Acoustic Voxels: Computational Optimization of Modular Acoustic Filters

Dingzeyu Li[*] David Levin[†] Wojciech Matusik[‡] Changxi Zheng[*]

[*]Columbia University [†]Disney Research [‡]MIT CSAIL

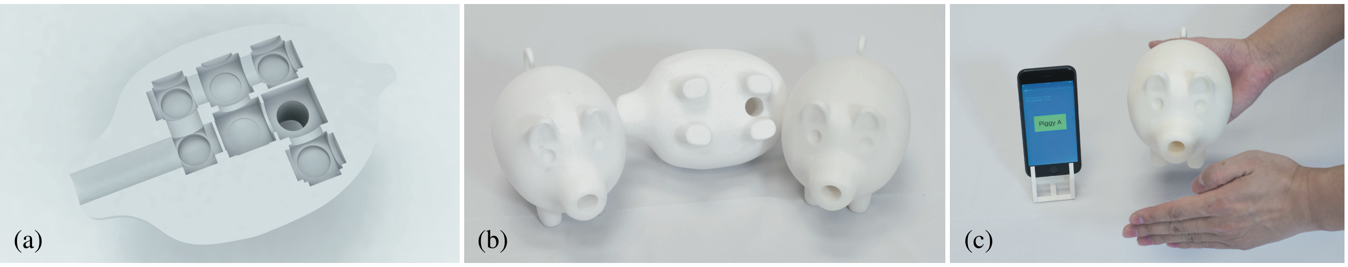


Figure 1: Acoustic Tagging. *By optimizing the structure of primitives* **(a)***, we control the acoustic response of an object when it is tapped* **(c)** *and thereby tag the object acoustically. Given three objects with identical shapes* **(b)***, we can use a smartphone to read the acoustic tags in realtime, by recording and analyzing the tapping sound, and thereby identify each object.*

Abstract

Acoustic filters have a wide range of applications, yet customizing them with desired properties is difficult. Motivated by recent progress in additive manufacturing that allows for fast prototyping of complex shapes, we present a computational approach that automates the design of acoustic filters with complex geometries. In our approach, we construct an acoustic filter comprised of a set of parameterized shape primitives, whose transmission matrices can be precomputed. Using an efficient method of simulating the transmission matrix of an assembly built from these underlying primitives, our method is able to optimize both the arrangement and the parameters of the acoustic shape primitives in order to satisfy target acoustic properties of the filter. We validate our results against industrial laboratory measurements and high-quality off-line simulations. We demonstrate that our method enables a wide range of applications including muffler design, musical wind instrument prototyping, and encoding imperceptible acoustic information into everyday objects.

Keywords: Shape optimization, fabrication, computational design, acoustic impedance, transmission loss

1 Introduction

Acoustic filters have numerous important applications, whether to produce a desired sound pitch or to attenuate undesired noise. These applications, ranging from wind instruments to mufflers and hearing aids, all rely on the same fundamental physical principle: when sound waves pass through a cavity, part of the waves reflect back and forth, effectively boosting or suppressing certain acoustic frequencies. In this process, the filtered frequencies are largely affected by the *shape* of the cavity.

However, for all but the simplest cavity shapes, the influence of the shape on the filtered frequency bands is complicated and unintuitive. Thus, the current process for improving the quality of acoustic filters requires many trial-and-error iterations over the shape. Furthermore, the design space is often limited to simple geometries such as pipes (e.g., for making flutes, trumpets, and industrial mufflers) since the acoustic behavior of only these simple shapes can be easily characterized. Current computational design tools support only these simple primitives and even then the design process requires strong expertise in this domain.

Meanwhile, recent advances in additive manufacturing have significantly facilitated rapid manufacturing of complex geometries. This trend opens up new possibilities for expanding the design space of acoustic filters, thus motivating the development of corresponding computational methods that can efficiently simulate and optimize the shape of the cavity in order to achieve desired acoustic filtering effects. In light of this, the goal of our work is to expand the range of acoustic filter design by employing complex cavity shapes computationally optimized and then physically realized using additive manufacturing.

We propose *Acoustic Voxels*, a computational method that assembles basic shape primitives into a complex geometry, one that produces the desired acoustic filtering. In particular, we consider a simple type of shape primitive, a hollow cube with circular holes on some of its six faces (Figure 4). We show that these primitives, albeit simple individually, offer a large design space for acoustic filters when *modularly* joined at their faces into a complex assembly. This modular scheme also permits fast and accurate estimation of the acoustic performance of a given assembly, thereby allowing automatically optimizing its structure to achieve target acoustic filtering properties while satisfying geometric constraints of inlet/outlet positions and overall shapes.

Our approach starts with precomputing the acoustic transmission for our parameterized shape primitives. At runtime, given an arbitrary assembly of these primitive filters, our method estimate its acoustic transmission, predicting the boosted and suppressed frequency regions. This, in turn, enables us to derive a formula to compute the gradient of the acoustic transfer with respect to shape parameters, and further develop an efficient combinatorial and continuous optimization algorithm to design desired filter structures. Our method combines a stochastic optimization method for computing the topology of the assembly (i.e., the way of arranging and connecting the primitives) with a gradient-based quasi-Newton method for computing the geometric parameters of each primitive shape in the assembly. We validate our method by running finite-element offline acoustic simulation and industrial laboratory tests performed by acoustic engineering professionals (§6).

Our proposed approach automates the design of acoustic filters. This simplified design process allows casual users to produce objects with custom acoustic properties. Our method also expands the range of acoustic filters that can be achieved, enabling exploration of many different applications. In addition to designing different types of noise attenuation components (e.g., mufflers), our method can customize musical instruments with non-conventional shapes. Furthermore, we can embed imperceptible acoustic information into the fabricated objects, and thus opens up new types of interactions with fabricated objects, extending current visually based design into audiovisual design.

2 Related Work

Sound simulation. The computer graphics community has a long history of simulating sound propagation in a virtual environment [Stettner and Greenberg 1989; Takala and Hahn 1992], starting from the geometric acoustical methods [Funkhouser et al. 1998; Funkhouser et al. 1999; Tsingos et al. 2001] which are fast but less accurate at low frequencies, and evolving to the wave-based methods [James et al. 2006; Raghuvanshi et al. 2010; Mehra et al. 2013; Raghuvanshi and Snyder 2014] to improve sound quality. The goal of these work is to add realistic wave scattering and room acoustic effects. These approaches have proven successful in many virtual environment applications, but not for fabricating acoustic structures. Moreover, the geometric scale in those simulations is typically meters or tens of meters, whereas we are interested in the sound propagation in small cavities at the centimeter scale.

Recently, Allen and Raghuvanshi [2015] proposed an interactive method for simulating wave propagation in wind instruments, modeled in 2D. This method produces realistic sound effects in realtime, but is unclear how to apply it for solving our inverse problem. Aside from requiring a physics-based 3D simulation and high accuracy for predicting fabricated results, our method needs a well-defined relationship between the sound transmission and the boundary geometry to formulate a tractable inverse problem.

Acoustic inverse problem. Acoustic inverse problems have intrigued scientists for decades, starting from Kac's famous question: "can one hear the shape of a drum?"[Kac 1966]. While Kac's question is about the vibrational patterns of a shape, similar questions that infer shapes from sound propagation and scattering patterns have been actively studied [Angell et al. 1997; Feijóo et al. 2004]. Monks et al. [2000] optimized room acoustics motivated by the applications in architectural design. Recently, Dokmanić et al. [2013] showed an algorithm for computing a convex polyhedral room shape using acoustic response recorded at multiple microphones. We also address an acoustic inverse problem, but from a different perspective. Our input is the acoustic response (i.e., impedance or transmission loss) measured at a pair of locations (i.e., between the inlet and the outlet), and our goal is not to reconstruct existing shapes but to construct new structures.

Transmission Line Matrix. Based on Huygens' model of wave propagation and the analogy between wave propagation and transmission lines, the Transmission Line Method has been widely used for computing electromagnetic waves [Caloz and Itoh 2005; Christopoulos 2006] and acoustics [Munjal 2014]. It first discretizes the computational domain into interconnected nodes. On the connecting interface, field information is propagated and coupled between adjacent nodes. By breaking down the whole domain into basic nodes, the computational performance can be significantly improved. Our method shares the similar idea, but a key difference lies in the new optimization framework. Our method optimizes the configuration of the nodes both geometrically and topologically, aiming to realize the desired acoustic filtering properties.

Muffler design. Noise attenuation is an important topic in many engineering fields. There has been a well established theory for modeling noise reduction in a cavity structure [Ingard 2009; Munjal 2014], and numerous approaches for improving a standard muffler have been developed with sub-chamber structures [Selamet et al. 2003], varying inlet and outlet sizes [De Lima et al. 2011], or perforated liners [Chiu 2010; Munjal 2014]. However, the optimization for desired target performance is not straightforward. Traditionally, mufflers are often analyzed using finite element methods and then used in a sensitivity analysis to compute the derivatives of the muffler metric with respect to shape parameters. In general, this is a computationally expensive process.

The application of our method for muffler design takes a different approach, namely tiling simple resonator shapes, without choosing a specific parametric shape *a priori*. Meanwhile, precomputed filtering properties of primitive resonators sidestep the expensive finite-element solves during the optimization and thus allows us to optimize for complex structures.

Computational design of music instruments. Our method can be applied to customize wind instruments, although that is not a primary goal of our work. Related to this aspect, existing work has explored the optimization of the bore shapes for brasswind instruments [Kausel 2001; Noreland et al. 2010]. Similar to our optimization target, Braden et al. [2009] use the input impedance of the instrument in the objective function to optimize bore shapes. These methods typically focus on a specific family of shapes and thereby formulate a continuous optimization problem. Our method, in contrast, aims to create acoustic filters using an arbitrary shape for a range of applications beyond wind instruments. More recently, Zoran [2011] has demonstrated the use of 3D printers for creating plausible wood instruments and for exploring new designs without any numerical optimization.

