JASA ARTICLE

Indoors audio classification with structure image method for simulating multi-room acoustics[a)]

Erez Shalev,[1,b)] Israel Cohen,[1] and Dmitri Lvov[2]
[1]*Andrew and Erna Viterby Faculty of Electrical and Computer Engineering, Technion – Israel Institute of Technology, Technion City, Haifa 3200003, Israel*
[2]*DSP Group Inc., 3 Arik Einstein Street, Herzelia 4659071, Israel*

ABSTRACT:
In this paper, we introduce an extension of the image method for generating room impulse responses in a structure with more than a single confined space, namely, the structure image method (StIM). The proposed method, StIM, can efficiently generate a large number of environmental examples for a structure impulse response, which is required by current deep-learning methods for many tasks, while maintaining low computational complexity. We address the integration of the environment representation, produced by StIM, into the training process, and present a framework for training deep models. We demonstrate the usage of StIM when training an audio classification model and testing with real recordings acquired by accessible day-to-day devices. StIM shows promising results for indoors audio classification, where the target sound source is not located in the same room as the microphones. StIM enables large scale simulations of multi-room acoustics with low computational complexity which is mostly beneficial for training of deep learning networks. © *2021 Acoustical Society of America*. https://doi.org/10.1121/10.0006781

(Received 4 January 2021; revised 25 August 2021; accepted 25 August 2021; published online 25 October 2021)
[Editor: Peter Gerstoft] Pages: 3059–3073

I. INTRODUCTION

With the advancement of machine learning methods and their impressive performance on classification tasks, and in audio signal processing tasks,[1] the task of classifying the source within an audio segment receives a high focus within the research community. This task is called sound event classification (SEC). Such capability has the potential to vastly improve applications such as smart cities,[2,3] smart houses,[4] robotics (surveyed by Young[5]), security and surveillance systems,[6–8] alarm systems for protection of infants, children, and the elderly community[9] and more. Due to recent advancements in deep and machine learning, SEC has reached impressive performances in the last decade. The second task in Detection and Classification of Acoustic Scenes and Events (DCASE2016) challenge[10] offered a dataset dedicated to SEC. Bilen *et al.*[11] explored a robust method for the evaluation of the task. Many forms of deep neural networks (DNN) were researched for the task of SEC. Cakir *et al.*,[12] Young *et al.*[13] and Adavanne *et al.*,[14] amongst others, experimented with their own DNN for polyphonic sound signals, be it a feed-forward DNN, a convolutional neural network (CNN), or recurrent neural network (RNN). Adavanne *et al.* also combined convolutional-recurrent neural network[15] (CRNN) as a baseline model for the third task in DCASE2019.[16] Specifically for indoors audio classification, Bai *et al.*[17] explored audio enhancement using beamformers as a mean to improve classification performances. Such an approach aims at eliminating reverberations, background interference, and noises.

Structural reverberations and environmental factors indeed pose a difficult challenge to current methods.[1] Though signal enhancement methods such as Mingsian *et al.*[17] are usually beneficial as a back-end process to any front-end task, they are not enough to overcome this challenge, as they are often case specific and do not offer robustness in every environment. The available datasets do not offer good representation for environmental variety. The Audioset[18] dataset is composed of audio taken from online videos, and as such have inherent environmental influence within them, which cannot be separated and controlled for research purposes. The DCASE2019 dataset[16] alone contains five different locations, obtained by recording and restoring a room impulse response (RIR). However, the dataset samples are already convolved with these RIRs and therefore, are also inherent within the dataset. Moreover, all RIRs are present in both the training and evaluation sets and so the dataset cannot establish robustness to environments. DNN methods performances are dependent on datasets which cover the distribution of the data they learn in order to generalize. This means that for any new class we wish to represent, not only do we have to record a sufficient number of examples to represent that class, but also a high number of examples of that class, in a sufficient number of locations, are required, which makes the problem of dataset acquisition much more challenging.

