_2^{-1/2}\sigma_3^{-3/2}\mathbf{a}. \tag{12}$$

After scaling, we restore the undamped natural frequency $\boldsymbol{\omega}$ back to the damped natural frequency. The damping coefficients $\mathbf{c}$ can also be obtained. Finally, $\mathbf{a}$, $\boldsymbol{\omega}$, $\mathbf{c}$ can be used to synthesize the waveform of sound S through additive synthesis as:

$$S(t) = \sum_{i=1}^{M} a_i e^{-c_i t} \sin(2\pi\omega_i t). \tag{13}$$

where a_i, ω_i, c_i represent the element of $\mathbf{a}$, $\boldsymbol{\omega}$, $\mathbf{c}$ respectively, and M represents the number of Mel sub-bands.

5 EXPERIMENT AND RESULTS

5.1 Dataset and Implementation

First, we use modal synthesis [10] to generate our dataset. The 3D models are from ModelNet40 [44], which has already been divided into a training set and a test set with sizes 9843 and 2468

respectively. We take out another 20% from the training set to use as the validation set.

All 3D objects are set to be the same material with the same scale when training. The hexahedral voxel representation of each object has a resolution of 32 ×32 ×32. For those thin objects, i.e. the thickness less than a voxel size, they will be processed with one-voxel thickness. We apply modal sound synthesis to the hexahedral voxel mesh of objects in the dataset.

Generally, the audible frequency range is 20HZ-20000HZ. However, through analysis, we find that objects with modes less than 100HZ only account for 0.6% of the dataset, and objects with modes less than 10000HZ account for 99.9% of the dataset. Because higher frequency modes in impact sound are less important due to relatively large damping coefficients and lower perceptual distinguishability, we compress the frequency range in practice to 100HZ-10000HZ to reduce the learning overhead.

The resolution of the sound feature map is set to 16 ×16 ×16, and we set the number of Mel sub-bands as $M = 32$. We normalize the log-amplitude to 0-1. For each model, we select all possible struck positions in the sound feature map to generate a corresponding sound feature. The echo combination of input voxel model, struck position, and output sound feature is regarded as a data point, as described in subsection 4.1. There are about 800,000 data points for training, 200,000 for validation, and 200,000 for testing.

We use $\lambda_1 = \lambda_2 = 1$ for loss function. We train the network with an Adam optimizer and a learning rate of 0.02, which is reduced by half every 20 epochs. We set the batch size to 64 and train 100 epochs for convergence. The training process took around 4 hours.

5.2 Experiments and Evaluation

We conducted the experiments and demonstrated the results of our method's prediction accuracy, sound fidelity, and efficiency through comparisons. We randomly selected some 3D objects from the test set and used modal synthesis and Deep-Modal to synthesize sound. The input voxelization model for these two methods is the same. The output sound of both methods is compacted to 100HZ-10000HZ which is consistent with our dataset. There is no other contact with the object except the excitation force. All the methods to be compared adopt the same configurations.

5.2.1 Prediction Accuracy. To our knowledge, our work is the first modal synthesis approach based on learning. We set a baseline, called the *shape-matching method*, as the spatial feature-related impact sound for comparison. We generated these sounds based on a 3D CNN model pre-trained on ModalNet40. We computed the second-to-last layer feature from the pre-trained model for the test object, then searched the training set for the best-matched object. We then produced the best-matched object's ground-truth sound from a random contact position. The 3D CNN pre-trained network model we selected is 3D Resnet [8]. The pre-trained model achieves 85% average accuracy for 3D object identification on ModelNet40.

We measure both our approach and the baseline on our test set. For each object from the test set, we randomly select a contact position and predict the mask and amplitudes. We then evaluate the prediction by comparing it with the ground truth. Our metric of comparison includes recall and precision of mask and mean squared error (MSE) of amplitudes. The mean results are shown in Table 1. As can be seen from the table, our method is superior to the baseline on all the metrics.

Table 1: System evaluation using MSE of amplitudes, recall and precision of mask prediction. Our method shows good performance on all these metrics.