In computer graphics, Umetani et al. [2011] have develop the first interactive tool for designing metallophones. The tool aims for interactivity but not for solving the inverse problem. Recently, Bharaj et al. [2015] have explored the inverse computational design of metallophones and have proposed a stochastic optimization method for this purpose. Unlike ours, both approaches focus on the modal vibrational sounds from solid vibrations but not the sound propagation inside a chamber.

Microstructure design. Recently in computer graphics, there has been a variety of work on designing macroscopic mechanical material properties through controlling their microscopic structures, based on the inverse homogenization theory [Sigmund 1994]. Along this line of research, existing work has used a data-driven approach to control nonlinear elasticity [Bickel et al. 2010] with multi-material 3D printing, while others tile precomputed structural patterns [Panetta et al. 2015; Schumacher et al. 2015] to obtain user-specified elastic properties. While we also combine small-scale primitives, In order to affect sound waves, the geometric size of our primitives is of a few centimeters, much larger than the microstructure scales in these approaches. Furthermore, rather than the elastic behaviors of microstructures, we focus on the sound propagation through the primitives.

Acoustic in HCI. Recent development in passive acoustic sensing inspired new HCI applications, such as the recent tangible input devices by analyzing the sound produced by a comb-like structure [Savage et al. 2015]. More relevant to our method, Laput et al. [2015] proposed Acoustruments to recover information from audio signals recorded through ducts. None of these previous works considers the inverse problem of acoustic optimization. Our method

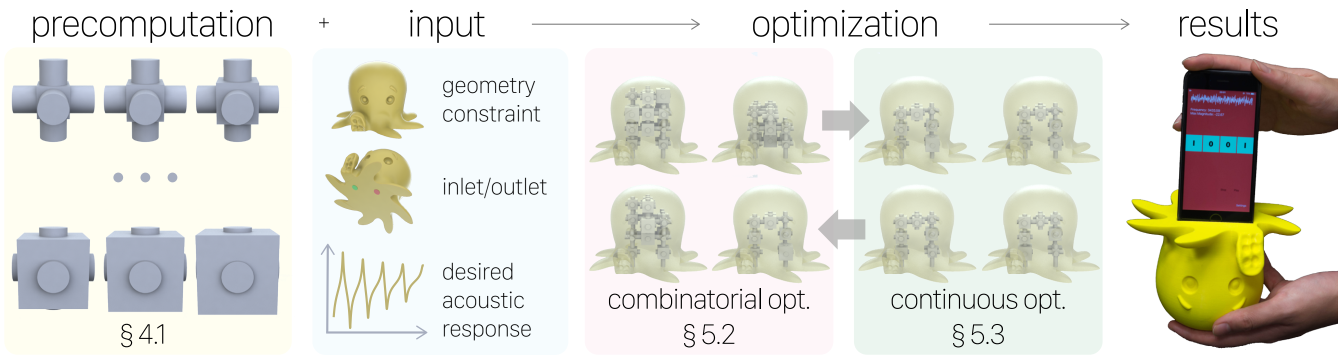


Figure 2: Overview. *Our method exploits precomputed transmission matrices of the primitives and uses a combinatorial and continuous optimization to construct the assembly of filters. Please refer to §3.1 for an outline of each step.*

complements to those work and offers a computational tool to develop new HCI applications, as we will demonstrate in §6.4.

Contributions. Compared to previous work, our method has the following contributions: (i) We propose to construct acoustic filters using primitive resonators. (ii) With modular assemblies, we develop a numerical optimization method to construct desired acoustic filters while sidestepping expensive finite-element solves. (iii) We demonstrate the use of our primitives and optimization method in the context of different applications including a new application that embeds acoustic signatures into 3D printed objects.

3 Background and Overview

We start by briefly reviewing the theory of acoustic filters and refer to the textbooks [Ingard 2009; Munjal 2014] for more details. A typical acoustic filter has a cavity structure connecting an inlet and an outlet— trumpets and motorcycle mufflers are classic examples. When sound waves enter into the inlet, travel through the cavity, and leave from the outlet, their frequency components are altered. In most applications, the physical size of a filter ranges from centimeters to tens of centimeters and their operating frequencies are up to thousands of Hz. To evaluate the performance of acoustic filters, the following two quantities are often used (Figure 3):

- **Input impedance.** Consider a steady-state sound transmission through a filter. In the frequency domain, the sound pressure and acoustic velocity at a location $\boldsymbol{x}$ are denoted as $p(\boldsymbol{x},\omega)$ and $v(\boldsymbol{x},\omega)$, respectively; both are complex values. The *acoustic impedance*, defined as $Z(\boldsymbol{x},\omega) = \frac{p(\boldsymbol{x},\omega)}{v(\boldsymbol{x},\omega)}$, indicates how much sound pressure is generated by a given air vibration of frequency ω at position $\boldsymbol{x}$. Particularly, we are interested in the impedance value at the inlet $\boldsymbol{x}_i$, $Z_{\text{IN}}(\omega) = Z(\boldsymbol{x}_i,\omega)$, called *input impedance* (Fig. 3). $Z_{\text{IN}}(\omega)$ usually varies strongly with respect to the frequency and has multiple local minima and maxima, which correspond to the sound frequencies that are the easiest and the most difficult to transmit through the filter. For example, the playing frequencies of a trumpet are very close to the local maxima of its input impedance.

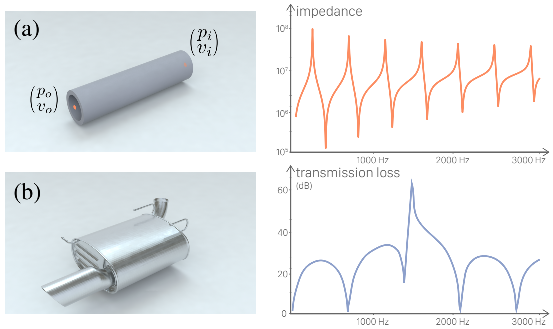


Figure 3: Acoustic filters examples. *(a) a duct as part of a wood instrument is measured using the input acoustic impedance; (b) Mufflers are often evaluated using transmission loss.*

- **Transmission loss.** To design acoustic filters for noise reduction, a widely used measure is the *transmission loss* [Munjal 2014], defined as the ratio, expressed in decibels (dB), of the acoustic power incident to the muffler to the power transmitted downstream into the environment. Concretely, if the inlet and the outlet of an acoustic filter are sufficiently small, its transmission loss is described as:

$$L_{\text{TL}}(\omega) = 10\log_{10}\left|\frac{S_{\text{i}}p_{\text{i}+}^2(\omega)}{S_{\text{o}}p_{\text{o}}^2(\omega)}\right|,$$

  where S_{i} and S_{o} are the cross-sectional area of the inlet and the outlet, respectively; $p_{\text{o}}(\omega)$ is the frequency-domain acoustic pressure of the transmitted sound wave at the outlet, away from the filter; and $p_{\text{i}+}(\omega)$ is the acoustic pressure of the incident wave at the inlet, also in the frequency domain. In short, $L_{\text{TL}}(\omega)$ measures how much the sound wave of frequency ω gets attenuated when passing through the filter.

Depending on specific applications, our goal is to optimize the internal structure of a filter in order to obtain target input impedance or transmission loss in a frequency range. In general, accurately predicting these quantities requires solving the acoustic wave equation or, in the frequency domain, the Helmholtz equation [Pierce et al. 1991; Allen and Raghuvanshi 2015]. Either approach is computationally expensive, especially for complex filter structures. Notably, the relationship between the geometry of the filter and the resulting impedance or transmission loss function is rather complex, obstructing us from formulating a well-defined optimization problem of this geometry. Therefore, we take a different approach by leveraging the concept of the transmission matrix.

Transmission matrix. If both the inlet and the outlet have a small cross-section, much smaller than the wavelength of the operated sound waves, one can reasonably assume that the acoustic pressure and velocity are both distributed uniformly over the cross-section [Ingard 2009; Rienstra and Hirschberg 2003]. Let $(p_{\text{i}}(\omega), v_{\text{i}}(\omega))$ and $(p_{\text{o}}(\omega), v_{\text{o}}(\omega))$ denote the complex-valued acoustic pressure and velocity in frequency domain, at the cross-sections of the inlet and the outlet, respectively (Figure 3-a). Their relationship can be approximated linearly,

$$\begin{pmatrix} p_{\text{o}}(\omega) \\ p_{\text{i}}(\omega) \end{pmatrix} = \begin{pmatrix} T_{11}^{\omega} & T_{12}^{\omega} \\ T_{21}^{\omega} & T_{22}^{\omega} \end{pmatrix} \begin{pmatrix} v_{\text{o}}(\omega) \\ v_{\text{i}}(\omega) \end{pmatrix}, \tag{1}$$

where T^ω_{ij} is i-th row and j-th column in the complex-valued *transmission matrix* at frequency ω. In this paper, we also denote this matrix as $\mathsf{T}(\omega)$ to emphasize its frequency dependence. Transmission matrices have been widely used in industrial muffler design [Ingard 2009], as it relates to the input impedance and transmission loss through simple formulas:

$$Z_{\text{IN}}(\omega)=\frac{T^\omega_{11}+T^\omega_{12}T^\omega_{21}-T^\omega_{11}T^\omega_{22}}{1-T^\omega_{22}} \text{ and} \tag{2}$$

$$L_{\text{TL}}(\omega)=20\log_{10}\left(\frac{1}{2}\left|\frac{T^\omega_{11}}{T^\omega_{21}}+\frac{T^\omega_{12}}{\rho c}-\frac{T^\omega_{11}T^\omega_{22}}{\rho c T^\omega_{21}}+\frac{\rho c}{T^\omega_{21}}-\frac{T^\omega_{22}}{T^\omega_{21}}\right|\right) \tag{3}$$

where ρ is the air density and c is the sound speed.