For indoors classification problems, the coverage of the environment examples can be achieved by using simulation methods. A clean dataset, obtained in a recording studio

[a)]This paper is part of a special issue on Machine Learning in Acoustics.
[b)]Electronic mail: erezsh@campus.technion.ac.il

with high performance equipment and minimum reverberations, can be convolved with a number of simulated RIRs, to cover the environmental representation. However, in order to represent the distribution of environmental influence, multiple examples are needed. The simulator should be capable of describing the use-case environments, preferably with low computational complexity.

Room structure simulation methods are divided into three major disciplines, namely, statistical methods, wave-based methods, and geometrical methods, sometimes called ray-based. Statistical methods, such as statistical energy analysis (SEA),[19,20] are not concerned with the temporal dynamic of a sound source and are used for steady state information. Such methods are more suitable for constant noises (such as an idling engine) and are problematic when addressing audio events of a short duration. In the SEC task, we anticipate signals which can be short time, such as speech, or human distress, such as falling or a baby's cry. Wave-based methods, such as the finite element method,[21,22] boundary element method,[23–25] and finite-difference time-domain,[26] represent the acoustic of a structure by solving the wave equation. Such methods are highly descriptive in terms of structure, although the solution is computationally demanding. Moreover, for some of these methods which are grid-based (such as the aforementioned wave-based methods), the computational complexity associated with these methods gets even higher with relation to the modeled frequency. Accordingly, these methods are more suitable for lower frequencies. Therefore, they are limited when a large number of examples is required. While wave-based methods can be compatible for classic algorithms, most of them are too slow for DNN, where a high number of examples is required. One could generate such a large number of examples once and publish the results for the use of others. However, there are many different parameters for such a simulation, and a dataset covering all the possible combinations will be huge. The benefit of a fast simulation is the ability to experiment with different ranges and combinations of parameters without the need in a huge dataset. Geometrical, or ray-based methods, use rays or particles which are reflected at the surface of the room, to model the travelling sound wave. Ray-tracing[27] is an example of a ray-based method, which is primarily focused on the signal's energy, rather than the pressure wave.[28] The computational complexity of such methods is dependent on the reflection order and not on the frequency. Thus, these methods are faster than the aforementioned wave-based methods for modeling higher frequency ranges. A commonly used simulation method is the image method (IM) of Allen and Berkley.[29] The IM is shown by Allen and Berkley to be a solution to the wave equation, given the source and receiver locations inside a rectangular enclosure with rigid walls. This approach is based on modeling waves reflected from walls by image sources in free field. IM provides this solution with a relatively low computational complexity, typical to ray-based methods, even when modeling higher frequencies. However, IM is limited to a single-space square-shaped rooms.

Due to the low computational complexity and the representation of short time signals, IM is currently the most suitable method for simulating many examples for a time dynamic signal source, although it is limited to a single confined space. A highly efficient implementation of this method was published by Habets.[30] Borish[31] introduced an expansion of IM to non-square rooms. Voländer[32] offered a ray-tracing combination, to obtain longer RIRs, with low computational complexity. Incident-angle dependency was researched by Rindel,[33] while Lam[34] studied adjustments for frequency dependent representation. When considering any indoors audio processing task, the source can sometimes be located in another room from the sensor. For example, a system which uses audio ques to detect aggression[35] can be utilized indoors to help fight child, women, and elder abuse. Smart home systems[4] could be installed with fewer microphones. Assistance robots[36] and surveillance robots,[37] which represent a non-stationary receiver, are likely to be located in another room from the source. However, none of the former adaptations to IM has yet addressed more than a single confided space.

In this work we introduce StIM, a structure image method, aimed at extending[29] into coupled rooms simulations. We explore and define a framework for using simulated RIRs during the training of a DNN SEC classifier, and we assess performances of models trained with either the IM,[30] the proposed StIM, or without any environmental representation.