Algorithm	MSE	Recall	Precision
Ours	0.0054	67.8%	71.8%
shape matching	0.0154	60.8%	63.4%

5.2.2 Sound Fidelity and User Evaluation. To examine Deep-Modal's ability to handle arbitrary objects with different materials, we demonstrate the results of impact sounds for various 3D objects from our test set with different materials (including common objects in the real world such as a ceramic bowl, a glass bottle, a steel cup, a wood door, and a plastic Xbox) between Deep-Modal and the ground truth (Modal Synthesis). The hit point on a surface is randomly selected with an impact force perpendicular to the surface. As shown in Figure 5, Deep-Modal can synthesize sounds for different materials and scales. These results are close to ground truth using the visualization of the sound spectrogram.

Considering that the user's perception is important in evaluating the quality of sound, especially for an interactive media-content application or virtual/augmented reality, we also conducted a user study through a Turing test similar to [19]. In total, 163 participants (each with normal hearing and wearing earphones) were enrolled in this test. We showed them two videos of a series of impact events – one playing the ground truth from modal synthesis, the other playing a synthesized sound from our Deep-Modal, as shown in Figure 5 and the supplementary material. They were asked to distinguish which one is the ground truth, i.e. a two-alternative forced-choice (2AFC). Our null hypothesis is that the two audio clips cannot be distinguished and are equally likely to be chosen. We employed a two-sided binomial test for our hypothesis.

We collected all the responses and found that the ground truth is preferred by 51.53% of the total responses and there is no significant evidence to indicate the difference between our model and the ground truth ($p = 0.75$). This validates the fidelity of our method for various sound synthesis applications, meaning that the participants are barely able to distinguish the truth. Furthermore, we provided a 7-point Likert scale (1 for totally different and 7 for totally same) to measure their similarity. The results show that the mean score is 5.11 with relatively high similarity.

To examine the Deep-Modal's ability to handle objects with different geometric shapes, we also demonstrate the results of impact sounds of various 3D objects with a ceramic material for our method and the ground truth, as shown in Figure 6. Accordingly, the 2^{nd} Turing test is performed on the same participants as the 1^{st} test. We found that the ground truth is preferred by 63.19% of the total responses. There is significant evidence to indicate there is a difference between our model and the ground truth ($p < 0.01$), which means some participants can distinguish our result from the ground truth. This is because our network cannot capture harmonic series well as, shown in Figure 6, and our audio sounds less clear than the

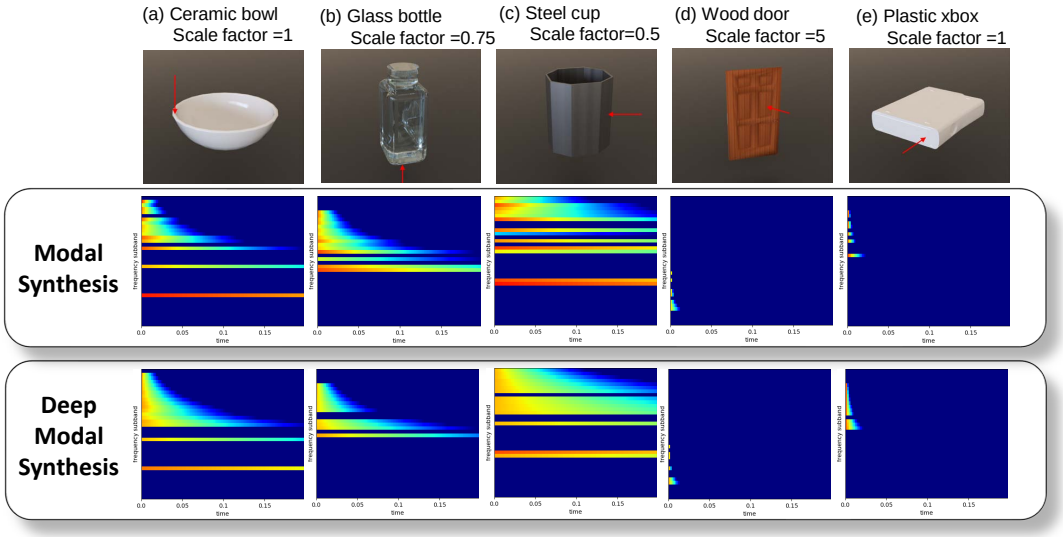

Figure 5: Sound synthesis of various 3D objects with different materials and sizes using spectrogram visualization.

