Received 16 April 2024, accepted 9 May 2024, date of publication 15 May 2024, date of current version 23 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3401226

APPLIED RESEARCH

SonoTraceLab—A Raytracing-Based Acoustic Modeling System for Simulating Echolocation Behavior of Bats

WOUTER JANSEN, (Graduate Student Member, IEEE), AND JAN STECKEL
Cosys-Lab, Faculty of Applied Engineering, University of Antwerp, 2000 Antwerp, Belgium
Flanders Make, 3920 Lommel, Belgium

Corresponding author: Jan Steckel (jan.steckel@uantwerpen.be)

This work was supported by Bijzonder OnderzoeksFonds (BOF) Research Grant by the University of Antwerp.

ABSTRACT Echolocation is the prime sensing modality for many species of bats, who show the intricate ability to perform a plethora of tasks in complex and unstructured environments. Understanding this exceptional feat of sensorimotor interaction is a key aspect into building more robust and performant man-made sonar sensors. In order to better understand the underlying perception mechanisms it is important to get a good insight into the nature of the reflected signals that the bat perceives. While ensonification experiments are an important way to better understand the nature of these signals, they are as time-consuming to perform as they are informative. In this paper we present SonoTraceLab, an open-source software package for simulating both technical as well as biological sonar systems in complex scenes. Using simulation approaches can drastically increase insights into the nature of biological echolocation systems, while reducing the time- and material complexity of performing them.

INDEX TERMS Sonar, microphone arrays, sound source localization, acoustic signal processing, ultrasound, hardware design, 3D ultrasound.

I. INTRODUCTION

Bats exhibit a wide range of behaviors that are of interest to scientists who want to understand the underlying mechanisms of perceiving the world using peculiar sensing modalities [1]. Indeed, bats rely mainly on echolocation to perceive their world, which poses an entirely different set of challenges to these animals compared to animals relying on vision to navigate their surroundings [2], [3], [4], [5]. To better understand this echolocation behavior, experiments have been performed to understand better the environments bats encounter, and how these environments manifest themselves in the sensory domain of bats. For example, nectar-feeding bats have been shown to be attracted to specialized leaves formed by neo-tropical pitcher plants [6], [7], [8], [9]. Similarly, the neo-tropical bat Micronycteris *microtus* can hunt insects in dense vegetation [2], [10], [11], and smooth surfaces have been shown to attract bats, as the echolocation signature of such surfaces resembles bodies of water [12], [13]. All these aforementioned studies have in common that the animals clearly react to specific properties of the scene being ensonified, and therefore, to understand the behavior, the nature and intricate details of the reflected signals should be well understood.

In order to better understand the echolocation behavior of bats, a plethora of ensonification experiments have been performed. Indeed, for static scenes such as the foraging behavior of bats close to pitcher plants [9] or bats hunting motionless prey [2], [11], and the hunting behavior of trawling bats who hunt over water surfaces [14], ensonification experiments have revealed insights into the underlying mechanisms that allow bats to solve these complex tasks successfully. Furthermore, in dynamic scenes, such as the recognition of fluttering prey items, ensonification experiments have yielded profound insights into potential signal processing schemes that underpin the hunting strategies [15], [16]. These successes strongly advocate for the performance of these ensonification experiments. However, ensonification

The associate editor coordinating the review of this manuscript and approving it for publication was Chengpeng Hao.

experiments come with significant drawbacks. Firstly, these experiments are complex to perform from both a mechanical and an acoustic point of view. Acoustically, it is important to accurately reproduce the external echolocation peripherals of the bats to yield relevant information about the interaction of the reflector filtering and the external acoustic filtering (called the Head-Related Transfer Function, HRTF) of the bat's emission and reception organs. Approaches to solving this have been found in implementing artificial, 3D printed pinnae [17], [18], [19], or using arrays of microphones and emitters to accurately replicate the directivity patterns of the bat [20], [21]. Furthermore, some bats utilize extremely high bandwidths in their echolocation calls [22], up to and beyond 180kHz, which poses significant issues in practical implementations of both emitters and receivers.

Mechanically, the full 6-DOF (Degrees Of Freedom) of the bat in relationship to the ensonified object needs to be controlled accurately. Typically, this is done using some kind of robotic setup using either a custom-made rig [8] or standard robotic arms [23]. However, these kind of setups introduce all kinds of additional complexity, from the ill-posedness of inverse kinematics [24] to unwanted additional reflections from the environment occurring and tainting the experimental data.

To overcome the previously detailed limitations, one can resort to simulation approaches. Indeed, acoustic wave theory is a well-understood field [26], and many successful simulation paradigms exist. Most notably, Finite Element Methods (FEM) [27], [28], [29], Boundary Element Methods (BEM) [30], [31] and Ray acoustics [32], [33], [34] are typical approaches towards solving the Helmholtz equations resulting from the acoustic simulation setup. Where FEM and BEM scale in computational complexity with the voxel size, they allow direct solutions to the wave equation to be calculated, causing them to be a more realistic approximation to the underlying physical problem. On the other hand, computational solutions based on ray acoustics do not scale directly with the scene size in relation to the wavelength, making them ideal for large-scale simulations. However, sound waves are not ideally modelled as rays under all conditions [26], which requires attention when implementing simulations where diffraction effects are of importance.

Furthermore, commercial simulation packages that implement these FEM and BEM approaches, such as Comsol [35] or Siemens Simcenter Acoustics [36] allow solving acoustical problems with one or multiple of the aforementioned simulation approaches. These simulation packages offer a powerful simulation engine which is validated in many industrial contexts. However, these simulation packages are often expensive to license and complex because to use of their broad applicability to a wide array of problem types. Indeed, the whole workflow of simulating complex scenarios encompasses modelling, meshing, parameterizing, simulating and post-processing of the data. Finally, that data needs to be exported to analysis software such as Matlab for subsequent hypothesis testing. Also, scripting and expanding these simulation engines is complex, because of the closed-source nature of these products. Despite all their apparent disadvantages, these tools might offer an interesting entry point to simulating echolocation scenarios, as demonstrated by the simulations performed in [37].

Alternatively to these closed-source commercial products, open-source simulation engines exist, such as k-Wave [38] or FIELD-II [39]. These solutions are free to use, open-source, and extremely useful in simulating ultrasound propagation. Indeed, FIELD-II is one of the default simulation engines used to simulate medical ultrasound applications [40], [41], [42], [43], and inspired other simulation engines for the same purpose of medical ultrasound, such as SIMUS [44]. However, these simulation engines are not ideal for the simulation of large-scale echolocation scenarios such as the echolocaiton behavior of bats. Furthermore, these software systems solve the Helmholtz equation through a finite element approach, which makes them ideally suited to relatively small problems in terms of the size of the scene compared to the wavelengths. Indeed, FEM scales cubicly in 3D with the voxel size, which causes problems to become computationally intractable for large scenes [45]. Another open-source application is the Bellhop acoustics toolbox [46], [47], [48]. This toolbox allows the simulation of acoustic pressure fields, and is focused for underwater acoustic simulation. However, it does not use mesh-based environment descriptions, which is something we want to implement in our simulation engine.

In order to overcome the limitations of these existing solutions, we set out to develop SonoTraceLab. This acoustic simulation engine (1) is open-source, (2) allows simple imports of arbitrary geometry without going through complex meshing procedures, (3) is tailored towards solving problems typically occurring in the investigation of active ultrasound sensing (both for biological as well as robotic ultrasound systems) and (4) allows the simulation of medium-sized environments (mesh sizes up to a million triangles). SonoTraceLab also allows for including arbitrary HRTF patterns in a post-processing step, a feature important for analyzing echolocation behavior, which is not feasible in currently available simulators. In what follows, we will first expand on the simulation paradigm powering SonoTraceLab, detailing how the simulation engine functions and its inputs and outputs. Next, we will demonstrate some typical scenarios encountered in bat echolocation, which will be validated using real-world measurements. Finally, we will demonstrate how our open-source implementation can be used, and then draw some conclusions and discuss the limitations of our work.