The rest of this work is organized as follows. Section II formulates the problem. Section III overviews highlights of the original image methods fundamentals required for the rest of the discussion. Section IV describes the principles of StIM. Section V explores the experimental results and shows the improvements achieved by using an RIR simulation methods. The section continues to explore the benefits of using StIM for a sound located in another room, for simulation or for real samples. Last, we conclude with a discussion in Sec. VI.

II. PROBLEM FORMULATION

Let $y(t)$ be an audio segment of length T, recorded at the receiver. We can write $y(t)$ as

$$y(t) = s(t) * h(t) + w(t), \tag{1}$$

where $s(t)$ is the target source, * is the convolution operation, $h(t)$ represents the system through which the signal is deformed on its way to the receiver, and $w(t)$ is white Gaussian noise (WGN).We note that $w(t)$ has an important role in the integration of $h(t)$ into the dataset. We used a WGN signal, but an argument could be made for $w(t)$ not necessarily being white or stationary. We leave the research of other possible $w(t)$ for further study. We focus this work on $h(t)$ representing a sound segment which travels between two rooms. Given a closed dictionary of classes $c \in \mathcal{C}$, we wish to classify the sound in $y(t)$ using a classification model $\hat{M}$ s.t. $\hat{c} = \hat{M}\{y(t)\}$. The accuracy performance of a model

https://doi.org/10.1121/10.0006781

P, for a test set of size N, is simply the percent of correct classifications

$$P = 100\frac{\sum_{n=1}^{N}\mathbb{I}_{c=\hat{c}}[n]}{N}, \tag{2}$$

where n is the sample index, $\mathbb{I}$ is the indicator function which equals 1 if $\hat{c} = c$, and 0 otherwise. Such an accuracy measure is simple and informative for our balanced dataset. Looking at the results of challenges such as DCASE[38,39] or ESC-50,[40] the majority and usually the most successful methods are DNN. Before the emergence of deep learning, other classical methods were used for audio classification tasks such as hidden Markov model,[41] Gaussian mixture model,[42] non-negative matrix factorization,[43] and support vector machines.[44] These methods achieve non-compatible results when compared with DNN models as shown by DCASE and ESC-50. Therefore, we limit our model discussion to DNN models.

III. HIGHLIGHTS ON ORIGINAL IMAGE METHOD

Consider a source $s(t)$ which is modeled as a singular point emitting waves of a spherical wave-front in all directions (similarly to a light source). These waves are reflected off surfaces and reverberate from the room's interior surfaces. The reverberations are represented as delayed, attenuated replicas, of $s(t)$, and are summed together with the direct path. Given a room architecture and information on the source and receiver locations, the result is an RIR, $h(t)$, which can be convolved with any source $s(t)$ to simulate the reverberations. Note that this resulting RIR is specific to the source and receiver locations within the room. Such RIR represents the information on reverberations by adequately attenuating each delayed replica with respect to time-decay, the order of reflections, and the absorption of the room's surfaces.

The original IM is simply a way of tracing the routes, that is, calculating the delay and the reflections with low complexity. We follow an efficient implementation of the IM by Habets,[30] and keep notations wherever possible. Figure 1 illustrates a low-order imaging, using the IM on a 2-D case, which is scalable for the 3-D case. Figure 1(a) represents the imaging (blue asterisk) of the original source (green asterisk) on the y axis, x axis, and a second order imaging on both x and y axes. A first order reflection path is represented by the orange line, whereas a second order reflection is represented by the green line. The first order reflection is demonstrated to create an isosceles triangle with base angles on the original and virtual (imaged) sources, and its obtuse angle on the left wall, where the dashed and solid lines meet. Due to the isosceles triangle, the path from the virtual source to the receiver is the same length as the path of the original source, when it is reflected off of the left wall. Given the length of the path d and the known speed of sound C, the delay can be calculated by

$$\tau = \frac{d}{C}. \tag{3}$$

Using this calculated delay, the decay of the signal due to the distance, can be accounted for by

$$h_v(t) = \frac{\delta(t-\tau)}{4\pi d}, \tag{4}$$

where $h_v(t)$ represents the impulse response of such a virtual source. The wall's reflection coefficient also has to be accounted for, as some of the sound is absorbed by the wall. This will be discussed later and is omitted in Eq. (4).