Challenges. Unfortunately, computing the transmission matrix of a filter structure is expensive. For each frequency ω, the standard approach first samples two sets of pressures, $(p_{\text{i}1}, p_{\text{o}1})$ and $(p_{\text{i}2}, p_{\text{o}2})$, at the inlet and the outlet. Each set of pressures, together with the zero-normal-velocity condition on the solid boundary of the filter, forms a complete boundary condition that can be used to solve the Helmholtz equation and uniquely determine the acoustic velocity $(v_{\text{i}1}, v_{\text{o}1})$ (and $(v_{\text{i}2}, v_{\text{o}2})$) at the inlet and outlet (see Appendix A for more details). Then, the transmission matrix can be computed by solving a 2×2 linear system,

$$\begin{pmatrix} p_{\text{o}1}(\omega) & p_{\text{o}2}(\omega) \\ p_{\text{i}1}(\omega) & p_{\text{i}2}(\omega) \end{pmatrix} = \begin{pmatrix} T^\omega_{11} & T^\omega_{12} \\ T^\omega_{21} & T^\omega_{22} \end{pmatrix} \begin{pmatrix} v_{\text{o}1}(\omega) & v_{\text{o}2}(\omega) \\ v_{\text{i}1}(\omega) & v_{\text{i}2}(\omega) \end{pmatrix}. \tag{4}$$

This process needs to solve the Helmholtz equation twice for each frequency ω. In addition, while it is straightforward to compute impedance and transmission loss using the transmission matrix, it remains hard, if not impossible, to compute the gradient of the transmission matrix with respect to the geometric parameters of the filter—this gradient is needed for optimizing the cavity shape of the filter (§5). Our approach addresses all these challenges.

3.1 Method Overview

Our method automatically constructs the internal structure of an acoustic filter that connects an inlet and an outlet of a 3D volume (Figure 4-right). We aim to control its input impedance or the transmission loss function as specified by the user.

System input and output. Concretely, out method takes as input three components, (i) a 3D volume in which the acoustic filter is placed, (ii) the positions of the inlet and the outlet, specified on the surface of the 3D volume, and (iii) the frequency locations to be boosted or suppressed. It then outputs the transmission geometry that fits into the 3D volume and that can be fabricated to produce the desired filtering effects (Figure 2).

To this end, we propose a primitive acoustic resonator, a family of simple hollow shapes serving as building blocks to assemble a complex acoustic filter. These primitives allow us to precompute their transmission matrices, which in turn enable a fast runtime algorithm to compute the acoustic impedance and transmission loss of any filters made from an assembly of the primitive resonators (§4). Leveraging the fast computation of transmission matrices, we further address the optimization of the inverse problem (§5), one that finds an assembly of the primitive shapes to achieve a target acoustic input impedance or transmission loss. To this effect, we formulate a combinatorial and continuous optimization problem, combinatorial in the sense of how to connect primitive shapes, and continuous in the sense of determining geometric parameters of the primitives. To solve it, we propose a hybrid method that interleaves a stochastic optimization, namely the Sequential Monte Carlo method, with a gradient-based quasi-Newton scheme.

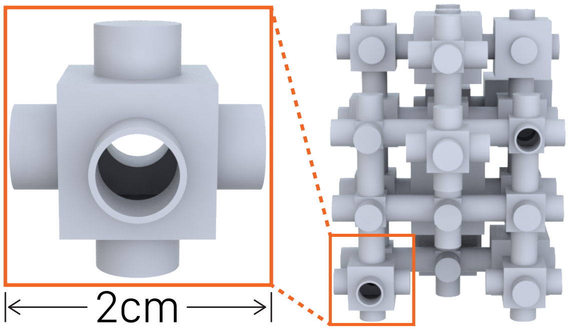


Figure 4: Modular filter. *Our primitive resonator is a single shape bounded by a 2cm×2cm×2cm cube (left). A combination of the primitives with varying shape parameters can form complex structure that connects an inlet to an outlet.*

4 Modular Acoustic Filter

4.1 Primitive Resonator

We propose to use a simple shape as our primitive resonator—a hollow cube with extruded cylinders on its six faces (Figure 4-left). All the cylindrical extrusions have the same radius and length and therefore the bounding boxes of all primitives stay the same. They can be composed together at their faces by connecting an inlet and an outlet to form a complex structure (Figure 4-right). Furthermore, the size of each hollow cube can change, providing one degree of freedom per element as control variables to influence acoustic filtering properties, in addition to the connectivity of the resonators.

Rationale. Using the simple primitives offers many advantages: (i) They can fill the interior volume of virtually any shape, as long as they are sufficiently small. This enables us to construct acoustic filters subject to various shape constraints. (ii) Computing the transmission matrix of any assembly becomes fast and accurate. (iii) With a hollow cube of a variable size, the primitive is in a one-dimensional shape space, which can be easily sample. For each sample, we precompute its transmission matrices and interpolate between neighboring transmission matrices. When composed into an assembly, these primitives offer a large number of degrees of freedom for controlling acoustic filtering properties. This idea is also similar to the concept of 3D symmetric condensed nodes for the computation of electromagnetic fields [Christopoulos 2006].

In this section, we describe how we compute the transmission matrix of an arbitrary assembly of the primitive resonators. Ultimately, our goal is to compute both the *topology* of the primitive assembly and the *geometric parameters* (i.e., the cube size of each element) for a desired input impedance or transmission loss.

Multi-port transmission matrix. We start by extending the concept of transmission matrix in Eq. (1) into a six-port transmission matrix. Since the radii of the six open ports of a primitive shape are small, it remains valid to assume that the frequency-domain acoustic pressure $p_{\text{i}}(\omega)$ and velocity $v_{\text{i}}(\omega)$ (i=1...6) are uniformly distributed over the cross sections at each port. Then a linear relationship similar to Eq. (1) holds:

$$\begin{pmatrix} p_1(\omega) \\ \vdots \\ p_6(\omega) \end{pmatrix} = \begin{pmatrix} T^\omega_{11} & \dots & T^\omega_{16} \\ \vdots & \ddots & \vdots \\ T^\omega_{61} & \dots & T^\omega_{66} \end{pmatrix} \begin{pmatrix} v_1(\omega) \\ \vdots \\ v_6(\omega) \end{pmatrix}. \tag{5}$$

In a way similar to Eq. (4), we compute for a given frequency ω the six-port transmission matrix T by sampling six different sets of pressures $\{p_i, i = 1...6\}$. Each set of pressures establishes the

(Dirichlet) boundary condition that uniquely solves the Helmholtz equation for sound propagation in the primitive resonator. After the six Helmholtz solves, it produces six corresponding acoustic velocities $\{v_i, i = 1...6\}$, which, together with $\{p_i\}$, can be substituted into Eq. (5) and uniquely determine the matrix T. Appendix A details this computational process.

Precomputation. Computing a transmission matrix T requires solving the Helmholtz equation on the cavity shape of the primitive six times. Moreover, it depends on not only the frequency but also the shape parameter, the cube size. Therefore, we sample a set of frequency values and cube sizes, precompute the T matrices, and store them in a database. We also note that the precomputation step can be accelerated using the recent asymptotic frequency sweeping method [Li et al. 2015]. With these precomputed six-port transmission matrices, we are able to interpolate the matrix of a primitive resonator of any frequency and cube size in the sampled range. This interpolation will be used later in the optimization step (§5) to compute optimal topology and geometry of the primitive assembly for a target acoustic filtering property.

4.2 Transmission Matrix of Resonator Assembly

Now we compute the transmission matrix of a resonator assembly, which consists of primitive resonators (the size of each resonator is specified). Each of the six ports of a resonator is either joined with a port of another resonator or closed with a solid wall. These ports are connected (possibly through multiple paths) from an inlet to an outlet. Our goal here is to compute the frequency-dependent 2×2 transmission matrix that relates the acoustic pressure and velocity at the outlet to those at the inlet, as described in Eq. (1).

We start with some notation. Consider an assembly composed of N primitive resonators. We use j to index the primitives and k to index the six ports of each primitive. Let $p_k^j(\omega)$ and $v_k^j(\omega)$ denote respectively the frequency-domain acoustic pressure and velocity at the k-th port of the j-th primitive. From the precomputation, for each primitive resonator j we also have a six-port (6×6) transmission matrix $\mathsf{T}_j(\omega)$ that relates the pressures $p_k^j(\omega)$ with velocities $v_k^j(\omega), k = 1...6$ at its six ports.

Similarly to the method used in Eq. (4), we sample two sets of pressures, $(\bar{p}_{\text{i}1}, \bar{p}_{\text{o}1})$ and $(\bar{p}_{\text{i}2}, \bar{p}_{\text{o}2})$, at the inlet and the outlet. We seek a fast method to compute the corresponding acoustic velocity, $(\bar{v}_{\text{i}1}, \bar{v}_{\text{o}1})$ and $(\bar{v}_{\text{i}2}, \bar{v}_{\text{o}2})$, without solving the expensive Helmholtz equations. We observe that we can construct a sparse linear system (visualized in Figure 5),

$$\mathsf{A}(\omega)\boldsymbol{x}(\omega) = \boldsymbol{b}(\omega), \tag{6}$$

to solve for the pressures $p_k^j(\omega)$ and velocities $v_k^j(\omega)$ of all ports ($j = 1...N, k = 1...6$). Here, $\boldsymbol{x}$ has $12N$ elements, stacking all the pressures and velocities of frequency ω at all ports. Every resonator contributes a linear relationship (5), resulting in a 6 linear equations which appears as a 6×12 submatrix (orange blocks in Figure 5). All the resonators together form a $6N \times 12N$ sub-block matrix. In addition, for the two ports that connect to the inlet and outlet, the pressures are the sampled values (i.e., the two green rows in Figure 5); at the closed ports, the velocities vanish; at every pair of connected ports, their pressures need to match and their velocities need to be additively inverse (e.g., the blue rows in Figure 5), as the sound waves flow along the same direction. All these constraints result in another $6N$ linear equations. Putting together these equations yield a full-rank sparse and $12N \times 12N$ linear system.

We also note that the matrix A depends on the cube sizes of the primitive resonators, as it is assembled using their transmission matrices T_i, but $\boldsymbol{b}$ is a constant. Later when optimizing the cube sizes, we will compute the derivative of A with respect to each cube size.