II. SIMULATION APPROACH

A typical simulation setup can be seen in figure 1. An echolocating agent (in this case, the bat Micronycteris *microtus* approaches a complex target, which in this case is a leaf

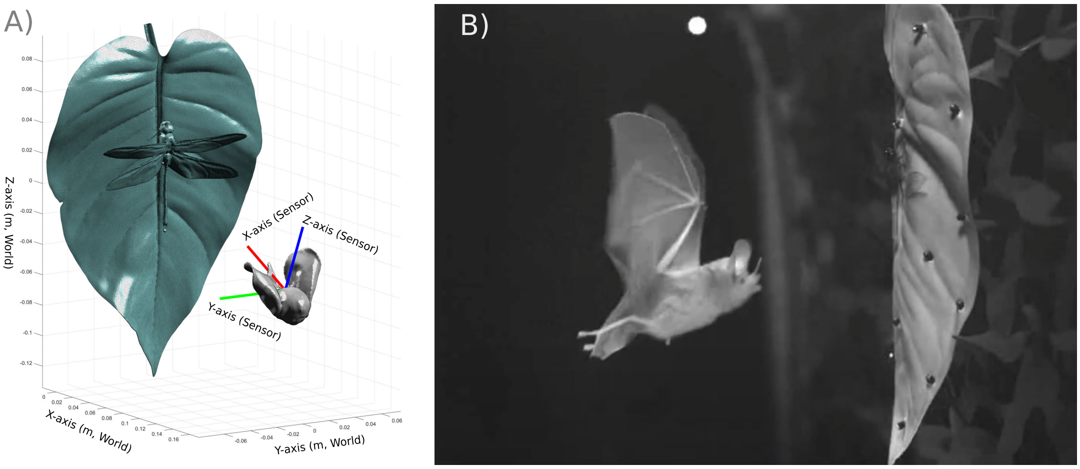


FIGURE 1. **A typical setup for an echolocation experiment. Panel A shows the Global X, Y, and Z axes, the local X, Y, and Z axes of the echolocation sensor (red, green, and blue arrows respectively), the 3D model of the echolocation sensor (in this case a model of the head of the bat Micronycteris *microtus* and the object to be ensonified, which in this case is a generic leaf with a model of a dragonfly. Panel B shows a still frame from a high-speed video showing a Micronycteris *microtus* performing the same task [25]).**

with a dragonfly. The world coordinate system $[X_w, Y_w, Z_w]$ is shown, to which the overall simulation is tied. For the remainder of this paper, we will omit the subscript W for brevity. The coordinate system of the sensor, $[X_s, Y_s, Z_s]$, is also shown, which is the overall coordinate system to which the sensor-specific components and behaviors are tied. This coordinate system can be placed into the environment and rotated around its axes using a 4×4 transformation matrix $T_{s,w}$ containing the rotation matrices and translation matrix. Panel B shows a still frame from a high-speed video of the same bat performing an approach to the leaf in a similar scenario. The goal of the simulation is to simulate the signals at the tympanum of the bat's ear during this approach phase, as these can then give insights into how the echolocation features the bat receives and might make use of to solve this hunting task.

III. IMPLEMENTATION DETAILS

The overall flowchart of the simulation can be found in figure 2. The simulation starts by importing scene geometry, encoded in the STL format [49]. Once the scene geometry is loaded, the mesh is analyzed, and minor reparations on the mesh are performed to make it watertight and ensure all the normals are pointing toward the outside of the mesh. Next, the local curvature of the mesh is calculated, which is used later in the simulation process to account for diffraction effects. Then, the simulation graph bifurcates into two parts: a part for raytracing and a part for diffraction. In the raytracing part, the solution for the Helmholtz equation is found using a ray acoustics approach, which assumes large Helmholtz numbers to be correct, from which the spatial impulse response is calculated. However, as in scenes similar to the one described in figure 1, the assumption of high Helmholtz numbers is not entirely correct, and therefore, diffraction effects cannot be neglected. To avoid solving the complete Helmholtz equation using FEM or BEM approaches, we resort to a Monte Carlo technique for approximating the diffraction echoes arising from the scene, using information about the local curvature of the mesh to locate diffraction echo candidates. This process calculates a spatial impulse response for the diffraction component.

Subsequently, as we only consider linear acoustic phenomena (a valid assumption for the sound pressure levels we consider in typical echolocation scenarios), the complete spatial impulse response can be obtained by summing the specular and diffraction spatial impulse responses. Then, as the echolocating agent under consideration does not have an isotropic emission and reception pattern, we apply a spatial ERTF (Echolocation Related Transfer Function, which is the product of the emission pattern and the HRTF) to the spatial impulse response using an array-based approximation approach, first described in the echolocation context in [20]. Finally, we convolve the ERTF-filtered spatial impulse responses with the echolocation call of the agent, and we obtain the signals at the left and right tympanum of the bat. In what follows, we will provide more details on each step in the computational graph.

A. MODEL PREPARATION

As stated before, the model mesh is stored in an STL file, which the simulation engine loads into a vertex matrix V_m and a face matrix F_m. Next, we perform several mesh cleaning functions to remove non-manifold faces, duplicate faces, and duplicate vertices. For this, we use the Matlab Lidar Toolbox [50], which, if wanted, other functions could replace to achieve the same goal. Next, we calculate the surface normals N_V for each vertex, which are then combined into the surface normals for each face N_F through averaging. This

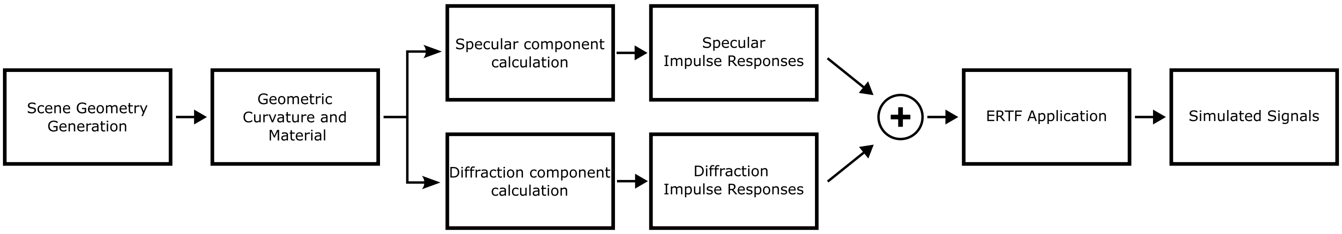

Scene Geometry Generation
Geometric Curvature and Material
Specular component calculation
Diffraction component calculation
Specular Impulse Responses
Diffraction Impulse Responses
+
ERTF Application
Simulated Signals

FIGURE 2. Block diagram illustrating the sequential stages of the simulation implementation for synthesizing acoustic signals.

information is then used to calculate the local curvature tensor C using the approach described in [51]. From this tensor, we calculate the magnitude of the curvature C_m per face. From this magnitude, we calculate two material properties of the scene mesh: the opening angle α of the acoustic BRDF (Bidirectional Reflectance Distribution Function) and the magnitude k (i.e., reflection strength) of the BRDF. These values (i.e., α and k) are frequency-dependent and must be set to appropriate values for the simulation to yield realistic results. It should be noted that one could perform experiments to obtain measured values for these BRDF functions, using approaches described in [52], [53], and [54], which, however, falls outside of the scope of this paper. The parameter $\alpha(v, f)$ encodes how wide of a directivity pattern the reflection from that vertex for that frequency is (i.e., how specular or omnidirectional that reflection is), and the parameter $k(v, f)$ encodes how strong the reflection of vertex v is. The resulting BRDF functions can be seen in figure 3. Panel A shows the opening angle α of the BRDF, and panel B shows the reflection strength k.

At this stage, SonoTraceLab supports a single emitter and multiple receivers, as this is a simulation setup which is applicable for many application scenarios, both in engineered sonar sensors [55], [56], [57], [58] as in biologically relevant echolocation setups, as both bats and dolphins can be approximated by a point-source with a known directivity [59]. The emitter and the receiver arrays are defined in the local sensor coordinate system $[X_s, Y_s, Z_s]$, as shown in figure 1. Figure 4, panel A) shows the coordinate system in more detail. In this case, the emitter is located at the origin of the sensor coordinate system, and two groups of receiver arrays are made, resembling a left and a right ear. Another layout can be seen in figure 7, where a 2D array is used for both ears, and the emitter is located below the origin. The whole sensor system can be located using a translation T and rotation R, which is combined into the transformation matrix $T_{s,w}$, and which is parameterized by the position of the sensor in world coordinates and the rotations $[\gamma_x, \gamma_y, \gamma_z]$ around the X, Y and Z axes respectively.