The imaging on the x axis creates an additional isosceles triangle on the lower wall, which is not traced in the

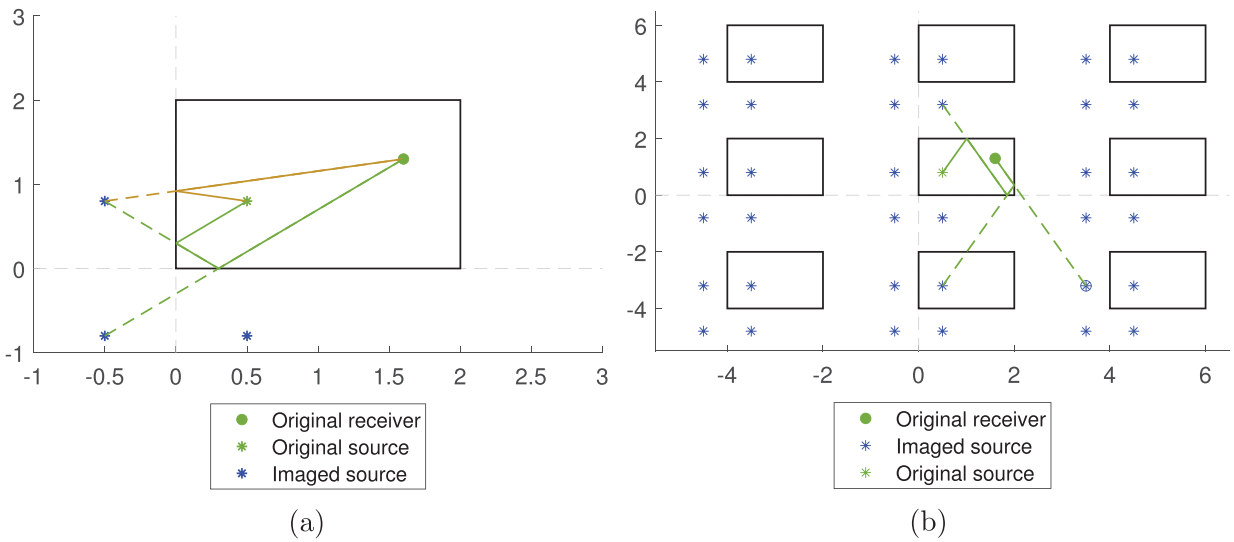

FIG. 1. (Color online) Original image method's illustration in 2-D case. The room's walls and the duplicates are marked in black rectangles. (a) Low order imaging using $p \in \mathcal{P} = (q, j, k)$. (b) High order imaging using both $m \in \mathcal{M} = (m_x, m_y, m_z)$ and $p \in \mathcal{P} = (q, j, k)$.

illustration. The imaging on both axes represents a second order reflection on both axes, where the absorption of both the lower and the left walls, has to be accounted for. Isosceles triangles can be traced for higher order and are illustrated in Fig. 1(b) using the green lines. The circled blue asterisks iindicate the imaged-source being traced, and the solid lines represents the real path inside the room.

For a general 3-D source located at $\vec{s} = (x_s, y_s, z_s)$ we define the imaging vector $\vec{p} = (q, j, k)$. We follow Habets,[30] and represent the imaging using $R_p = (x_p, y_p, z_p)$ as

$$R_p = (x_s(1-2q), y_s(1-2j), z_s(1-2k)), \tag{5}$$

where $q, j, k \in \{0, 1\}$. For example, a 3-D replication on two axes x and y which is not replicated on z, will be represented by setting $(q, j, k) = (1, 1, 0)$. Hence, the set of vectors $p \in \mathcal{P}$ controls the low order reflections (up to three walls).