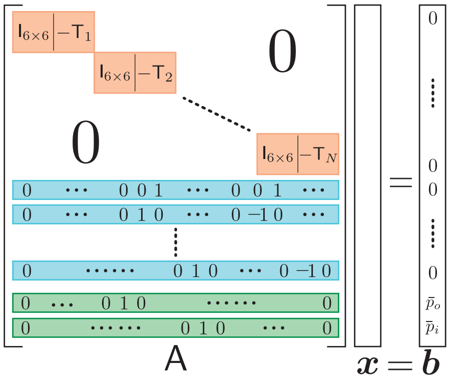

$\mathsf{I}_{6\times6}$ | $-\mathsf{T}_1$
$\mathsf{I}_{6\times6}$ | $-\mathsf{T}_2$
$\mathsf{I}_{6\times6}$ | $-\mathsf{T}_N$
0
0
0 ··· 0 0 1 ··· 0 0 1 ···
0 ··· 0 1 0 ··· 0 −1 0 ···
0 ······ 0 1 0 ··· 0 −1 0
0 ··· 0 1 0 ······ 0
0 ······ 0 1 0 ··· 0
A
=
0
0
0
0
$\bar{p}_o$
$\bar{p}_i$
$\boldsymbol{x} = \boldsymbol{b}$

Figure 5: Linear solve of a filter assembly. *The top orange part refers to the transmission matrices related to each node in the assembly. The middle blue part specifies the connection information by mapping the velocity and pressure values. The bottom two green rows are the given boundary conditions at the inlet and the outlet.*

Computational efficiency. This process computes the transmission matrix at a frequency ω by solving the sparse linear system, $\mathsf{A}\boldsymbol{x} = \boldsymbol{b}$, twice. Both have the same A matrix, so it only needs to be factorized once. In addition, across all frequencies, the sparsity pattern of A stays the same. To exploit this invariant, we use the symbolic factorization (reordering) only once for the entire computation and update the numerical data for each frequency sample, all implemented using the Direct Sparse Solver provided in Intel MKL. As a result, the computation of transmission matrices for all frequency samples (nearly 1000 samples) typically finishes in a few seconds.

5 Optimization

We now focus on the inverse problem: computing a structure of a primitive assembly and the parameter of each primitive in the assembly in order to realize a desired acoustic filtering property. We formulate this problem as a combinatorial and continuous optimization (§5.1). To address both the combinatorial and the continuous aspects of the problem, our algorithm interleaves a stochastic optimization method with a quasi-Newton method (§5.2 and §5.3).

5.1 Problem Formulation

Optimization objective. Our optimization goal, the acoustic filtering property, depends on a specific application, whether it is a target impedance $Z_{\text{IN}}(\omega)$ (e.g., for wind instruments) or a target transmission loss $L_{\text{TL}}(\omega)$ (e.g., for engine mufflers) in a frequency range $[\omega_l, \omega_r]$. Both quantities can be computed from the transmission matrix $\mathsf{T}(\omega)$ of a given assembly using Eq. (2) and Eq. (3), respectively. Thus, we discretize the frequency range using a set of samples $\omega_i \in [\omega_l, \omega_r], i = 1...N_\omega$ and define a unified objective function in a least-squares form:

$$J = \sum_{i=1}^{N_\omega} \left(g(\mathsf{T}(\omega_i)) - \bar{g}_i\right)^2. \tag{7}$$

Here, $g(\mathsf{T}(\omega_i))$ is the acoustic filtering quantity depending on the transmission matrix at a sampled frequency ω_i. For instance, to control the input impedance, we use $g(\mathsf{T}(\omega_i)) = \log_{10}|Z_{\text{IN}}(\omega_i)|$; to control the transmission loss, we use $g(\mathsf{T}(\omega_i)) = L_{\text{TL}}(\omega_i)$. $\bar{g}_i$ is the target acoustic filtering quantity at the frequency ω_i. These values are user-controlled, e.g., by specifying a target curve in the frequency domain.

We note that while this objective function suits well for our applications (§6), our optimization method does not depend on this particular choice, as presented in the rest of this section.

Shape constraint. In many applications, filters are often embedded in a limited space. To account for this requirement, we allow the user to specify a 3D surface mesh to constrain the volume of the assembly in the optimization process. Before the optimization starts, we voxelize the 3D mesh into a lattice, where each grid cell represents a possible placement of a primitive resonator, and the grid connects an inlet and an outlet, both specified on the mesh boundary (see video). By construction, the resulting assembly of resonators are guaranteed to satisfy the shape constraint and connect the inlet and outlet.

Optimization variables. We have two types of optimization variables: (i) a string of binary bits $\boldsymbol{s}$ indicating the lattice grid connectivity and (ii) a vector $\boldsymbol{u}$ stacking the cube sizes of primitive resonators used in the assembly. We index each grid cell interface in the lattice. If two primitives are joined at an interface i, then the corresponding bit in $\boldsymbol{s}$ is set to one. If a face of the grid cell is not connected with its neighboring grid cell, the corresponding bit in $\boldsymbol{s}$ is set to zero and the resonator port on that face is closed with a solid boundary. As we will describe later, this bit string representation is particularly suitable for our stochastic sampling algorithm. With these optimization variables, we rewrite the acoustic filtering quantity $g(\mathsf{T}(\omega))$ in Eq. (7) as $g(\mathsf{T}(\boldsymbol{s}, \boldsymbol{u}, \omega_i))$ and explicitly write J as $J(\boldsymbol{s}, \boldsymbol{u})$ because the transmission matrix T depends on both the topology (described by $\boldsymbol{s}$) and the geometry (described by $\boldsymbol{u}$) of the primitive assembly.

Method rationale and overview. The optimization variables reflect the combinatorial and continuous nature of our problem. The problem of determining the placement and connectivity of the primitives in the lattice is combinatorial; and determining the cube sizes of each primitive is continuous. A typical method of solving a combinatorial optimization relies on a Monte Carlo method to sample in the parameter space and accept or reject samples probabilistically. The efficiency of this method critically depends on the performance of evaluating the objective function, as it often requires a large number of samples. From this perspective, our fast computation of the transmission matrix (§4.2), a necessary component for evaluating the objective function (7), lays out an important cornerstone for using a stochastic optimization algorithm. Meanwhile, if the connectivity is given, optimizing the cube sizes for each primitive is a continuous problem, for which a gradient-based method is more efficient.

We propose to use a stochastic optimization method to optimize the connectivity of the primitives. When evaluating the objective function of a sampled resonator structure (i.e., the $\boldsymbol{s}$), we compute the cube size for each primitive (i.e., the $\boldsymbol{u}$) using a gradient-based continuous optimization method that minimizes the objective function with the fixed resonator structure. This is because continuous optimization, leveraging gradient descent, is more efficient than stochastically sampling cube sizes. Effectively, our method is a hybrid that interleaves a Monte Carlo sampling with a quasi-Newton optimization scheme.

5.2 Combinatorial Optimization of Connectivity

To solve a combinatorial optimization problem, one simple and popular approach is to use simulated annealing [Kirkpatrick et al. 1983], a method that can be interpreted as a single sequence of Markov-Chain Monte Carlo (MCMC) sampling [Robert and Casella 2013]. One way of improving its efficiency is to use multiple sequences of MCMC sampling, for which an efficient method is Sequential Monte Carlo (SMC). In computer graphics, SMC has been applied for rendering, character control, and procedural modeling [Pegoraro et al. 2008; Hämäläinen et al. 2014; Ritchie et al. 2015]. In numerical optimization, SMC methods have been used for optimizing non-convex, non-differentiable, and high-dimensional objective functions [Mıguez et al. 2010]. In the following, we outline our modified SMC algorithm, followed by highlighting the components that are specifically tailored for our problem.

Algorithm 1 SMC for Resonator Assembly Optimization

```
 1: procedure MODIFIED–SMC($G_N$, $M$, threshold)
 2:     while true do
 3:         for each topology in $G_N$ do
 4:             $\hat{p}_n$ ←continuous_optimization($p_n$)          ▷ §5.3
 5:             compute the objective $J_i$ for $\hat{p}_n$
 6:         end for
 7:         if best objective > threshold then(end optimization)
 8:         weighted sample $M$ topologies based on the objectives
 9:         for each topology in $G_N$ do                            ▷ §5.2
10:             if selected then
11:                 perturb connectivity of the current graph
12:             else
13:                 resample a new random graph
14:                 MCMC step: probabilistically accept the sample
15:         end for
```

Modified SMC algorithm. As outlined in Algorithm 1, we maintain N_s different samples of the lattice connectivity, that is, a set of binary-bit strings $\{\boldsymbol{s}_i, i = 1...N_s\}$. At each iteration, the algorithm performs the following steps:

1. Evaluate the objective function J_i for each sampled connectivity $\boldsymbol{s}_i, i = 1...N_s$ (Line 3-6 in Algorithm 1).
2. Select the best M samples that produce the lowest objective values and perturb them. The perturbation of bit strings is similar to the mutation operation in a genetic algorithm.
3. Replace the rest of the $N_s - M$ samples with new samples using an MCMC sampling step (Line 13-14 in Algorithm 1).

These steps repeat until the best objective value drops below a threshold (Line 7 in Algorithm 1).

Evaluation of objective function. Given a sampled connectivity, we evaluate the objective function J defined in §5.1. Since J depends on both the connectivity and the primitive cube sizes and the latter has not yet been determined, we treat the evaluation as another optimization problem, one that minimize the objective function over all possible cube sizes but with a fixed connectivity. This is a continuous optimization problem, which we solve in §5.3.

Random sample of connectivity. To initialize the set of lattice connectivities and to replace the worst $N_s - M$ samples at the third step of the algorithm, we need to sample bit strings $\boldsymbol{s}_i$. To this end, we use a simple rejection sampling scheme, starting by random sampling of a bit string. Since we must ensure the inlet and outlet are connected through primitive resonators, after sampling a bit string we verify whether the corresponding connectivity structure connects the inlet with the outlet (e.g., using a depth-first search on the lattice) and reject the sample if does not.