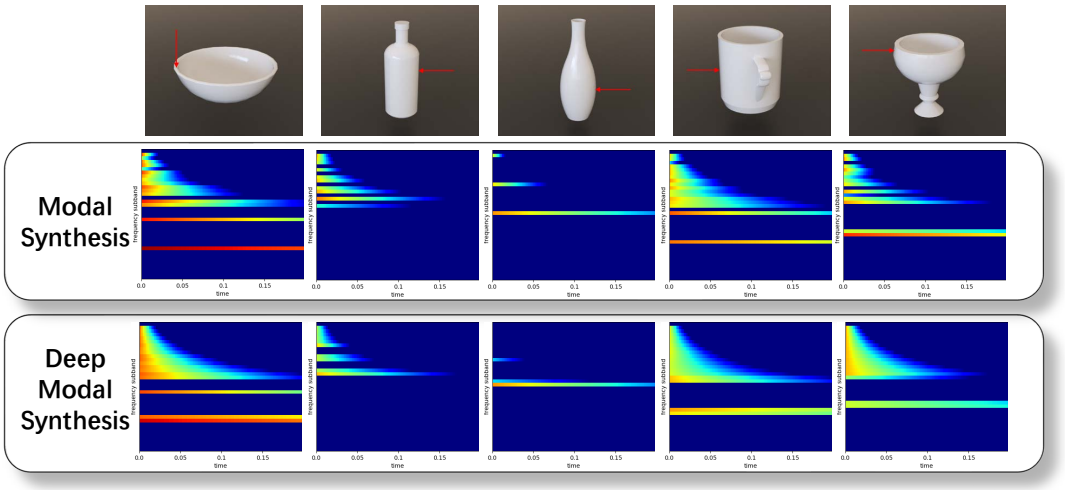

Figure 6: Sound synthesis of various 3D objects with the same ceramic material using spectrogram visualization.

ground truth. This disadvantage is obvious when the material is ceramic. Furthermore, the results from a 7-point Likert scale used to measure the similarity show that the mean score is 5.45. This validates the fidelity of our method to some extent.

To examine Deep-Modal's ability to handle different positions or different impacts of an object, we demonstrate the results of impact sound of a ceramic bowl, as shown in Figure 7. Deep-Modal can synthesize the results that are close to the ground truth. Accordingly, we performed the 3^{rd} Turing test using the same participants as the 1^{st} test. We found that the ground truth is preferred by 46.01% of the total responses, and there is no significant evidence to indicate the difference between our method and the ground truth (p = 0.35). Furthermore, the results from a 7-point Likert scale show that the mean score is 5.54. This validates the fidelity of our method.

Overall, we highlight the high fidelity of our method when compared to the ground truth for various aspects.

5.2.3 Efficiency. Deep-Modal can predict mode data for any shape in real-time. We show the comparison on time cost using four models with different geometric complexities from the test set in Table 2. All the methods must perform voxelization and this process can be completed in real-time on a GPU [13] (less than 3ms). Time cost of both modal synthesis and our method are counted from the end of voxelization to the moment when the frequency and amplitude of modes are obtained. As seen from this table, Deep-Modal has very high efficiency for all the objects due to the universality of the network trained. Along with the increment of geometric complexity, the time cost of modal analysis increases drastically. In contrast, Deep-Modal only takes around 0.01s on a GPU (GeForce GTX 1080 Ti). Modal synthesis is run on a CPU (Intel i7-8700) because the sparse generalized eigenvalue algorithm is generally hard to accelerate using GPU. A recent work reported only 6.4× speedup can be obtained using two high-end GPUs (NVIDIA Tesla P100s) [5], which