Let the room dimensions be defined by $L = (L_x, L_y, L_z)$. The original IM now proceeds and allocates replicas of this room, where the lower left corner of the room is allocated to all combination of $R_m = (2m_xL_x, 2m_yL_y, 2m_zL_z)$ for each replica. Here, $m_x, m_y, m_z \in \mathbb{N}$ are in some range $-M < m_i < M$. These replicas of the original room and the sources (both original source and images) represent higher order reflections. The path traced in Fig.1(b) represents a path for a virtual source, where $M = 1$. The figure illustrates the original source (green asterisk), the receiver (green circle), un-traced virtual sources (blue asterisks), and the traced virtual source (circled blue asterisks). Here, the parameters for the traced virtual source are $(q, j) = (1, 0)$ and $(m_x, m_y) = (1, -1)$. A similar example for such illustration can be found in *Room Acoustics*, Sec. 4.1, by Kuttruff.[45]

Finally, given the receiver location (x_r, y_r, z_r), the path length d between the virtual source and the receiver is calculated by

$$\begin{aligned} R_p &= [x_s(1-2q) - x_r,\ y_s(1-2j) - y_r,\ z_s(1-2k) - z_r], \\ R_m &= [2m_xL_x, 2m_yL_y, 2m_zL_z], \\ d &= ||R_p + R_m||, \end{aligned} \tag{6}$$

which we can plug into Eq. (3) and account for the decay. The number of replications M of the room is either set by calculation or with respect to a user input value. When calculated, the constraint for enough replications is that the longest delay time T_{60} is the time interval in which the decay level drops down by 60 dB. This is commonly known as T_{60} (*Room Acoustics*, Sec. 3.8, by Kuttruff[45]).

The reflection order of a source, located at distance $d = ||R_P + R_m||$ with respect to the receiver, is given by

$$O_{p,m} = |2m_x - q| + |2m_y - j| + |2m_z - k|, \tag{7}$$

and the full impulse response is given by

$$h(\vec{s}, \vec{r}, t) = \sum_{p\in\mathcal{P}} \sum_{m\in\mathcal{M}} \beta_1^{|m_x-q|} \beta_2^{|m_x|} \beta_3^{|m_y-j|} \beta_4^{|m_y|} \times \beta_5^{|m_z-k|} \beta_6^{|m_z|} \frac{\delta(t-\tau)}{4\pi d}. \tag{8}$$

The attenuation effect, due to each single reflection from a face i, is accounted for by multiplying once with the respective reflection coefficient, β_i.

IV. STRUCTURE IMAGE METHOD

We wish to modify the original method to allow a source which is located in an adjacent room. For simplicity, we first consider a source that is located in a void outside a shoe-box room. Only after analyzing this case, we proceed to analyze the influence of an adjacent room.

A. Allocation of a source outside a room

We start by considering the original IM and the efficient implementation by Habets,[30] when the source is outside the room. Figure 2(a) first raises a problem with some low order reflections. The original source (green asterisk) has a direct path which goes through the wall. Relying on *Fundamentals of Acoustics*, Sec. 6.2, by Kinsler,[46] we consider the wall as a thin transition layer. This assumption means that the wave has the same direction of movement after transitioning through the wall. The assumption is frequency dependent, as is the original IM, and holds for frequencies below the limit $c_w/2L_w$, where c_w is the speed of sound within the wall's material and L_w is wall's thickness. The difference is that the signal, now, has to be multiplied by a transfer coefficient, rather than a reflection coefficient, and the wall acts as a low-pass filter, with a cutoff frequency f_c, of $f_c = c_w/2L_w$, attenuating high frequencies.