Connectivity perturbation. We perturb the connectivity string $\boldsymbol{s}_i$ using a mutation.Specifically, we randomly select a bit in a string s_i and flip it. In addition, this mutated string is subject to two constraints: (i) the corresponding connectivity structure needs to retain the connection between the inlet and the outlet; and (ii) the mutated bit needs to influence the resonator paths that connect the inlet and the outlet; otherwise, the mutation makes no difference to the connected component between the inlet and the outlet. We check the mutated bit string against both requirements and reject the mutation if it fails the check.

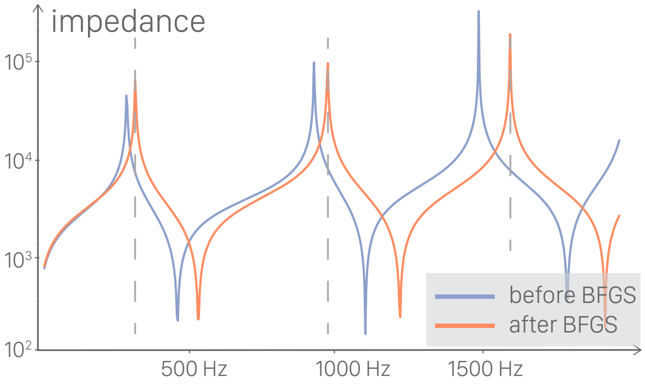


Figure 6: *Before and after BFGS optimization. Combinatorial sampling is difficult to converge to the user-specified target quickly due to its random nature. Enforcing local optimization for each sample reaches the desired acoustic target faster.*

5.3 Local Continuous Optimization

Next we discuss how to evaluate the objective function after sampling a lattice structure. This evaluation optimizes the cube sizes $\boldsymbol{u}$ of each primitive in the lattice structure in order to compute the minimal objective function value. To achieve this, we first compute the gradient of J with respect to $\boldsymbol{u}$ from Eq. (7),

$$\frac{\partial J(\boldsymbol{s},\boldsymbol{u})}{\partial \boldsymbol{u}} = 2\sum_{i=1}^{N_\omega}(g(\mathsf{T}(\boldsymbol{s},\boldsymbol{u},\omega_i)) - \bar{g}_i)\frac{\partial g(\mathsf{T}(\boldsymbol{s},\boldsymbol{u},\omega_i))}{\partial \boldsymbol{u}}. \tag{8}$$

The function g depends on the transmission matrix T, which further depends on the acoustic pressures and velocities at every port of all the primitive resonators, according to Eq. (6). To compute the partial derivative of g, applying the chain rule yields:

$$\frac{\partial g(\mathsf{T}(\boldsymbol{s},\boldsymbol{u},\omega_i))}{\partial \boldsymbol{u}} = \underbrace{\left(\frac{\partial g}{\partial \mathsf{T}}\frac{\partial \mathsf{T}}{\partial \boldsymbol{x}}\right)}_{\boldsymbol{m}^T}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{u}}, \tag{9}$$

where $\boldsymbol{x}$, as used in Eq. (6), stacks frequency-domain pressures and velocities of all the ports of the primitives. If N denotes the number of primitive resonators of the assembly, then $\boldsymbol{m}$ is a vector of the length $12N$, independent of the cube sizes of the primitives. $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{u}}$ is a $12N \times N$ matrix. To compute this matrix, recall that in Eq. (6), the matrix A depends on the cube sizes of the primitives, and $\boldsymbol{b}$ is a constant. Differentiating both sides of Eq. (6) with respect to $\boldsymbol{u}$ yields:

$$\mathsf{A}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{u}} + \frac{\partial \mathsf{A}}{\partial \boldsymbol{u}} = \boldsymbol{0}, \tag{10}$$

which is a linear system with N right-hand-side vectors, $-\frac{\partial \mathsf{A}}{\partial \boldsymbol{u}}$. Since A is assembled from the transmission matrices of all primitives in the assembly(recall §4.2), $\frac{\partial \mathsf{A}}{\partial \boldsymbol{u}}$ involves the derivatives of the transmission matrices with respect to the cube sizes. We compute them by interpolating our precomputed primitive transmission matrices.

It seems straightforward to compute $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{u}}$ by factorizing A only once and solving the linear system N times, and use Eq. (9) and (8) to compute the gradient of the objective function. However, if N is large, even the repeated back substitutions for solving Eq. (10) are slow. Especially when used in a Monte Carlo sampling step, this would significantly reduce the efficiency of the overall optimization algorithm.

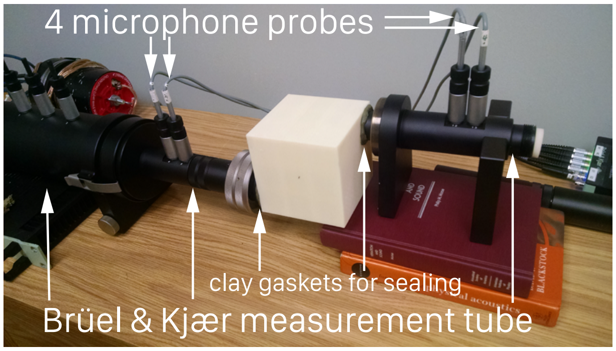


Figure 7: *Industrial laboratory measurement setup.*

Speedup with Adjoint Method. Fortunately, this computation can be largely accelerated using the adjoint method, one that has been applied in computer graphics mainly for animation control problems [McNamara et al. 2004; Wojtan et al. 2006; Barbič et al. 2009]. The key idea is based on the observation that computing a matrix-vector product, $\boldsymbol{m}^T\mathsf{B}$ such that $\mathsf{AB} = \mathsf{C}$, is equivalent to computing $\boldsymbol{t}^T\mathsf{C}$ such that $\mathsf{A}^T\boldsymbol{t} = \boldsymbol{m}$. The advantage of the latter is that only a single linear-system solve for the vector $\boldsymbol{t}$ is needed. In our problem, this amounts to first solving

$$\mathsf{A}^T\boldsymbol{t} = \left(\frac{\partial g}{\partial \mathsf{T}}\frac{\partial \mathsf{T}}{\partial \boldsymbol{x}}\right), \text{ followed by computing } \frac{\partial g}{\partial \boldsymbol{u}} = \boldsymbol{t}^T\frac{\partial \mathsf{A}}{\partial \boldsymbol{u}}.$$

For all our examples, this method results in nearly 10× speedups over the straightforward approach.

With the computation of the gradient $\frac{\partial J}{\partial \boldsymbol{u}}$ depicted, we apply it to a quasi-Newton method to minimize J. In our implementation, we use the Limited-memory BFGS Bounded (L-BFGS-B) method [Zhu et al. 1997]. In practice we found local gradient descent step complements the combinatorial sampling. Figure 6 illustrates the effectiveness of the local optimization of the impedance curve.

6 Results

We now present the experiments we conducted to test our method. In all examples, we sample the frequency range every 3Hz from 20Hz to 5kHz to precompute transmission matrices. The cube size of the primitive resonator varies depending on specific applications: For muffler design and acoustic signatures, the cube size is between 6mm and 2cm, sampled every 1mm. For laboratory tests and wind instrument design, the cube size is between 25cm and 35mm, also sampled every 1mm. The precomputation takes a few hours on a 16-core cluster.

We fabricated our designs using Stratasys uPrint SE Plus, a filament-based 3D printer with a layer resolution at 0.254mm. We use ABS-P430 plastic as the model material and a dissolvable support material which can be washed away upon finish. The fabrication time varies from a few hours to a day, primarily depending on geometric size of a given model.

6.1 Validation

The fundamental building block of our assembly structure optimization is the fast computation of a transmission matrix for an assembly (recall §4.2). We validate its accuracy using finite-element-method (FEM) simulation and industrial laboratory tests.

Finite-element simulation. We compute the transmission loss using Code Aster [Aubry 2013], a well-developed and carefully tested finite-element solver for mechanics. We follow the routine outlined in §3, solving for the acoustic velocities with different boundary

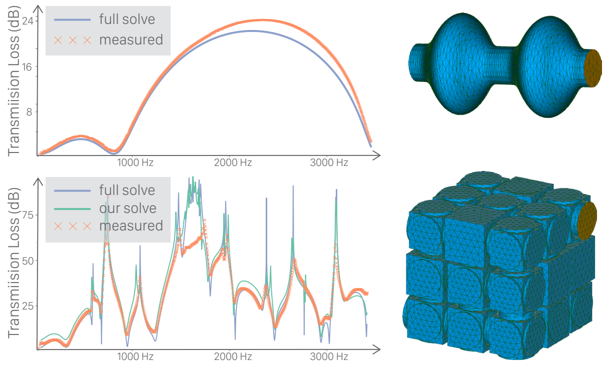


Figure 8: *Double muffler and cube measured by Brüel & Kjær . Our method agrees closely with both the expensive FEM solve and the lab measurement. There is a large difference around 1600Hz, where the measurement input signal does not have sufficient power to pass through. This is caused by the wide and high transmission loss values around this region, according to the lab technicians.*

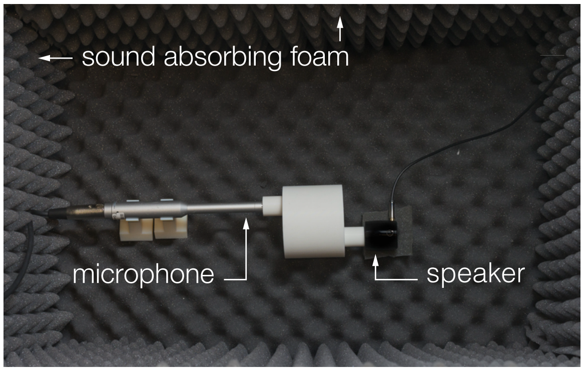


Figure 10: Recording setup *to record the sounds before and after our filtering. The chamber inner surface is surrounded by sound absorbing foam to minimize ambient noise from outside as well as the wave reflection/refraction inside the chamber.*

conditions using Code Aster. We then compute the transmission matrix by assembling and solving the Equation (4).