B. IMPULSE RESPONSE CALCULATION USING SPECULAR REFLECTIONS

In this section, we will discuss the raytracing process used to solve the specular part of the Helmholtz equation. The goal is to calculate the impulse response $h_i(t)$ for the i-th microphone from source s, propagated throughout the environment. To achieve this, we break the problem into a raytracing problem, where N rays are emitted into the environment (typical values for N are around 300.000 in our experiments). For each of the N rays, we calculate the propagation path throughout the environment using a specular reflection model for each model face. The overall process can be seen in figure 4, panels B1 through B4. It should be noted that while the figure shows a 2D view of the problem, the simulation works in 3D. Intersections of cast and reflected rays are calculated using a custom-made GPU implementation of the Moller-Trumbore algorithm [60], implemented in CUDA (for achieving faster computation) and linked to Matlab through CUDA-MEX. In the final panel (B4) of figure 4, two rays from the last reflection point are shown, one to each microphone. Each of these rays samples another value of the red BRDF function, yielding different reflection strengths and path lengths due to the geometric layout of the problem. The raytracing process yields a transfer function per ray, called $H(f, n, i)$, which encodes the complex transfer function between the source and the i-th microphone for the n-th ray for each frequency f. This transfer function is calculated as follows:

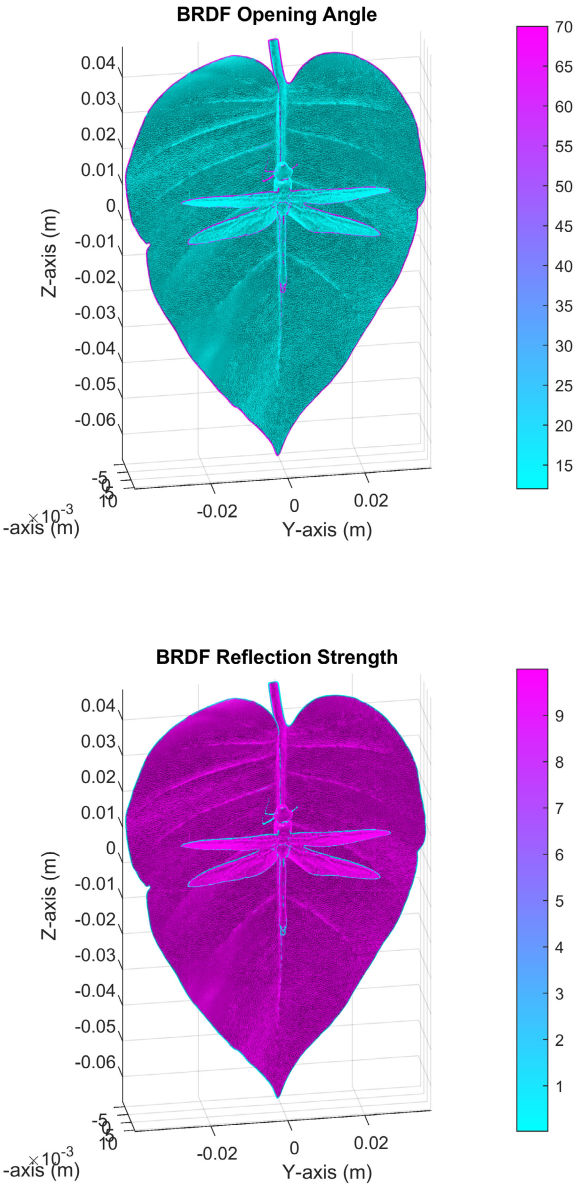
$$H(f, n, i) = H_m(f, n, i) \cdot e^{jk_w r_{i,n}} \cdot \frac{1}{r_{i,n}^2} \tag{1}$$

where H_m represents the magnitude of the reflection and is calculated from the BRDF functions of the reflections during the raytracing process, and the term $e^{jk_w r_{i,n}}$ represents the delay of the signal due to path propagation. The term $\frac{1}{r_{i,n}^2}$ represents the path loss of the signal due to spherical spreading. This complex transfer function $H(f, n, i)$ can then be transformed into the time domain using an inverse Fast Fourier Transform $\mathcal{F}^{-1}$:

$$h(t, n, i) = \mathcal{F}^{-1}\Big[H(f, n, i)\Big] \tag{2}$$

Finally, we can calculate the impulse response $h_i(t)$ for the i-th microphone between source s, propagated through the environment, by summing the N impulse responses:

$$h_i(t) = \sum_{n=1}^{N} h(t, n, i) \tag{3}$$


BRDF Opening Angle

BRDF Reflection Strength

FIGURE 3. **Overview of the acoustic BRDF (Bidirectional Reflectance Distribution Function) applied to the loaded mesh using local curvature to modulate the parameters α and k. The top panel shows the opening angle of the BRDF (i.e., how specular or omnidirectional a reflection on the local vertex is) and the reflectance strength (i.e., how strong a reflection is from this vertex). It should be noted that the opening angle towards areas with high curvature shows a significantly wider opening angle, which is desired because diffraction echoes arise at locations with high curvature and manifest themselves as almost omnidirectional reflections. However, these diffraction echoes should be significantly weaker than the specular reflected echoes, which are reflected in the BRDF reflection strength plot.**

Here, the impulse responses $h_i(t)$ contain the approximation to the solution of the Helmholtz equation under the assumption that the Helmholtz number is large, i.e., that specular reflections are the dominant reflection type occurring in the scene. It is important to understand that the impulse response $h_i(t)$ can be very complex, depending on the scene that is ensonified. Depending on the scene, and based on the number of rays used in the raytracing approach, $h_i(t)$ can contain thousands of non-zero elements, because the raytracing engine in SonoTraceLab allows multipath reflections to occur (up to a given reflection depth, typically set to 8). furthermore, using the right models, effects of weather conditions such as the impact of rain drops on the echo signals can be simulated as well using the raytracing approach.

C. IMPULSE RESPONSE CALCULATION USING DIFFRACTION

As stated in the previous section, raytracing-based approaches are only valid for large Helmholtz numbers. However, in many situations encountered in echolocation scenarios, diffraction echoes are a significant aspect of the overall impulse response. Nevertheless, solving the full Helmholtz equation is computationally untractable for typical scene geometries. Therefore, we resort to a Monte Carlo approximation of the wave equation solution and utilize the local curvature calculated over the scene geometry to find diffraction echo candidate locations. Concretely, we randomly distribute a fixed number of points onto mesh areas with high local curvature. We use an importance sampling approach [61] based on the cumulative probability distribution to distribute these points onto the mesh. Then, we follow a similar approach to impulse response generation as in the specular reflection case. First, we calculate the ranges $r_{i,m}$ between the i-th microphone and the m-th diffraction point, added to the distance between the diffraction point and source s. Then, we synthesize the transfer function $G(f, m, i)$ as follows:

$$G(f, m, i) = G_m(f, m, i) \cdot e^{jk_w r_{i,m}} \cdot \frac{1}{r_{i,m}^2} \tag{4}$$

Similarly to the specular reflections, G_m represents the magnitude of the reflections calculated from the BRDF angle and material property. The term $e^{jk_w r_{i,m}}$ represents the delay due to the distance between the diffraction point and the microphone, and $\frac{1}{r_{i,m}^2}$ represents the effect of path loss due to spherical spreading. This complex transfer function $G(f, m, i)$ can then be transformed into the time domain using an inverse Fast Fourier Transform $\mathcal{F}^{-1}$:

$$g(t, m, i) = \mathcal{F}^{-1}\Big[G(f, m, i)\Big] \tag{5}$$

Finally, we can calculate the impulse response $g_i(t)$ for the i-th microphone between source s, propagated through the environment, by summing the N impulse responses:

$$g_i(t) = \sum_{m=1}^{M} g(t, m, i) \tag{6}$$

The term $g_i(t)$ contains the partial solution to the Helmholtz equation for the diffraction components of the scene.

D. ERTF IMPLEMENTATION

The sources s and the receivers m_i are omnidirectional in nature. However, as most real-world sensors of echolocating

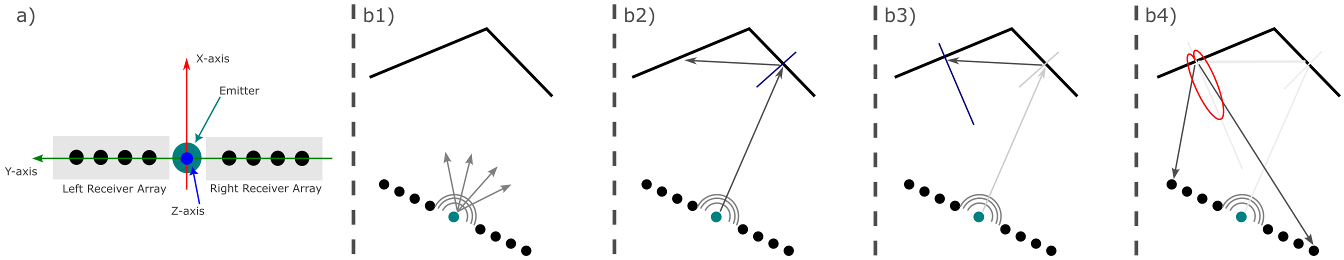


FIGURE 4. Panel a) shows the used coordinate system with two receiver arrays for the left and right ear, a single source, and the coordinate system attached to the sensor. Panel b shows an overview of the raytracing-based partial solution to the Helmholtz equation. In panel b1, a set of rays are cast into all directions of the frontal hemisphere. Panel b2 shows the intersection of one ray with the scene geometry, which is reflected around the surface normal. The ray further propagates (b3) and reflects from the scene geometry a final time (b4). The BRDF (red ellipse) is sampled by each microphone, depicted by two black lines intersecting the ellipse.