The direct path is a viable path that has to be multiplied by the respective coefficient. However, when considering the virtual imaged sources, we can see in the example shown in Fig. 2(a) that only the imaging on y axis (green plus) has a viable reflection from the bottom wall. This virtual source needs to be multiplied with both the transition coefficient of the left wall, and the reflection coefficient of the bottom wall. The virtual source inside the room (red $\times$) has no wall in its path to be reflected from the real source, hence, there is no such path through which the source can travel to the receiver. The imaging on the x axis (also a red $\times$) uses the virtual source inside the room in the path reflection analysis, which practically does not exist. Therefore, this source also does not have a viable path. This problem is intensified when the original source is located in the lower left corner (green plus). In this case, only the direct path is a viable path, while all three images do not represent a real reflection path.

This problem is further complicated when we scale up the reflection order. Let us assume that we could define a method for resolving which of the low order reflections are viable when given a source location, thus eliminating the low order difficulties. The next step in IM would be to simply duplicate the room while only considering these viable images. Figure 2(b) illustrates the validity of paths with higher order reflections. The original source (green asterisk) is located with relatively close proximity to the left wall, with respect to the room dimensions. All the blue asterisks

https://doi.org/10.1121/10.0006781

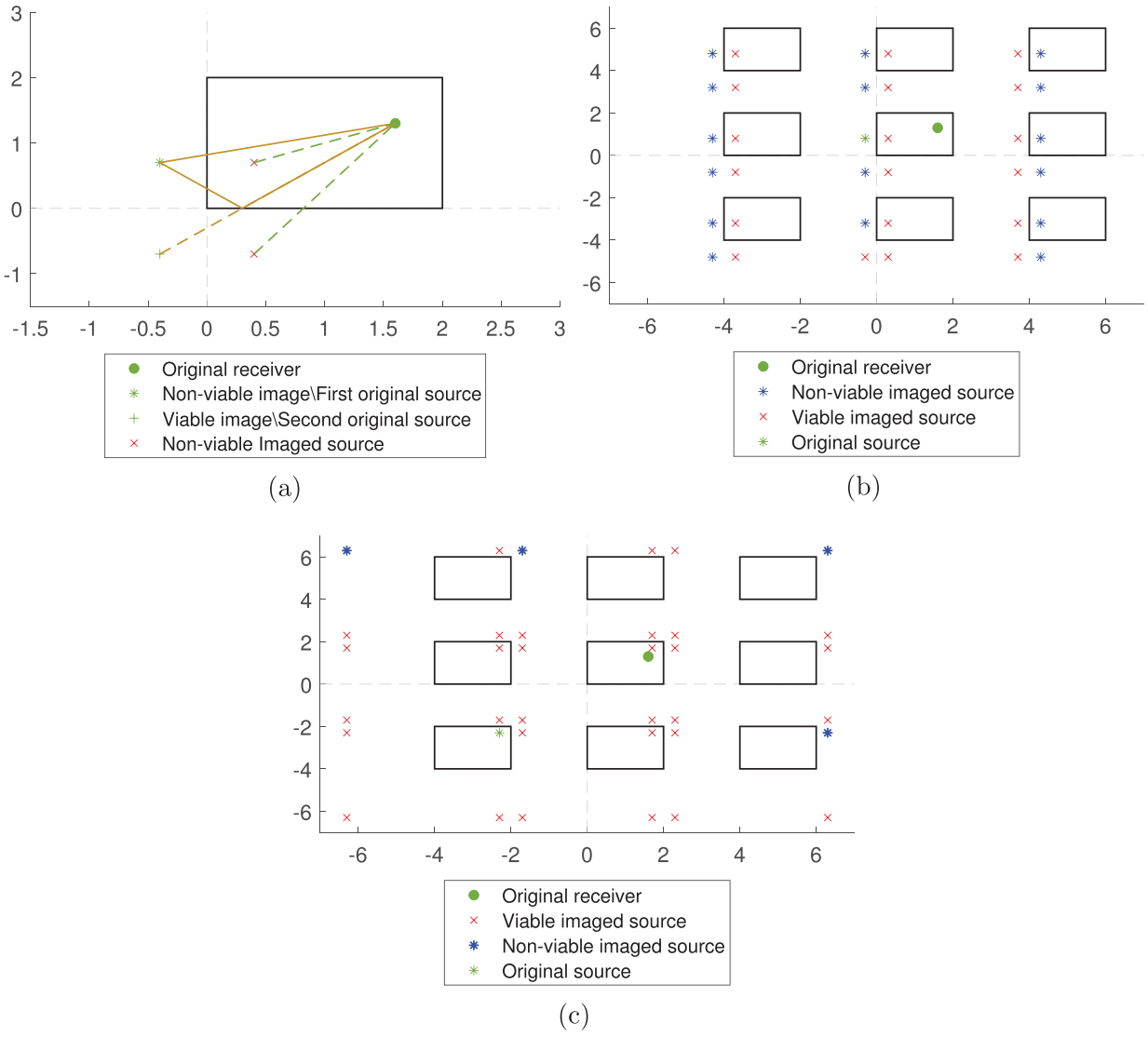

FIG. 2. (Color online) Original image method's 2-D illustration for a source outside a shoe-box room.(a) Low order imaging of a source located outside the shoe-box room. Two examples are given where the original source is different. (b) High order imaging of a source located outside a shoe-box room with relatively close proximity and with direct path to a single wall. (c) High order imaging of a source located outside a shoe-box room, relatively far from the room walls, with respect to the room's dimensions and with direct paths to two walls.

represent virtual sources with viable paths, while the virtual sources indicated by a red $\times$ do not have viable paths. It is clear from the scattering of the viable sources that there is no original pattern of a low order case, which we can replicate, in order to scale up. Moreover, there is no apparent relation between the parameters q, j, k, m_x, m_y, and m_z to the validity of the path, and no apparent order can be found, even in terms of the drawing [note the virtual source at $(q,j,m_x,m_y) = (0,1,0,-1)$]. In other words, there is no way to translate this back to an efficient algorithm, which would drop paths that are not viable and take into account only viable ones.