Industrial laboratory test. We sent fabricated samples to Brüel & Kjær 's acoustic laboratory for independent, third-party tests conducted by their acoustic professionals. Brüel & Kjær is the world's largest manufacturer and supplier of acoustic measurement equipment and solutions. They measured the transmission loss of our samples using Brüel & Kjær 4206-T measurement tubes with the 4-microphone technique [Tao and Seybert 2003], sweeping the frequency range every 4Hz from 20Hz to 3500Hz under the condition of 21°C (room temperature), 98.9kPa (pressure), and 44% of relative humidity. To ensure best acoustic seal during the tests, clay gaskets were also added between the measurement tubes and our test samples (see Fig. 7).

Comparison. The comparison shows that our fast computation of transmission loss agrees with both the finite-element simulation and laboratory experiments closely, as in Figure 8. The top plot in Figure 8 validates the agreement between the finite-element simulation and the laboratory tests using a double-chamber muffler, which is known to be an effective broadband filter. It lacks the curve from our computation model, simply because this model is not made from our primitive resonators. We use this test to examine the use of the numerical and experimental methods. The bottom plot reports the transmission loss of an assembly muffler made of 3×3×3 primitive resonators, comparing the results from finite-element simulation (blue curve), Brüel & Kjær 's laboratory measurement (orange dots), and our fast computation (blue curve). They all agree with each other closely. Particularly, our computational model is able to predict the peaks and valleys on the transmission loss curve, with the differences from the measurement less than 20Hz on average. These peaks and valleys indicate the most and the least attenuated frequencies when sound passes through the filter, and they will be of practical importance to control when one designs a muffler, as demonstrated later in §6.2.

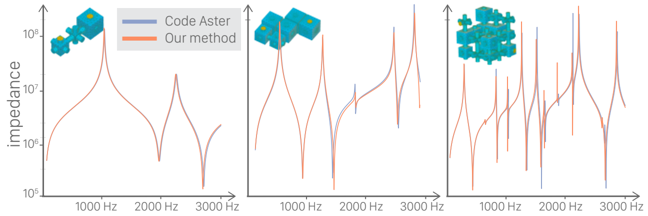


Figure 9: Impedance comparison *with Code Aster. In the sequence of three models with increasing complexity, our method agrees with Code Aster closely.*

We also run three validation tests on impedance curves to compare the error of our fast computation and full FEM solve. Figure 9 shows that our method robustly computes the impedance curve and introduces slight numerial instability as the model gets more complicated. In terms of computational performance, our method is much faster than the finite-element simulation. For example, to compute the transmission loss curve of this 3×3×3 resonator assembly, which involves computation at 1000 frequency samples, our method takes 1.2 seconds, while the finite-element method takes around 22 hours, resulting in 77k× speedup.

6.2 Application I: Muffler Design

Man-made mechanisms produce noise, with clear patterns exhibited in the sound spectrum. For instance, the aircraft and automobile engine noise have pronounced frequency components related to revolutions per minute (RPM) of the engine cranks. The car horns have particular frequency patterns regulated by local government (i.e., 390Hz and its harmonics in U.S.). Traditionally, mufflers are designed at a large granularity, aiming to filter sound in a wide band of frequency range, partially because of its ease of control using relatively simple muffler geometries.

Engine noise muffler. Here we demonstrate the possibility of controlling muffler behavior at finer granularity using our modular

		Optimization		
	# DoFs	type	# targets	avg. time
PIGGY	21	Z	3	9m
OCTOPUS	76	TL	8	2h10m
BOB	258	Z	13	7h
ENGINEMUFF	20	TL	3	15m
EARENGINE	51	TL	3	11m
EARHORN	127	TL	7	1h15m
HIPPO	122	Z	4	51m

Table 1: Optimization Statistics *The number of DoFs is the sum of number of feasible nodes and number of connecting faces. Optmization time is averaged over all the optimized targets for each example. The number of targets is the number of peaks and valleys that we want to optimize in each example.*

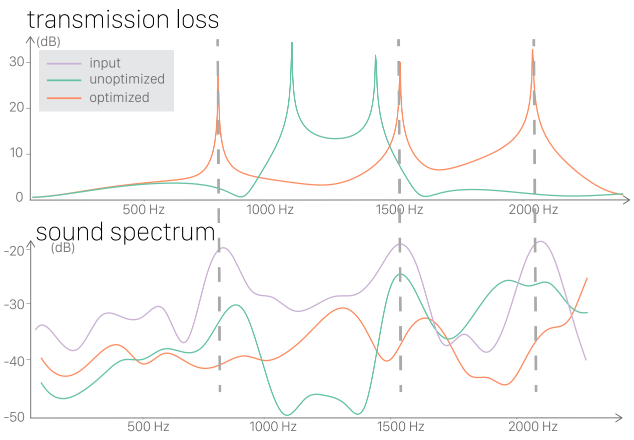


Figure 11: Engine Muffler. *We compare an unoptimized muffler and an optimized one. The three noisy peaks are suppressed to lower levels with the optimized muffler.*

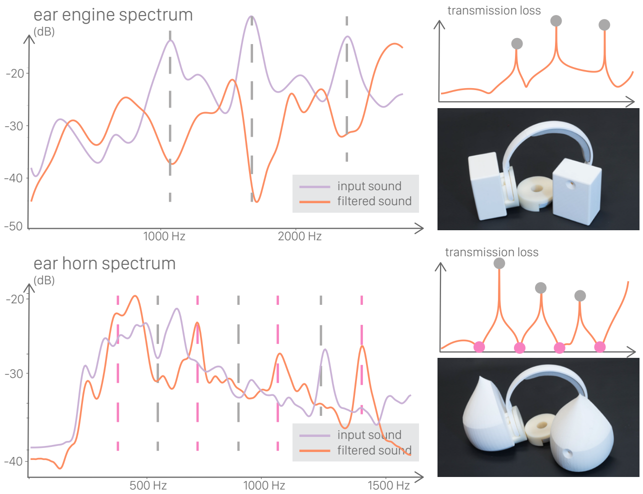


Figure 12: Acoustic earmuff. *We customized two earmuffs (top and bottom) that can be modularly mounted in a headset. In the plots on the left, the orange curves show the filtered sounds where the peaks and valleys correspond to the purple points on the right.*

filter, because of its ability to construct complex muffler structures. We aim to construct mufflers that selectively attenuate sound near a set of discrete frequency values. Our first example is to attenuate a recorded engine noise, which has peaks in frequency domain at 850Hz, 1550Hz, and 2100Hz. To filter these frequency components, we uniformly sample frequencies $\omega_i, i = 1..N_\omega$, which include the peak frequencies. We then define an objective function (7), in which the $g(\mathsf{T}(\omega_i))$ compute the transmission loss (using Equation (3)), and $\bar{g}_i$ is a large value at the peak frequencies and zero otherwise. The muffler structure is optimized with a combination of 8 resonators, and the quantitative results is ploted in Figure 11 (orange curve). We also compare the result with a muffler that has the same volume of the internal chamber but unoptimized structure (blue curve), showing that the optimized muffler indeed attenuates the unwanted frequency peaks. Please refer to the video for their audible differences.

Acoustic earmuffs. Our next example of muffler design is for acoustic earmuffs. There has been a variety of acoustic earmuffs targeting at different application scenarios, such as hunting, construction work, and riding motorcycles. While some earmuffs also employ a microphone mounted in the headset to actively reduce a broadband noise, many others reply on acoustic structures and materials for noise reduction and have the advantage of robustness and long working time (without any battery). Our method can also design passive earmuffs, but complement this category by allowing user customization.

We demonstrate two earmuffs that can be modularly mounted in the headset (see video) and switch to different ones when needed. The first one is customized to reduce engine noise having peak frequencies at 1000Hz, 1600Hz, and 2200Hz (Figure 12-top). The second one is designed for riding motorcycles (Figure 12-bottom). We aim for reducing aerodynamic noise while allowing the rider to hear car horns for the sake of safety. Therefore the objective function is to suppress a broadband noise without heavily filtering car horn sound at 390Hz and its harmonics. Both earmuffs are computed by optimizing the structure of 42 primitive resonators. As shown in the plots of Figure 12, our mufflers indeed filter out frequency components we desired.

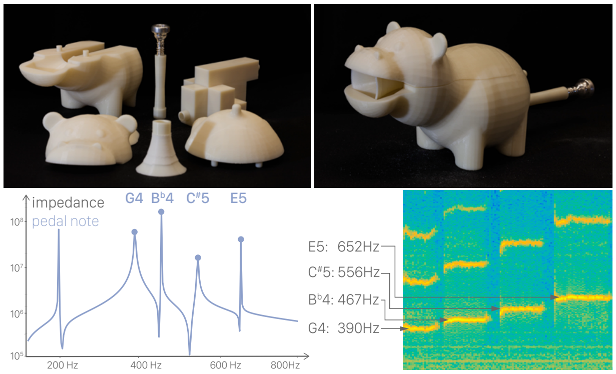


Figure 13: Wind instrument. *We optimize for 4 notes for the* HIPPO *trumpet to play, located at the impedance maximums, the first one being the pedal note, a sustained tone. The spectrogram of our recording confirms the accuracy of our optimization framework.*

6.3 Application II: Wind Instruments

Acoustic resonator is a key part of wind instruments. While nonlinear excitation mechanism of a wind instrument (such as the mouth piece) is also important [Allen and Raghuvanshi 2015], critically affecting the timbre of the instrument, the acoustic resonator serves to modulate the excitation and controls the pitch. In particular, it is known that the playable notes of a wind instrument correspond to the peaks of the resonator's input impedance, except its first peak (called pedal note).