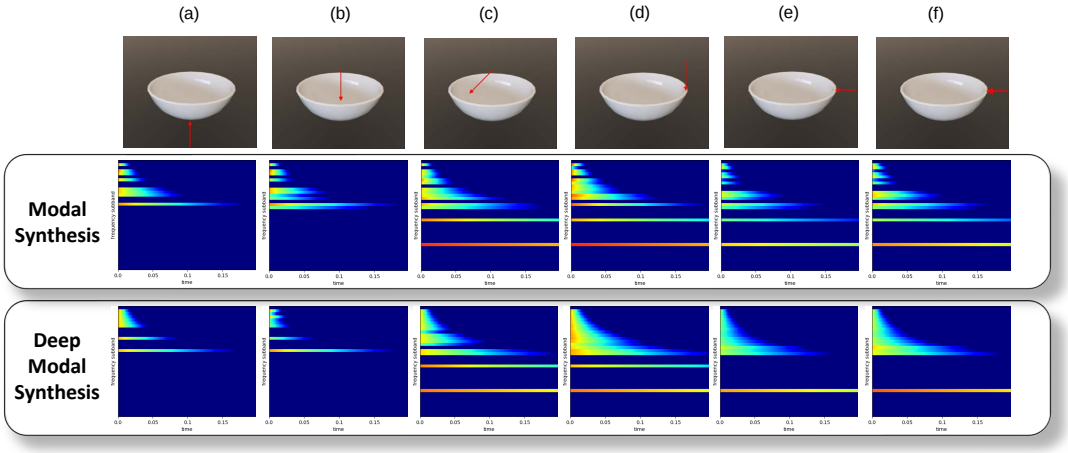

Figure 7: Sound synthesis of a ceramic bowl at various excited positions. We use spectrogram to visualize the similarity of our method with the ground truth (Modal Sound). (f) represents the same contact position as (e) with larger force.

shows that GPU acceleration is less significant and not applicable to modal analysis.

Table 2: Time cost of modal synthesis vs. Deep-Modal on different objects. We highlight the high performance of our method for objects with different geometric complexities.

Object	Voxel Num	Modal Synthesis	Ours
bowl	2366	2.0s	**~0.01s**
cup	5965	13.2s	
dresser	7448	43.5s	
glass box	13985	119.8s	

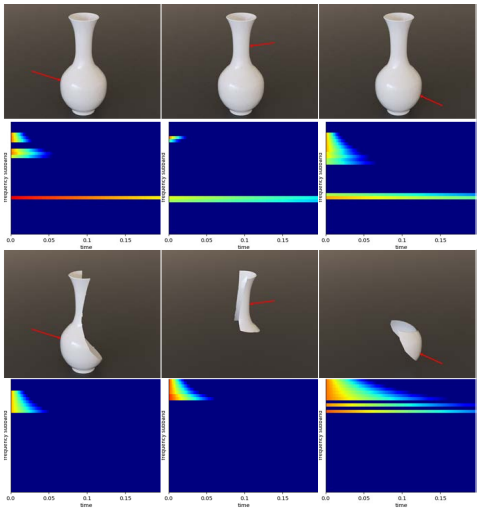

Figure 8: Sound synthesis for dynamically generated objects (a fractured scene) using our method.

5.2.4 Dynamic Object. We perform an extra experiment to synthesize the sound of newly generated objects after breaking or fracturing happens. We highlight the performance on complex scenarios with broken or fractured objects as shown in Figure 8, where new shapes are generated on the fly. Prior modal synthesis algorithms cannot handle such benchmarks for real-time sound synthesis. On the other hand, our approach can synthesize the audio effects in real-time. However, as fragment-like shapes have not been included in the training set, the accuracy of prediction should be improved with larger data sets for learning.

6 CONCLUSION AND FUTURE WORK

We propose Deep-Modal, which can perform approximate modal analysis in real-time. We designed our neural network to convert the shape features of the objects into audio features. We showed the higher fidelity and efficiency of our method when compared to the modal synthesis method in terms of spectrum and audio.

Our method still has limitations. Our input models must first be converted to hexahedral cells, which are not as accurate as tetrahedral cells. We treat the modes in the same sub-band as one mode and make the contact positions in the same voxel the same by approximation; this process may result in some loss of accuracy. In addition, the performance of our network also needs to be improved. Our work has two future directions. One direction is to improve the performance of Deep-Modal by increasing the quality and size of the dataset and optimizing the network. Another direction is to modify the sound representation and present a new network that can take acoustic transfer into account [1, 11, 16]. Finally, we also want to perform quantitative analysis of spectrograms' similarity using relevant distance (e.g. earth-mover's distance) and accumulated error over time for our future work, and a reasonable metric based on this measure for evaluating the quality of sound should be given.