The comfortable example of a source located with relatively close proximity to the wall is an easier case. A more difficult case to analyse, would be of a source outside the room, which is located at a very far point from the wall, or at a point that has direct paths to two walls, meaning, waves can enter the room through each wall. These two possibilities further complicate the allocation of a source outside the room with the existing method. We have yet to attend to the consideration of multiplying the high order reflections, by the correct coefficients with respect to the walls. Figure 2(c) illustrates a source where two walls are in the source's direct path, and its distance from the wall is greater than any one dimension of the room. The original source is represented by a green asterisk, virtual sources without a viable path are represented by red $\times$ symbols and blue asterisks represent virtual sources with viable paths. Figure 2(c) further emphasises the difficulty of translating the existing IM into a fast and efficient algorithm for a source outside a shoe-box room. This difficulty poses a limitation for generating a large number of RIRs with low computational complexity. Such limitations induce the mentioned challenges of training a DNN model on a large number of simulated RIRs, which

represent sources arriving from different rooms, thus, improving the classification's robustness to the environmental factors.

B. Receiver imaging

We first suggest a minor alteration to the original image method. Instead of imaging the source, we choose to image the receiver. Imaging of the receiver and the source is demonstrated in Fig. 3. The original source (green asterisk) is imaged on the y axis to create a virtual source (blue asterisk). The original receiver (green circle) is also imaged on the y axis and creates a virtual receiver (blue circle). The lines crossing from the imaged source to the real receiver, and from the imaged receiver to the real source, are obviously equal. Note that these lines create two isosceles triangles connected by their obtuse angles. This minor alteration may be achieved by modifying Eq. (6) to

$$\begin{aligned} R_p &= [x_r(1-2q)-x_s,\, y_r(1-2j)-y_s,\, z_r(1-2k)-z_s], \\ R_m &= [2m_xL_x, 2m_yL_y, 2m_zL_z], \\ d &= ||R_p + R_m||. \end{aligned} \tag{9}$$