We applied our method to customize trumpets. Our customization is twofold: we wish to control the set of notes that a trumpet can play while customizing its shape, which, in our case, a cartoon hippopotamus shape. The resulting trumpet still relies on the standard mouthpiece for excitation. Given a set of notes, we define an objective function (7) that maximizes the impedance values at the frequencies of those notes. We customized 3 different trumpets, whose playable notes are [G4,D5], [C4, G4, C5], and [G4, B^{b}4, C$^\#$5, E5], respectively. As shown in Figure 13, our optimized primitive assembly can be placed inside of the hippopotamus shape and are playable. In the supplemental video, we demonstrate that the resulting musical notes produced by our customized resonators are in tone, whereas the unoptimized resonator deviates a lot from the desired notes. We note that while it is known that the players can "bend" the notes by around a semitone, it is difficult to rely on this controllability to play

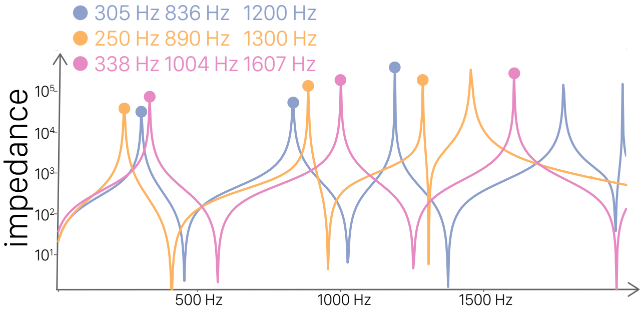


Figure 14: Acoustic tagging. *We optimize three identical piggy shapes such that they all have different impedance curves. When tapped with a palm on their nose, the filtered sounds are different. The iPhone application used for recognition is shown in Fig. 1.*

in tune especially without our assembly optimization.

6.4 Application III: Acoustic Signatures

Our acoustic filter design opens up possibilities for new applications. Inspired by the recent work on creating tangible input devices that interact through acoustics [Laput et al. 2015; Savage et al. 2015], we demonstrate two examples, namely *acoustic tagging* and *acoustic encoding*.

Acoustic tagging. Our method enables a new way of tagging 3D shapes. This is similar in spirit to the recent work on tagging 3D fabricated shapes by modulating material distribution and decoding using Terahertz imaging [Willis and Wilson 2013], but from a completely different perspective, the acoustics.

Our key idea is to embed tags into the acoustic filtering effects of a shape, by computationally optimizing its internal structure without largely changing its visual appearance, as long as the shape has two holes serving as the inlet and outlet (Figure 1-a). Even with a single tapping using a palm at a hole, one can produce an acoustic wave passing through the internal structure and output a filtered noise. A simple FFT-based algorithm can recognize the output sound and decode the tags. Compared to the existing tagging approaches, this method requires no electronics during installation and detection (unlike Radio Frequency Identification tags) or multi-material fabrication (unlike [Willis and Wilson 2013]). It relies on our optimization method to physically realize a specific acoustic signature that can be reliably read by a computer program. In our examples, we choose to make each tag to have distinct peaks of their impedance curves, and thereby allowing for robust, FFT-based decoding.

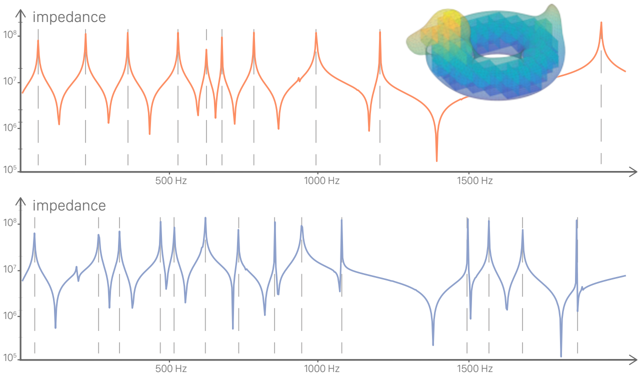
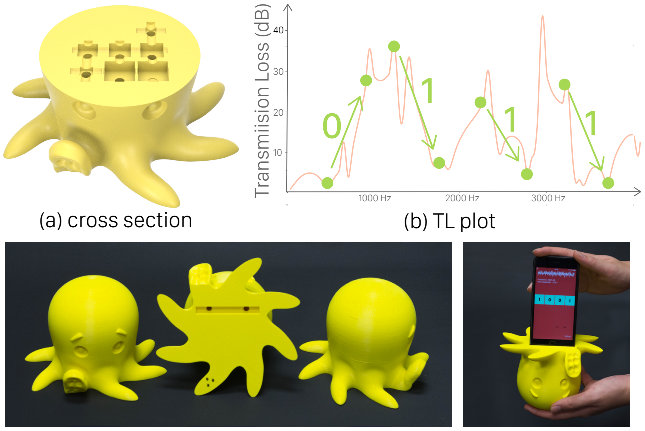


Figure 15: Bob. *In this example we optimize for two sets of frequency peaks (top and bottom); each has more than 10 target frequency peaks, indicated by the dotted vertically lines. For both cases, out optimized acoustic filters are able to achieve the desired peaks.*



(a) cross section

(b) TL plot

(c) fabricated octopus with different bit pattern

Figure 16: Acoustic encoding. *By embedding more voxels in the geometry, we achieve finer-grained control of the acoustic properties, exemplified by encoding 4 binary bits of information.*

We demonstrate this approach by fabricating three identical piggy shapes (Figure 1-b), each with an target acoustic impedance curve peaking at different frequency values (Figure 14). Using our Acoustic Voxels approach, we realize these impedance curves with our primitive assemblies. We have implemented a simple iPhone application that decodes a recorded tapping sound and detect the resonant frequencies which correspond to the local maximums on the impedance curve. As shown in the video (and Figure 1), the iPhone application can reliably detect the tags and identify the piggies.

To take the complexity of our optimized muffler further, we voxelized Bob, the duck-shaped lifesaver, with the inlet at the beak and the outlet at the tail (Figure 15). We optimized for two sets of frequency peaks on the impedance curve; each has more than 10 peaks. We evaluated this example by comparing the target impedance against the optimized impedance computed using our simulation model, without fabricating the models, because of the 3D printer's limitation on the geometric size of the fabricated shapes (Figure 15). This example promises for tagging a large pool of objects or controlling the filtering behaviors at a finer granularity in future.

Acoustic encoding. Taking one step further, we demonstrate the ability to encode bit strings, which can be interpreted as virtually any type of information, akin to the idea of QR code but visually less distracting. The idea is again using acoustic filter to modulate frequencies in a controlled way. Instead of controlling acoustic impedance curves, here we explore the possibility of encoding in the transmission loss curve, with a simple coding scheme: To encode N bits of information, we evenly sample 2N frequency values and group the samples pairwise. Let the frequencies are grouped as $(\omega_1, \omega_2), (\omega_3, \omega_4), ..., (\omega_{2N-1}, \omega_{2N})$. We encode a "1" at the i-th bit if the transmission loss value at ω_{2i-1} is smaller than that at ω_{2i}, and encode a "0" if the value at ω_{2i-1} is larger than that at ω_{2i} (Figure 16-b). By setting an objective function that maximizes and minimizes the transmission loss at corresponding frequencies, we optimize for an acoustic filter that physically realizes this coding scheme.

We fabricated three objects with an identical, octopus-like surface shape (Figure 16), and use them to encode different 4-bit strings, including "0000", "1001", and "0111". As shown in the video, we have implemented another iPhone application that plays a white noise from its speaker while simultaneously recording from its microphone. When aligning the iPhone speaker and microphone with two holes (i.e., the inlet and outlet) on the object, the white noise passes through the internal structure of the shape and gets filtered. By detecting the filtered amplitudes at the pre-specified frequencies ω_i, the application decodes the bit strings. In future, the application can be made to interpret the bit strings in a specific context and enable other new applications (§7).

7 Conclusion and Future Work

Our method is mostly suitable for controlling impedance and transmission loss at discrete frequencies, but has limited ability to control a broadband of frequencies. For this purpose, the traditional muffler design is more suitable. Currently, we use only one rigid material and optimize the filter's chamber shape, while automotive mufflers often use composite materials. Also, we consider only a single type of primitive resonators. Extending our method to more primitive shapes and materials can offer a larger palette for better acoustic filtering control. Practically, we have some difficulties to ensure the internal structure of a filter being thoroughly cleaned after 3D printing, as it is hard to examine given its structural complexity.

So far, we have demonstrated the control transmission loss and impedance curves up to 4500Hz. While this is motivated by the fact that most muffler and instrument applications operate in this frequency range, we are restricted by the precomputation time needed for computing transmission matrices at higher frequencies, as it requires significantly higher finite-element resolution and longer time to solve the Helmholtz equation at higher frequencies. In the example of Bob, we optimized for more then ten peaks in the impedance curve. To control more peaks, a higher resolution of lattices is needed, leading to a much longer optimization time. It is an interesting future work to further speed up the optimization process for more detailed control of acoustic filtering.

In conclusion, we present Acoustic Voxels, a computational method that optimizes assembly of primitive resonators to realize a target acoustic filtering property, described in acoustic impedance or transmission loss. We demonstrated our algorithm with three types of applications, including muffler design, wind instruments, and a new way of customizing 3D-printed shapes with acoustic signatures. In future, this idea can be carried over to design acoustic filters at different scales, such as at high frequencies for ultrasonic imaging and at low frequencies for improving room acoustics. Further, we are interested in exploring new HCI applications enabled by acoustic signatures as well as new acoustic meta-materials enabled by computational optimization.

A Precomputation of Transmission Matrix

For a general primitive shape with six ports, precomputing the tranmission matrix T^{ω} amounts to solving the Helmholtz equation 6 times, with different Neumann boundary conditions. In our implementation, for the i-th solve, we set $v_i = 1$ and $v_j = 0$ if $j \neq i$. The transmission matrix is calculated as:

$$T^{\omega}_{ji} = p^{i}_{j} \tag{11}$$

where p^i_j is the solution on face k under the ith boundary condtion.