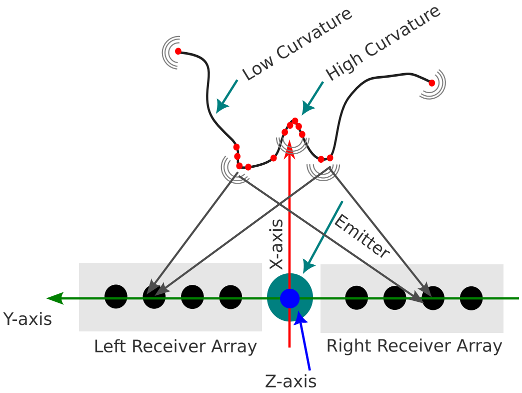


FIGURE 5. Overview of the approach of solving the diffraction aspects of the Helmholtz equation. We calculate the curvature using local differential geometry and sample locations with high curvature to create diffraction echoes. The generated samples then sample the local BRDF properties and provide input to the impulse response generation module.

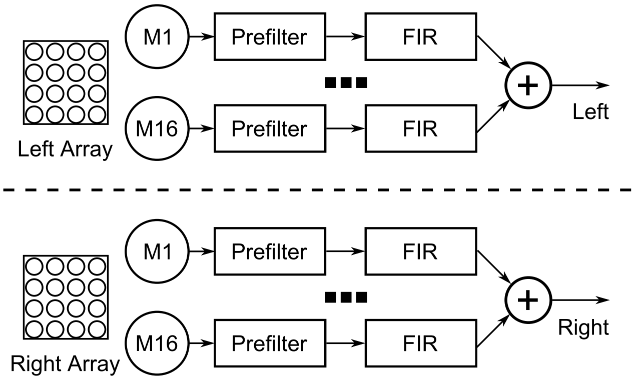


FIGURE 6. Overview of the array signal processing chain to implement the ERTF spatial filters. Each microphone is passed to an optional prefilter, followed by a FIR filter calculated using a least-squares model described in.

agents have non-ideal spatial directivity patterns, it is important that these spatial transfer functions are represented in SonoTraceLab. This is especially important for the simulations of biological echolocation systems, as the HRTF or ERTF are instrumental in the behavior of these animals [62], [63], [64]. However, tailor-made HRTF implementations are intractable in the simulation applied in SonoTraceLab. More importantly, it might be necessary to simulate array-based sensors that adjust their directivity patterns in post-processing or bats that alter their pinna shapes during echolocation. Therefore, we opted to apply an array-based approach to implementing ERTFs, similar to the approach we presented in [20]. In that work, we presented an ERTF fitting process to an arbitrary array of microphones using a least-squares approach and a far-field array response model. The overview of the signal processing process is shown in figure 6. For a given ERTF $E(f, \psi)$, parameterized by frequency f and direction ψ, a set of FIR filters $h_e(t, i)$ can be synthesized that, once applied to each microphone and summed, result in a single signal with the desired spatial directivity pattern. For example, for an arbitrary set of signals $s_i(t)$, the computation to apply an ERTF is as follows:

$$s_f(t) = \sum_{i=1}^{I} h_e(t, i) * s_i(t) \tag{7}$$

which yields a signal $s_f(t)$ representing the filtered signal at the inner ear.

E. ACOUSTIC SIGNAL GENERATION

At this point, we have all the necessary parts to implement the final step of the simulation, namely the generation of the signals at the tympanum of the bat. First, we need to generate the complete spatial impulse response of the scene $h_t(t, i)$ by combining the partial solutions obtained from the raytracing module and the diffraction module:

$$h_t(t, i) = a_r \cdot h_i(t) + a_d \cdot g_i(t) \tag{8}$$

with a_r and a_d gain factors for the ray-acoustics and the diffraction parts of the signals. Then, we filter these signals using the ERTF FIR filters to obtain a spatially filtered spatial impulse response, both for the left ear ($h_l(t, i)$) and the right ear ($h_r(t, i)$):

$$h_l(t) = \sum_{i=1}^{I} h_{e,l}(t, i) * h_t(t, i) \tag{9}$$

$$h_r(t) = \sum_{i=1}^{I} h_{e,r}(t, i) * h_t(t, i) \tag{10}$$

Now, with assuming a signal $s_e(t)$ emitted by the source s, we can calculate the signals arriving at the tympanum of the bat as follows:

$$s_l(t) = h_l(t) * s_e(t) \tag{11}$$

$$s_r(t) = h_r(t) * s_e(t) \tag{12}$$

With the generation of these signals, we have reached the end of the mathematical description of the simulation process, and we want to congratulate the reader on making it to this point in this paper. What will follow is a set of validation experiments showing the functionality of SonoTraceLab.

IV. VALIDATION

So far in this paper, we have shown how SonoTraceLab is implemented and how an approximation of the solution to the acoustic Helmholtz equation can be found in a computationally efficient manner. In this section, we will delve deeper into the practical applications of SonoTraceLab and present several examples of how SonoTraceLab can be used in biologically relevant echolocation scenes. Concretely, we will demonstrate three scenarios. In the first scenario, we will show how the ERTF implementation can yield sensor arrays with a spatial directivity function similar to the ERTF of a real bat. In the second scenario, we will show the spatial reflectivity function of the leaves of pitcher plants, both measured and simulated. Finally, we will show an example of the hunting behavior of Micronycteris *microtus*, approaching a leaf with a dragonfly.

A. SPATIAL FILTERING THROUGH ERTF FITTING

As discussed in the previous sections, we have implemented an array-based spatial filtering method that allows the implementation of virtually any ERTF in post-processing. To validate this approach, we simulated a scene with a single sphere of 5cm in front of a sensor as shown in figure 7, bottom left corner. The sphere was placed on 1000 points, distributed uniformly using an equal area sphere partitioning algorithm on the [65] frontal hemisphere with a radius of 1m. Then, we calculated the left and right impulse responses using the approach described in the previous section. Finally, we calculated the power spectral density of the reflected impulse response and plotted the so-obtained ERTFs. These results can be seen in figure 7. The figure shows the desired (i.e., simulated using a BEM based on micro-CT scans [64], [66]). These are shown for the bat Phyllostomus *discolor* and the bat Micronycteris *microtus*, for various frequency ranges, plotted using a Lambert Equal Area projection. We also plot the simulated responses from SonoTraceLab for the left and right ERTF in the same frequency ranges. It becomes apparent that these ERTFs show a significant degree of similarity, with similar degrees of correspondence as discussed in [20]. This experiment indicates two major functionalities of SonoTraceLab. Firstly, it indicated that the simulation approach for microphone arrays is correct and that time and phase differences of the reflected signals are correctly calculated. Secondly, it indicates that arbitrary ERTF functions can be implemented in SonoTraceLab, an important aspect for simulating the behavior of bats and dolphins.

B. REFLECTOR FILTERING OF BIOLOGICALLY RELEVANT REFLECTORS

As a second validation experiment, we set out to recreate the reflectivity function of the biologically inspired reflector shapes described in our previous work [37]. In that previous work, we proposed reflector shapes based on the shapes of the leaves of neo-tropical pitcher plants and used those as guiding beacons for robotic navigation. In that paper, we measured the reflectivity pattern of several 3D printed shapes, as shown in figure 8, left panel. The plot shows the impulse response both in the frequency domain as well as in the time domain for incidence angles ranging from -90° to 90° in steps of 1°. We simulated a similar setup in SonoTraceLab and calculated the same responses as in the real-world experiment. Here, we obtain very similar characteristics in both the time-domain representation of the impulse response and the frequency-domain response.