7 ACKNOWLEDGEMENTS

This project was supported by the National Key R&D Program of China (No.2017YFB0203002, No.2017YFB1002700) and National Natural Science Foundation of China (No.61632003, No.61631001).

REFERENCES

[1] Lakulish Antani, Anish Chandak, Micah Taylor, and Dinesh Manocha. 2011. Direct-to-indirect acoustic radiance transfer. *IEEE Transactions on Visualization and Computer Graphics* 18, 2 (2011), 261–269.

[2] Marc Aretz. 2012. *Combined wave and ray based room acoustic simulations of small rooms*. Vol. 12. Logos Verlag Berlin GmbH.

[3] Nicolas Bonneel, George Drettakis, Nicolas Tsingos, Isabelle Viaud-Delmon, and Doug James. 2008. Fast modal sounds with scalable frequency-domain synthesis. In *ACM SIGGRAPH 2008 papers*. 1–9.

[4] Jeffrey N Chadwick, Changxi Zheng, and Doug L James. 2012. Precomputed acceleration noise for improved rigid-body sound. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–9.

[5] Adam Dziekonski and Michal Mrozowski. 2018. A GPU solver for sparse generalized eigenvalue problems with symmetric complex-valued matrices obtained using higher-order FEM. *IEEE Access* 6 (2018), 69826–69834.

[6] Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T Freeman, and Michael Rubinstein. 2018. Looking to listen at the cocktail party: a speaker-independent audio-visual model for speech separation. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–11.

[7] Ziqi Fan, Vibhav Vineet, Hannes Gamper, and Nikunj Raghuvanshi. 2020. Fast acoustic scattering using convolutional neural networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 171–175.

[8] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. 2018. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 6546–6555.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[10] DL James, TR Langlois, R Mehra, and C Zheng. 2016. Physically Based Sound for Computer Animation and Virtual Environments. ACM SIGGRAPH 2016 Course.

[11] Doug L James, Jernej Barbič, and Dinesh K Pai. 2006. Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources. *ACM Transactions on Graphics (TOG)* 25, 3 (2006), 987–995.

[12] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida. 2018. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5010–5019.

[13] Ignacio Llamas. 2007. Real-time voxelization of triangle meshes on the GPU. In *SIGGRAPH Sketches*. 18.

[14] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3431–3440.

[15] Daniel Maturana and Sebastian Scherer. 2015. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 922–928.

[16] Ravish Mehra, Nikunj Raghuvanshi, Lakulish Antani, Anish Chandak, Sean Curtis, and Dinesh Manocha. 2013. Wave-based sound propagation in large open scenes using an equivalent source formulation. *ACM Transactions on Graphics (TOG)* 32, 2 (2013), 1–13.

[17] Alejandro Newell, Kaiyu Yang, and Jia Deng. 2016. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*. Springer, 483–499.

[18] James F O'Brien, Chen Shen, and Christine M Gatchalian. 2002. Synthesizing sounds from rigid-body simulations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 175–181.

[19] Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H Adelson, and William T Freeman. 2016. Visually indicated sounds. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2405–2413.

[20] R Falcon Perez. 2018. *Machine-learning-based estimation of room acoustic parameters*. Ph.D. Dissertation. Aalto University, School of Electrical Engineering.

[21] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 652–660.

[22] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. 2016. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5648–5656.

[23] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*. 5099–5108.

[24] Nikunj Raghuvanshi and Ming C Lin. 2006. Interactive sound synthesis for large scale environments. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games*. 101–108.

[25] Nikunj Raghuvanshi, Rahul Narain, and Ming C Lin. 2009. Efficient and accurate sound propagation using adaptive rectangular decomposition. *IEEE Transactions on Visualization and Computer Graphics* 15, 5 (2009), 789–801.