Acknowledgements

We thank the anonymous reviewers for their feedback. We are grateful to Jui-Hsien Wang for early discussion, Ioannis Kymissis and Andrei Shylo for supporting 3D printing facilities, Chang Xiao for iOS development help, Yonghao Yue for rendering advices, Yun Fei for improving the video, Timothy Sun for video narrative, Henrique Maia for proofreading, and Alec Jacobson for discussion on wind instrument examples. We also thank Syed Haris Ali for open sourcing EZAudio library. Piggy, Octopus, and Bob are provided by courtesy of www.craftsmanspace.com, Makerbot, and Keenan Crane, respectively. This research was supported in part by the National Science Foundation (CAREER-1453101) and donations from Adobe. 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.

References

Allen, A., and Raghuvanshi, N. 2015. Aerophones in flatland: Interactive wave simulation of wind instruments. *ACM Trans. Graph. 34*, 4 (July), 134:1–134:11.

Angell, T., Jiang, X., and Kleinman, R. 1997. A distributed source method for inverse acoustic scattering. *Inverse Problems 13*, 2, 531.

Aubry, J.-P. 2013. *Beginning with Code_Aster*. Framasoft.

Barbič, J., da Silva, M., and Popović, J. 2009. Deformable object animation using reduced optimal control. *ACM Trans. Graph. 28*, 3 (July), 53:1–53:9.

Bharaj, G., Levin, D. I. W., Tompkin, J., Fei, Y., Pfister, H., Matusik, W., and Zheng, C. 2015. Computational design of metallophone contact sounds. *ACM Trans. Graph. 34*, 6 (Oct.).

Bickel, B., Bächer, M., Otaduy, M. A., Lee, H. R., Pfister, H., Gross, M., and Matusik, W. 2010. Design and fabrication of materials with desired deformation behavior. *ACM Trans. Graph. 29*, 4 (July).

Braden, A. C., Newton, M. J., and Campbell, D. M. 2009. Trombone bore optimization based on input impedance targets. *The Journal of the Acoustical Society of America 125*, 4.

Caloz, C., and Itoh, T. 2005. *Electromagnetic metamaterials: transmission line theory and microwave applications*. John Wiley & Sons.

Chiu, M.-C. 2010. Shape optimization of multi-chamber mufflers with plug-inlet tube on a venting process by genetic algorithms. *Applied Acoustics 71*, 6, 495–505.

Christopoulos, C. 2006. *The Transmission-Line Modeling (TLM) Method in Electromagnetics*. Morgan & Claypool Publishers.

De Lima, K. F., Lenzi, A., and Barbieri, R. 2011. The study of reactive silencers by shape and parametric optimization techniques. *Applied Acoustics 72*, 4, 142–150.

Dokmanić, I., Parhizkar, R., Walther, A., Lu, Y. M., and Vetterli, M. 2013. Acoustic echoes reveal room shape. *Proceedings of the National Academy of Sciences 110*, 30, 12186–12191.

Feijóo, G. R., Oberai, A. A., and Pinsky, P. M. 2004. An application of shape optimization in the solution of inverse acoustic scattering problems. *Inverse problems 20*, 1, 199.

Funkhouser, T., Carlbom, I., Elko, G., Pingali, G., Sondhi, M., and West, J. 1998. A beam tracing approach to acoustic modeling for interactive virtual environments. In *Proc. of SIGGRAPH 98*, 21–32.

FUNKHOUSER, T. A., MIN, P., AND CARLBOM, I. 1999. Real-time acoustic modeling for distributed virtual environments. In *Proc. of SIGGRAPH 99*.

HÄMÄLÄINEN, P., ERIKSSON, S., TANSKANEN, E., KYRKI, V., AND LEHTINEN, J. 2014. Online motion synthesis using sequential monte carlo. *ACM Trans. Graph. 33*, 4 (July).

INGARD, U. 2009. *Noise reduction analysis*. Jones & Bartlett Publishers.

JAMES, D. L., BARBIC, J., AND PAI, D. K. 2006. Precomputed Acoustic Transfer: Output-sensitive, accurate sound generation for geometrically complex vibration sources. *ACM Trans. Graph. 25*, 3 (July), 987–995.

KAC, M. 1966. Can one hear the shape of a drum? *American Mathematical Monthly*, 1–23.

KAUSEL, W. 2001. Optimization of brasswind instruments and its application in bore reconstruction. *Journal of New Music Research 30*, 1, 69–82.

KIRKPATRICK, S., GELATT, C. D., AND VECCHI, M. P. 1983. Optimization by simulated annealing. *Science 220*, 4598, 671–680.

LAPUT, G., BROCKMEYER, E., HUDSON, S. E., AND HARRISON, C. 2015. Acoustruments: Passive, acoustically-driven, interactive controls for handheld devices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ACM, 2161–2170.

LI, D., FEI, Y., AND ZHENG, C. 2015. Interactive acoustic transfer approximation for modal sound. *ACM Transactions on Graphics 35*, 1.

MCNAMARA, A., TREUILLE, A., POPOVIĆ, Z., AND STAM, J. 2004. Fluid control using the adjoint method. 449–456.

MEHRA, R., RAGHUVANSHI, N., ANTANI, L., CHANDAK, A., CURTIS, S., AND MANOCHA, D. 2013. Wave-based sound propagation in large open scenes using an equivalent source formulation. *ACM Trans. Graph. 32*, 2 (Apr.), 19:1–19:13.

MIGUEZ, J., CRISAN, D., AND DJURIC, P. M. 2010. Sequential monte carlo methods for the optimization of a general class of objective functions. *SIAM Journal on Optimization*.

MONKS, M., OH, B. M., AND DORSEY, J. 2000. Audioptimization: goal-based acoustic design. *Computer Graphics and Applications, IEEE 20*, 3, 76–90.

MUNJAL, M. 2014. *Acoustics of Ducts and Mufflers*, second ed. John Wiley & Sons.

NORELAND, J. D., UDAWALPOLA, M. R., AND BERGGREN, O. M. 2010. A hybrid scheme for bore design optimization of a brass instrument. *Journal of the Acoustical Society of America 128*, 3, 1391–1400.

PANETTA, J., ZHOU, Q., MALOMO, L., PIETRONI, N., CIGNONI, P., AND ZORIN, D. 2015. Elastic textures for additive fabrication. *ACM Trans. Graph. 34*, 4, 135.

PEGORARO, V., WALD, I., AND PARKER, S. G. 2008. Sequential monte carlo adaptation in low-anisotropy participating media. In *Computer Graphics Forum*, vol. 27, Wiley Online Library.

PIERCE, A. D., ET AL. 1991. *Acoustics: an introduction to its physical principles and applications*. Acoustical Society of America Melville, NY.

RAGHUVANSHI, N., AND SNYDER, J. 2014. Parametric wave field coding for precomputed sound propagation. *ACM Trans. Graph. 33*, 4 (July), 38:1–38:11.

RAGHUVANSHI, N., SNYDER, J., MEHRA, R., LIN, M., AND GOVINDARAJU, N. 2010. Precomputed wave simulation for real-time sound propagation of dynamic sources in complex scenes. *ACM Trans. Graph. 29*, 4 (July), 68:1–68:11.

RIENSTRA, S. W., AND HIRSCHBERG, A. 2003. An introduction to acoustics. *Eindhoven University of Technology 18*, 19.

RITCHIE, D., MILDENHALL, B., GOODMAN, N. D., AND HANRAHAN, P. 2015. Controlling procedural modeling programs with stochastically-ordered sequential monte carlo. *ACM Transactions on Graphics (TOG) 34*, 4, 105.

ROBERT, C., AND CASELLA, G. 2013. *Monte Carlo statistical methods*. Springer Science & Business Media.

SAVAGE, V., HEAD, A., HARTMANN, B., GOLDMAN, D. B., MYSORE, G., AND LI, W. 2015. Lamello: Passive acoustic sensing for tangible input components. In *Proc. CHI 2015*, ACM, 1277–1280.

SCHUMACHER, C., BICKEL, B., RYS, J., MARSCHNER, S., DARAIO, C., AND GROSS, M. 2015. Microstructures to control elasticity in 3d printing. *ACM Trans. Graph. 34*, 4, 136.

SELAMET, A., DENIA, F., AND BESA, A. 2003. Acoustic behavior of circular dual-chamber mufflers. *Journal of Sound and Vibration 265*, 5, 967–985.

SIGMUND, O. 1994. Materials with prescribed constitutive parameters: an inverse homogenization problem. *International Journal of Solids and Structures 31*, 17, 2313–2329.

STETTNER, A., AND GREENBERG, D. P. 1989. Computer graphics visualization for acoustic simulation. *SIGGRAPH Comput. Graph. 23*, 3 (July).

TAKALA, T., AND HAHN, J. 1992. Sound rendering. In *Computer Graphics (Proceedings of SIGGRAPH 92)*, 211–220.

TAO, Z., AND SEYBERT, A. 2003. A review of current techniques for measuring muffler transmission loss. Tech. rep., SAE Technical Paper.

TSINGOS, N., FUNKHOUSER, T., NGAN, A., AND CARLBOM, I. 2001. Modeling acoustics in virtual environments using the uniform theory of diffraction. In *ACM SIGGRAPH 2001*.

UMETANI, N., KAUFMAN, D. M., IGARASHI, T., AND GRINSPUN, E. 2011. Sensitive couture for interactive garment modeling and editing. *ACM Trans. Graph. 30*, 4, 90.

WILLIS, K. D. D., AND WILSON, A. D. 2013. Infrastructs: Fabricating information inside physical objects for imaging in the terahertz region. *ACM Trans. Graph. 32*, 4 (July), 138:1–138:10.

WOJTAN, C., MUCHA, P. J., AND TURK, G. 2006. Keyframe control of complex particle systems using the adjoint method. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '06, 15–23.

ZHU, C., BYRD, R. H., LU, P., AND NOCEDAL, J. 1997. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Softw. 23*, 4.

ZORAN, A. 2011. The 3d printed flute: Digital fabrication and design of musical instruments. *Journal of New Music Research 40*, 4, 379–387.