C. TARGET POSITION DISCRIMINATION USING BIOSONAR

As a final validation example, we have simulated the spatial reflectivity of a setup relevant to the hunting behavior of Micronycteris *microtus*. This bat hunts in the neotropical forests in Panama and regularly attacks silent and motionless prey [2], [11], [67]. In previous work, we have discovered the underlying mechanism that supports this behavior, namely the exploitation of the acoustic mirror effect [2], which supports easy prey presence discrimination. In that paper, we performed measurements of the reflectivity pattern of a leaf with and without a dragonfly. We found that the strongest difference between the reflectivity patterns can be found when approaching from oblique angles. These reflectivity patterns can be found in figure 10. Behavioral experiments showed that the bat is most likely to approach from these oblique angles, maximizing the information obtained from a few measurements. We recreated this setup in SonoTraceLab with a generic model of a leaf [68] and dragonfly [69] and performed similar reflectivity pattern measurements as in our previous work. From the resulting measurements, we observe similar patterns: a strong reflection when echolocating straight into the leaf and weaker reflections from oblique angles. Furthermore, the difference between the empty and full leaf is also the largest from oblique angles. It should be noted that the simulation is not an exact recreation of

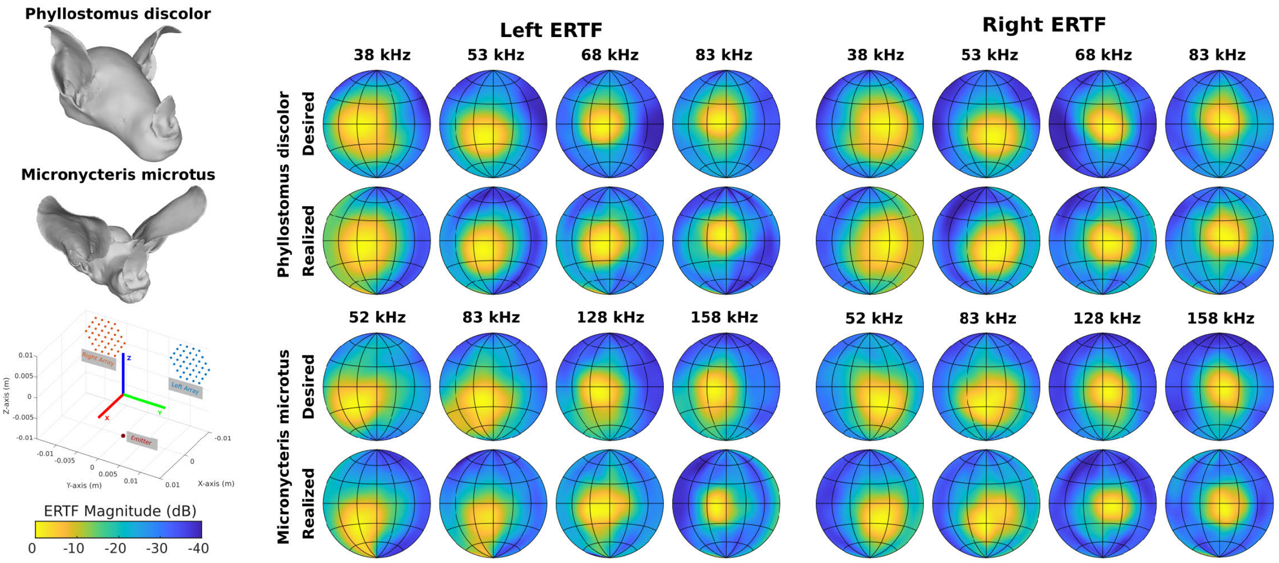


FIGURE 7. Overview of the validation experiment on fitting arbitrary ERTFs. This figure shows the results of fitting two ERTFs of two bats (Phyllostomus *discolor* and Micronycteris *microtus* on the microphone array shown in the bottom left of the figure. The top left shows the 3D models of the heads of these bats (not to scale). We show both the desired ERTFs (simulated using a Boundary Element Method), and the realized ones coming from the SonoTraceLab simulation.

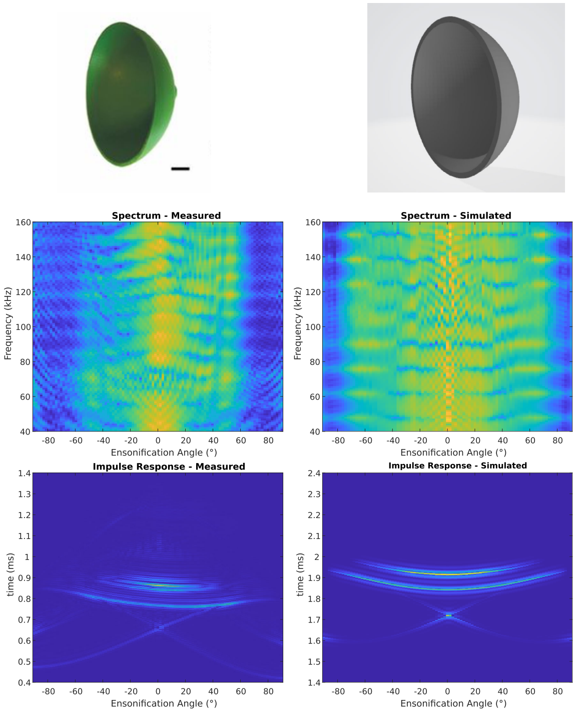


FIGURE 8. Spatial reflectivity patterns of a biologically inspired reflector shape, showing a distinct spatial response in the frequency domain (by means of stable notches over incidence angle). We both show the measured responses (left column) and the simulated responses using SonoTraceLab. Furthermore, we also show the time-domain impulse responses.

the measurements because the leaf shape and dragonfly shape and position are not matched exactly. Therefore, the qualitative comparison is applicable here, as they show that SonoTraceLab can recreate these biologically relevant cues based on 3D models.

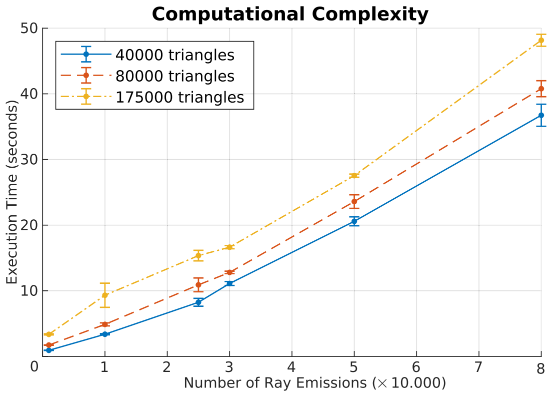


FIGURE 9. Test results for evaluating the computational complexity and scalability for the scene geometry mesh density and amount of emitted rays during raytracing. The simulation was executed ten times for each combination with the mean and standard deviations shown.

D. COMPUTATIONAL COMPLEXITY AND SCALABILITY

The main variables that impact the execution time of the simulation are the resolution and size of the scene geometry, defined in amount of mesh faces on the one hand, and the number of rays emitted into the environment on the other hand. To validate how the simulation execution time varies and scales across both variables, the simulation scenario described at the beginning of this section is repeated with different mesh resolutions and number of emitted rays. In that scenario the number of mesh faces was 82.485. In this test, this geometry was both halved and doubled in resolution. The