[26] Zhimin Ren, Hengchin Yeh, and Ming C Lin. 2013. Example-guided physically based modal sound synthesis. *ACM Transactions on Graphics (TOG)* 32, 1 (2013), 1–16.

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.

[28] Atul Rungta, Carl Schissler, Ravish Mehra, Chris Malloy, Ming Lin, and Dinesh Manocha. 2016. SynCoPation: Interactive synthesis-coupled sound propagation. *IEEE transactions on visualization and computer graphics* 22, 4 (2016), 1346–1355.

[29] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4779–4783.

[30] Auston Sterling and Ming C Lin. 2016. Interactive modal sound synthesis using generalized proportional damping. In *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. 79–86.

[31] Auston Sterling, Nicholas Rewkowski, Roberta L Klatzky, and Ming C Lin. 2019. Audio-material reconstruction for virtualized reality using a probabilistic damping model. *IEEE transactions on visualization and computer graphics* 25, 5 (2019), 1855–1864.

[32] Auston Sterling, Justin Wilson, Sam Lowe, and Ming C. Lin. 2018. ISNN: Impact Sound Neural Network for Audio-Visual Object Classification. In *Computer Vision – ECCV 2018*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer International Publishing, Cham, 578–595.

[33] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. 2015. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*. 945–953.

[34] Zhenyu Tang, Nicholas J Bryan, Dingzeyu Li, Timothy R Langlois, and Dinesh Manocha. 2020. Scene-Aware Audio Rendering via Deep Acoustic Analysis. *IEEE Transactions on Visualization and Computer Graphics* 26, 5 (2020), 1991–2001.

[35] Zhenyu Tang, John D. Kanu, Kevin Hogan, and Dinesh Manocha. 2019. Regression and Classification for Direction-of-Arrival Estimation with Convolutional Recurrent Neural Networks. In *Proc. Interspeech 2019*. 654–658.

[36] Kees van de Doel and Dinesh K Pai. 1996. Synthesis of shape dependent sounds with physical modeling. Georgia Institute of Technology.

[37] Kees Van Den Doel, Paul G Kry, and Dinesh K Pai. 2001. FoleyAutomatic: physically-based sound effects for interactive simulation and animation. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 537–544.

[38] Kees Van den Doel and Dinesh K Pai. 1998. The sounds of physical shapes. *Presence* 7, 4 (1998), 382–395.

[39] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. [n.d.]. WaveNet: A Generative Model for Raw Audio. In *9th ISCA Speech Synthesis Workshop*. 125–125.

[40] Jui-Hsien Wang and Doug L James. 2019. KleinPAT: optimal mode conflation for time-domain precomputation of acoustic transfer. *ACM Trans. Graph.* 38, 4 (2019), 122–1.

[41] Jui-Hsien Wang, Ante Qu, Timothy R Langlois, and Doug L James. 2018. Toward wave-based sound synthesis for computer animation. *ACM Trans. Graph.* 37, 4 (2018), 109–1.

[42] Yuxuan Wang, R.J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif Saurous. 2017. Tacotron: Towards End-to-End Speech Synthesis. In *Proc. Interspeech 2017*. 4006–4010.

[43] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. 2019. Dynamic graph cnn for learning on point clouds. *ACM Transactions On Graphics (TOG)* 38, 5 (2019), 1–12.

[44] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3D shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1912–1920.

[45] Bin Xiao, Haiping Wu, and Yichen Wei. 2018. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*. 466–481.

[46] Qiong Zhang, Lu Ye, and Zhigeng Pan. 2005. Physically-based sound synthesis on GPUs. In *International Conference on Entertainment Computing*. Springer, 328–333.

[47] Zhoutong Zhang, Qiujia Li, Zhengjia Huang, Jiajun Wu, Josh Tenenbaum, and Bill Freeman. 2017. Shape and material from sound. In *Advances in Neural Information Processing Systems*. 1278–1288.

[48] Changxi Zheng and Doug L James. 2010. Rigid-body fracture sound with precomputed soundbanks. In *ACM SIGGRAPH 2010 papers*. 1–13.

[49] Changxi Zheng and Doug L James. 2011. Toward high-quality modal contact sound. In *ACM SIGGRAPH 2011 papers*. 1–12.