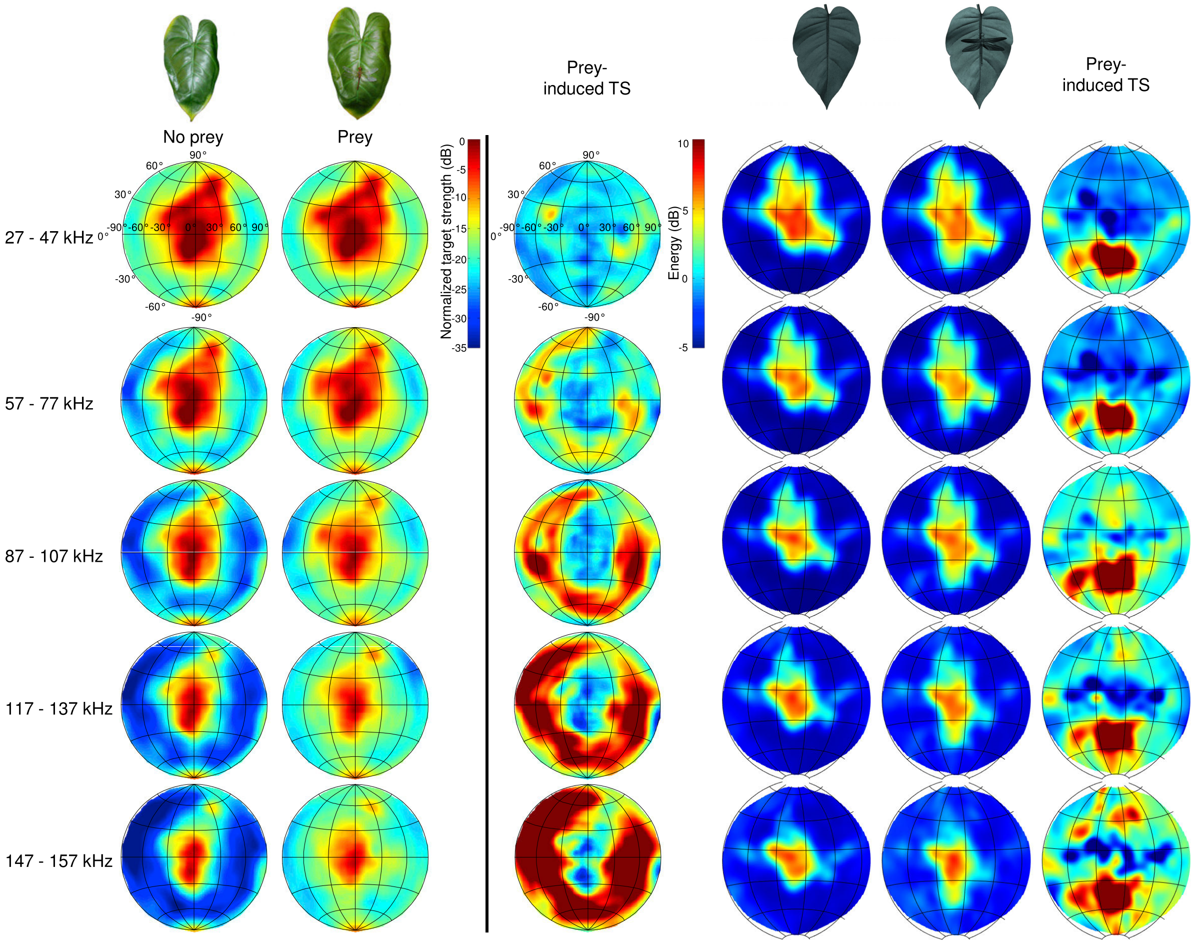


FIGURE 10. Overview of the validation experiment based on the hunting behavior of Micronycteris *microtus* when hunting silent and motionless prey. The measurements in the left column were taken from [2], which were performed to measure the spatial reflectivity pattern of leaves with and without prey. The difference between the full and empty leaf is most prominent from oblique angles (indicated by the "Prey-induced Target Strength" column). A simulation setup with similar characteristics (i.e., leaf with or without a dragonfly) was analyzed using SonoTraceLab, which results in reflectivity patterns with similar cues (i.e., larger differences from oblique angles). As these cues are biologically relevant for explaining the hunting behavior of M. *microtus*, this indicates the validity of SonoTraceLab for analyzing biological echolocation scenarios.

amount of ray emissions was scaled from 10.000 to 800.000. Each combination was executed ten times. The results are shown in figure 9. These were gathered on a laptop with an Intel i9-12900HK CPU and an NVIDIA RTX 3070 GPU. The experiments showed a linear increase in execution time for both variables. As such, we can assume the time complexity to be $\mathcal{O}(n \times m)$ with n and m describing the amount of mesh faces and ray emissions, respectively.

V. OPEN-SOURCE USAGE

In this section, we want to detail the open-source aspect of SonoTraceLab. We made the simulation package available through GitHub [70]. The source code repository can be found on Github under **https://cosyslab.app.link/sonotracelab** so researchers can further use and validate it in bioacoustics and 3D sonar sensor design. The source code comes with a few example scenarios similar to the experiments performed in this paper, which should allow the users of SonoTraceLab to be up and running efficiently.

Currently, SonoTraceLab is delivered solely as a Matlab API, lacking a graphical user interface (GUI). This design choice emphasizes programmability, allowing users to incorporate its functionalities into their existing Matlab workflows. This approach offers flexibility and customization options. We tried to build a high-level API that is easy to use, and abstracts away all the low-level details of mesh preparation, cleaning, raytracing and post-processing.

In terms of usage, the overall approach to performing a simulation is always the same. As a first step, SonoTraceLab loads and prepares the model mesh. This is done by providing a Matlab struct object, which contains the path to the mesh and some configuration parameters, such as its surface material reflection properties and the mesh pose, to the `prepareMeshSurface` function. It should be noted, a function represented as `functionName` is the name of the function in the SonoTraceLab API. Similarly, a Matlab struct must be made to describe the sensor and the emitter and receiver(s) coordinates. Lastly, a struct must be set up which defines other simulation parameters such as, for example, the directional-sampling density and amount of samples in the impulse response, as well as the speed of sound to be used in the simulation. These three struct objects are the arguments for the function `calculateImpulseResponseFast` to perform the specular and diffraction simulation steps. The output of this function will be a Matlab struct containing the impulse responses of all the receivers and for specular and diffraction separately, as well as information about the mesh's vertices that were responsible for the reflections and information such as individual reflection strengths. Within the GITHUB page of SonoTraceLab, examples can be found on how to use the API.

VI. CONCLUSION

In this paper, we presented SonoTraceLab, a simulation engine written in Matlab that allows the simulation of both technical as well as biological sonar sensor systems using straightforward object meshes. In order to solve the Helmholtz equation, we apply two sub-methods: raytracing for the specular reflections and a Monte Carlo approximation using importance sampling over mesh curvature for the diffraction echos. We provided details into the operation of SonoTraceLab, and explained how the simulator operates. Based on this discussion, we have shown various validation experiments showing the capability of SonoTraceLab to simulate the relevant biological cues present in real-world echolocation scenarios. Finally, we showed how SonoTraceLab can be obtained and used by other researchers through our open-source publication of SonoTraceLab.

As stated before, the simulations done with SonoTraceLab can recreate the biologically relevant cues that underpin several echolocation mechanisms found in bats. We believe that by using simulation engines like the one presented here, research in biological echolocation can be significantly accelerated, as many more experiments can be performed in a much shorter time-frame as compared to real-world measurements. Therefore, SonoTraceLab can serve as a hypothesis generation tool, which subsequently can be validated using real-world measurements and biological observations. In the introduction, we stated that a simulation engine such as SonoTraceLab can increase insights into biological systems while reducing time- and material complexity. For example, consider the measurements and simulations performed in figure 8. To perform measurements for these kind of reflectors, one needs to first build a measurement setup, model the reflector shapes, and manufacture them using a 3D printer. Next, measurements need to be performed, and the resulting data needs to be analyzed. From the experience of the authors, who have done multiple of such experiments in the past, this whole process would take about a work week to obtain the first measurements and analysis, with a few hours for each additional reflector that needs to be measured. Achieving the same result in SonoTraceLab took the authors about two hours, with each additional reflector taking a few minutes of computation time. Switching from one scenario to another is as complex as building the new mesh of the scene is. A person well versed in mesh manipulation can achieve this in a matter of minutes. Attention should be given to the mesh discretization (ie, surface triangles need to be uniform in size) and the surface normals should point in a uniform direction. But other than these small technical details, running novel scenarios is a matter of loading a new mesh model in SonoTraceLab.

Despite its applicability, several limitations remain in SonoTraceLab. Firstly, it should be noted that we only approximate the solution of the Helmholtz equation. Therefore, our simulation approach will never replace a real partial differential equation solver. However, as FEM models become computationally intractable quickly, using a system like SonoTraceLab is a valid tool for rapid iterations over various hypotheses. Secondly, SonoTraceLab deals poorly with time-varying mesh geometry, such as fluttering insects. While performing mesh deformations and recalculating the curvature is possible, it is time-consuming. Optimizations could be made in this domain and will be part of the future developments of SonoTraceLab. Furthermore, several more technical issues hinder actual large-scale raytracing simulations (with tens of millions of rays). Migration to other raytracing libraries, such as the NVIDIA OptiX library, would be an interesting step forward. This is also the subject of further developments on SonoTraceLab.

REFERENCES

[1] H.-U. Schnitzler, C. F. Moss, and A. Denzinger, "From spatial orientation to food acquisition in echolocating bats," *Trends Ecol. Evol.*, vol. 18, no. 8, pp. 386–394, Aug. 2003.

[2] I. Geipel, J. Steckel, M. Tschapka, D. Vanderelst, H.-U. Schnitzler, E. K. V. Kalko, H. Peremans, and R. Simon, "Bats actively use leaves as specular reflectors to detect acoustically camouflaged prey," *Current Biol.*, vol. 29, no. 16, pp. 2731–2736, Aug. 2019.

[3] Y. Prat and Y. Yovel, "Decision making in foraging bats," *Current Opinion Neurobiol.*, vol. 60, pp. 169–175, Feb. 2020.

[4] L. Stidsholt, S. Greif, H. R. Goerlitz, K. Beedholm, J. Macaulay, M. Johnson, and P. T. Madsen, "Hunting bats adjust their echolocation to receive weak prey echoes for clutter reduction," *Sci. Adv.*, vol. 7, no. 10, Mar. 2021, Art. no. eabf1367.

[5] K. A. McGowan and L. N. Kloepper, "Different as night and day: Wild bats modify echolocation in complex environments when visual cues are present," *Animal Behaviour*, vol. 168, pp. 1–6, Oct. 2020.

[6] R. Simon, M. W. Holderied, and O. von Helversen, "Size discrimination of hollow hemispheres by echolocation in a nectar feeding bat," *J. Experim. Biol.*, vol. 209, no. 18, pp. 3599–3609, Sep. 2006.

[7] R. Simon, M. W. Holderied, C. U. Koch, and O. von Helversen, "Floral acoustics: Conspicuous echoes of a dish-shaped leaf attract bat pollinators," *Science*, vol. 333, no. 6042, pp. 631–633, Jul. 2011.

[8] R. Simon, K. Bakunowski, A. E. Reyes-Vasques, M. Tschapka, M. Knörnschild, J. Steckel, and D. Stowell, "Acoustic traits of bat-pollinated flowers compared to flowers of other pollination syndromes and their echo-based classification using convolutional neural networks," *PLOS Comput. Biol.*, vol. 17, no. 12, Dec. 2021, Art. no. e1009706.

[9] M. G. Schöner, C. R. Schöner, R. Simon, T. U. Grafe, S. J. Puechmaille, L. L. Ji, and G. Kerth, "Bats are acoustically attracted to mutualistic carnivorous plants," *Current Biol.*, vol. 25, no. 14, pp. 1911–1916, Jul. 2015.

[10] I. Geipel, C. E. Kernan, A. S. Litterer, G. G. Carter, R. A. Page, and H. M. T. Hofstede, "Predation risks of signalling and searching: Bats prefer moving katydids," *Biol. Lett.*, vol. 16, no. 4, Apr. 2020, Art. no. 20190837.

[11] I. Geipel, K. Jung, and E. K. V. Kalko, "Perception of silent and motionless prey on vegetation by echolocation in the gleaning batMicronycteris microtis," *Proc. Roy. Soc. B: Biol. Sci.*, vol. 280, no. 1754, Mar. 2013, Art. no. 20122830.

[12] P. Stilz, "How glass fronts deceive bats," *Science*, vol. 357, no. 6355, pp. 977–978, Sep. 2017.

[13] S. Greif and B. M. Siemers, "Innate recognition of water bodies in echolocating bats," *Nature Commun.*, vol. 1, no. 1, p. 107, Nov. 2010.

[14] K. Übernickel, R. Simon, E. K. V. Kalko, and M. Tschapka, "Sensory challenges for trawling bats: Finding transient prey on water surfaces," *J. Acoust. Soc. Amer.*, vol. 139, no. 4, pp. 1914–1922, Apr. 2016.

[15] B. Fontaine and H. Peremans, "Compressive sensing: A strategy for fluttering target discrimination employed by bats emitting broadband calls," *J. Acoust. Soc. Amer.*, vol. 129, no. 2, pp. 1100–1110, Feb. 2011.

[16] C. F. Moss and M. Zagaeski, "Acoustic information available to bats using frequency- modulated sounds for the perception of insect prey," *J. Acoust. Soc. Amer.*, vol. 95, no. 5, pp. 2745–2756, May 1994.

[17] P. Caspers and R. Müller, "A design for a dynamic biomimetic sonarhead inspired by horseshoe bats," *Bioinspiration Biomimetics*, vol. 13, no. 4, Jun. 2018, Art. no. 046011.

[18] F. Schillebeeckx, F. De Mey, D. Vanderelst, and H. Peremans, "Biomimetic sonar: Binaural 3D localization using artificial bat pinnae," *Int. J. Robot. Res.*, vol. 30, no. 8, pp. 975–987, Jul. 2011.

[19] J. Sutlive, A. Singh, S. Zhang, and R. Müller, "A biomimetic soft robotic pinna for emulating dynamic reception behavior of horseshoe bats," *Bioinspiration Biomimetics*, vol. 16, no. 1, Jan. 2020, Art. no. 016016.

[20] J. Steckel and H. Peremans, "A novel biomimetic sonarhead using beamforming technology to mimic bat echolocation," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 59, no. 7, pp. 1369–1377, Jul. 2012.

[21] J. Steckel, F. Schillebeeckx, and H. Peremans, "Biomimetic sonar, outer ears versus arrays," in *Proc. IEEE SENSORS*, Limerick, Ireland, Oct. 2011, pp. 821–824.

[22] B. M. Siemers and H.-U. Schnitzler, "Echolocation signals reflect niche differentiation in five sympatric congeneric bat species," *Nature*, vol. 429, no. 6992, pp. 657–661, Jun. 2004.

[23] T. Nguyen, D. Vanderelst, and H. Peremans, "Sensorimotor behavior under informational constraints: A robotic model of prey localization in the bat," in *Proc. Conf. Artif. Life*, 2021, p. 61.

[24] S. Arimoto, M. Sekimoto, H. Hashiguchi, and R. Ozawa, "Natural resolution of ill-posedness of inverse kinematics for redundant robots: A challenge to Bernstein's degrees-of-freedom problem," *Adv. Robot.*, vol. 19, no. 4, pp. 401–434, Jan. 2005.

[25] *Micronycteris Microtis Capturing a Dragonfly*, Cosys-Lab, Nov. 2018. [Online]. Available: https://www.youtube.com/watch?v=aJdEFgdTpl4

[26] A. D. Pierce, *Acoustics: An Introduction to Its Physical Principles and Applications*. Cham, Switzerland: Springer, 2019.

[27] G. C. Everstine, "Finite element formulatons of structural acoustics problems," *Comput. Struct.*, vol. 65, no. 3, pp. 307–321, Nov. 1997.

[28] F. Ihlenburg, *Finite Element Analysis of Acoustic Scattering*. Cham, Switzerland: Springer, 1998.

[29] L. L. Thompson, "A review of finite-element methods for time-harmonic acoustics," *J. Acoust. Soc. Amer.*, vol. 119, no. 3, pp. 1315–1330, Mar. 2006.

[30] S. Kirkup, *The Boundary Element Method in Acoustics*. Lancashire, U.K.: Integrated Sound Software, 2007.

[31] Kirkup, "The boundary element method in acoustics: A survey," *Appl. Sci.*, vol. 9, no. 8, p. 1642, Apr. 2019.

[32] K. Onaka, M. Morise, and T. Nishiura, "A design of 3-dimensional sound field simulator based on acoustic ray tracing and HRTF," in *Proc. IEEE 13th Int. Symp. Consum. Electron.*, May 2009, pp. 233–236.

[33] N. Röber, S. Andres, and M. Masuch, *HRTF Simulations Through Acoustic Raytracing*. Halle, Germany: University and State Library of Saxony-Anhalt, 2006.

[34] N. Röber, U. Kaminski, and M. Masuch, "Ray acoustics using computer graphics technology," in *Proc. 10th Int. Conf. Digit. Audio Effects*, 2007, pp. 117–124.

[35] COMSOL. *COMSOL: Multiphysics Software for Optimizing Designs*. Accessed: May 15, 2024. [Online]. Available: https://www.comsol.com/

[36] *Siemens NX Acoustics*. Accessed: May 15, 2024. [Online]. Available: https://plm.sw.siemens.com/en-U.S./simcenter/simulation-test/acoustic-simulation/

[37] R. Simon, S. Rupitsch, M. Baumann, H. Wu, H. Peremans, and J. Steckel, "Bioinspired sonar reflectors as guiding beacons for autonomous navigation," *Proc. Nat. Acad. Sci. USA*, vol. 117, no. 3, pp. 1367–1374, Jan. 2020.

[38] B. E. Treeby and B. T. Cox, "K-wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields," *J. Biomed. Opt.*, vol. 15, no. 2, 2010, Art. no. 021314.

[39] F.-I. Field-II. *Field II Ultrasound Simulation Program*. Accessed: May 15, 2024. [Online]. Available: https://field-ii.dk/

[40] D. Bæk, J. A. Jensen, and M. Willatzen, "Modeling transducer impulse responses for predicting calibrated pressure pulses with the ultrasound simulation program field II," *J. Acoust. Soc. Amer.*, vol. 127, no. 5, pp. 2825–2835, May 2010.

[41] J. A. Jensen, "Simulation of advanced ultrasound systems using field II," in *Proc. 2nd IEEE Int. Symp. Biomed. Imag., Macro Nano*, Jul. 2004, pp. 636–639.

[42] M. Sharifzadeh, H. Benali, and H. Rivaz, "An ultra-fast method for simulation of realistic ultrasound images," in *Proc. IEEE Int. Ultrason. Symp. (IUS)*, Sep. 2021, pp. 1–4.

[43] Y. Zhou, "Fast algorithm for simulation of signals in medical ultrasound blood flow imaging," Ph.D. dissertation, Institutt for Elektronikk OG Telekommunikasjon, Dept. Electron. Telecommun., Norwegian Univ. Sci. Technol., 2012.

[44] A. Cigier, F. Varray, and D. Garcia, "SIMUS: An open-source simulator for medical ultrasound imaging. Part II: Comparison with four simulators," *Comput. Methods Programs Biomed.*, vol. 220, Jun. 2022, Art. no. 106774.

[45] X. Cao, Y. Cai, and X. Cui, "A parallel numerical acoustic simulation on a GPU using an edge-based smoothed finite element method," *Adv. Eng. Softw.*, vol. 148, Nov. 2020, Art. no. 102835.

[46] P. Alexander, A. Duncan, and N. Bose, "Modelling sound propagation under ice using the ocean acoustics library's acoustic toolbox," *Proc. Acoust. Soc. Aust.*, 2012. [Online]. Available: https://espace.curtin.edu.au/handle/20.500.11937/8743

[47] N. Morozs, W. Gorma, B. T. Henson, L. Shen, P. D. Mitchell, and Y. V. Zakharov, "Channel modeling for underwater acoustic network simulation," *IEEE Access*, vol. 8, pp. 136151–136175, 2020.

[48] M. B. Porter, "The bellhop manual and user's guide: Preliminary draft," *Heat, Light, Sound Res., Inc.*, vol. 260, 2011.

[49] *STL Files Explained Learn About the STL File Format*. Accessed: May 15, 2024. [Online]. Available: https://www.adobe.com/au/creativecloud/file-types/image/vector/stl-file.html

[50] L. T. Matlab. *Remove Surface Mesh Defects—MATLAB RemoveDefects—MathWorks Benelux*. MathWorks Benelux. Accessed: May 15, 2024. [Online]. Available: https://nl.mathworks.com/help/lidar/ref/surfacemesh.removedefects.html

[51] S. Rusinkiewicz, "Estimating curvatures and their derivatives on triangle meshes," in *Proc. 2nd Int. Symp. 3D Data Process., Visualizat. Transmiss.*, 2004, pp. 486–493.

[52] W. Binek, A. Pilch, and T. Kamisinski, "Application of fitted to measurements sound reflecion model in geometrical acoustics simulations," in *Proc. Inter-Noise Noise-Con Congr.*, vol. 259, 2019, pp. 1189–1194.

[53] J. Durany, T. Mateos, and A. Garriga, "Analytical computation of acoustic bidirectional reflectance distribution functions," *Open J. Acoust.*, vol. 5, no. 4, pp. 207–217, 2015.

[54] W. Matusik, H. Pfister, M. Brand, and L. McMillan, "Efficient isotropic BRDF measurement," Ph.D. dissertation, 2003. [Online]. Available: https://dash.harvard.edu/handle/1/4238987

[55] R. Kerstens, D. Laurijssen, and J. Steckel, "ERTIS: A fully embedded real time 3D imaging sonar sensor for robotic applications," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 1438–1443.

[56] R. Kerstens, D. Laurijssen, G. Schouten, and J. Steckel, "3D point cloud data acquisition using a synchronized in-air imaging sonar sensor network," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 5855–5861.

[57] T. Verellen, R. Kerstens, D. Laurijssen, and J. Steckel, "Urtis: A small 3D imaging sonar sensor for robotic applications," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 4801–4805.

[58] N. Balemans et al., "S2L-SLAM: Sensor fusion driven SLAM using sonar, LiDAR and deep neural networks," in *Proc. IEEE Sensors*. IEEE, 2021.

[59] J. A. Thomas, C. F. Moss, and M. Vater, *Echolocation in Bats and Dolphins*. London, England: University of Chicago press, 2004.

[60] V. Shumskiy, "GPU ray tracing—Comparative study on ray-triangle intersection algorithms," in *Transactions on Computational Science*. Cham, Switzerland: Springer, 2013, pp. 78–91.

[61] M. Corsini, P. Cignoni, and R. Scopigno, "Efficient and flexible sampling with blue noise properties of triangular meshes," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 6, pp. 914–924, Jun. 2012.

[62] F. De Mey, F. Schillebeeckx, D. Vanderelst, A. Boen, and H. Peremans, "Modelling simultaneous echo waveform reconstruction and localization in bats," *Biosystems*, vol. 100, no. 2, pp. 94–100, May 2010.

[63] M. K. Obrist, M. B. Fenton, J. L. Eger, and P. A. Schlegel, "What ears do for bats: A comparative study of pinna sound pressure transformation in chiroptera," *J. Experim. Biol.*, vol. 180, no. 1, pp. 119–152, Jul. 1993.

[64] D. Vanderelst, F. De Mey, H. Peremans, I. Geipel, E. Kalko, and U. Firzlaff, "What noseleaves do for FM bats depends on their degree of sensorial specialization," *PLoS ONE*, vol. 5, no. 8, Aug. 2010, Art. no. e11893.

[65] P. Leopardi, "A partition of the unit sphere into regions of equal area and small diameter," *Electron. Trans. Numer. Anal.*, vol. 25, no. 12, pp. 309–327, 2006.

[66] F. De Mey, J. Reijniers, H. Peremans, M. Otani, and U. Firzlaff, "Simulated head related transfer function of the phyllostomid bat phyllostomus discolor," *J. Acoust. Soc. Amer.*, vol. 124, no. 4, pp. 2123–2132, Oct. 2008.

[67] S. E. Santana, I. Geipel, E. R. Dumont, M. B. Kalka, and E. K. V. Kalko, "All you can eat: High performance capacity and plasticity in the common big-eared bat, micronycteris microtis (chiroptera: Phyllostomidae)," *PLoS ONE*, vol. 6, no. 12, Dec. 2011, Art. no. e28584.

[68] D. Liu. (2019). *Leaf 3D Model*. [Online]. Available: https://skfb.ly/6SuSN

[69] FFISH.Asia. (2023). *Anax Parthenope 3D Model*. [Online]. Available: https://skfb.ly/oJNSP

[70] C.-l. Github. (2024). *Cosys-Lab*. [Online]. Available: https://github.com/Cosys-Lab/SonoTraceLab

WOUTER JANSEN (Graduate Student Member, IEEE) received the M.Sc. degree in electronics and ICT engineering from the University of Antwerp, Belgium, in 2019, and the Ph.D. degree. He joined the Cosys-Lab Research Group, University of Antwerp. His current research interests include further advancements in the application of state-of-the art in-air sonar sensors for industrial use cases by developing tools and techniques for learning semantics with this sensing modality.

JAN STECKEL received the Dr.-Ing. degree in electronic engineering from the Karel de Grote University College, Hoboken, in 2007, and the Ph.D. degree from the Active Perception Laboratory, University of Antwerp, in 2012. His Ph.D. dissertation titled "Array processing for in-air sonar systems -drawing inspirations from biology." During this period, he developed state-of-the art sonar sensors, both biomimetic and sensor-array based. During the Ph.D. period, he was an Active Member with the Centre for Care Technology, University of Antwerp, where he was in charge of various healthcare-related projects concerning novel sensor technologies. Furthermore, he pursued industrial exploitation of the patented 3D array sonar sensor, which was developed in collaboration during the Ph.D. degree. In 2015, he became a Research Professor with the Co-Design of Sensors and Systems (Cosys-Lab), University of Antwerp, where he researches sensors, sensor arrays, and signal processing algorithms using an embedded, constrained systems approach. He is also a member with Flanders Make, the network for innovation in the manufacturing industry in Belgium.