We have generated two RIRs using the original IM and the receiver IM. The first pick, which is the loudest and represents the direct path, had a real mean square of 0.05. A comparison of the two showed a maximal error of $\sim 10^{-15}$ order of magnitude, which could be assigned to a numerical error. Hence, we can say that the methods are identical. Figure 4 compares an RIR generated by the receiver image method and a real RIR measured in a real room with objects and some furniture. Figure 4(a) depicts the room of sizes $(L_x, L_y, L_z) = (2.55, 3.3, 2.5)$ and a source and receiver located at $(x_s, y_s, z_s) = (1.25, 2.7, 0.55)$ and $(x_r, y_r, z_r) = (0.95, 0.3, 1.2)$, respectively. The real RIR was measured using a sine sweep.[47] It is clear from the measurement, that both the original and receiver image methods are not an accurate representation of a real environment. The receiver

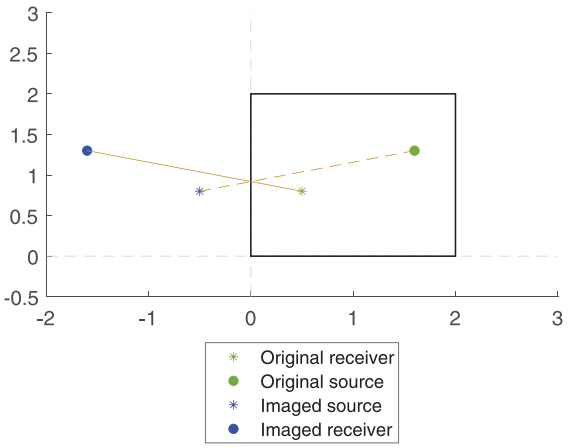

FIG. 3. (Color online) Imaging of both the source and receiver on the y axis.

image method is capable of generating an RIR that is identical to the original image method, given a small alteration. This small alteration has no cost for the computation complexity over the original method and is very easy to implement.

C. Allocation of a source outside a room with receiver imaging

Now let us analyze a source allocated in a void outside of a shoe-box room using the receiver image method. It is easy enough to skip the low order case and move directly to a high order reflection example in Fig. 5. All problems presented in Sec. IV A for the original IM are reduced, and all that is left is to determine whether the line of sight between the source and the virtual receiver goes through the original room or not. This is demonstrated in Fig. 5(a). All blue circles receivers and their paths (green dotted lines) go through the original room [the room located at (0, 0)], while all other receivers are red squares (paths in orange dotted lines). This is also true for the low order case which we skipped.

Figure 5(b) shows a path tracing of one such virtual receiver. Note that even though the method drops the irrelevant receivers, it still uses them in order to trace the paths of relevant receivers.

D. Intersection with walls for a given virtual source

Given a virtual receiver located at $\vec{v} = (x_v, y_v, z_v)$ and a source at $\vec{s} = (x_s, y_s, z_s)$, we wish to find if the room is located on the direct line between the points, and through which wall the line penetrates the room. The line between the points is given by

$$l(u) = \vec{s} - u(\vec{v} - \vec{s}) \tag{10}$$

in a parametric form using the parameter u. We constrain the location of the room to $0 \le u \le 1$, so that it is not located on the infinite line, inside the segment, between the two points.

Let a shoe-box room of size (L_x, L_y, L_z) be located at the origin. We calculate the parameter u for which the line intersects with each wall, and we infer the 3-D point of the intersection (x, y, z). Considering the wall on the x axis, for example, we first require the points to be on opposite sides of the wall. Then, we also require $0 \le y \le L_y$ and $0 \le z \le L_z$, to ensure that the point is within the wall's limits. All intersections with respect to the faces of the room are pre-calculated in Table I. Walls that are parallel to an axis received either the value 0 in the table when they are on the axis or L_i if they are located on L_i, with respect to the axis i.

At this point, for implementation efficiency, we define $\bar{u} = u|(x_v - x_s)(y_v - y_s)(z_v - z_s)|$ to eliminate the denominators. We pre-calculate the following 13 helper variables for the conditioning phase:

https://doi.org/10.1121/10.0006781

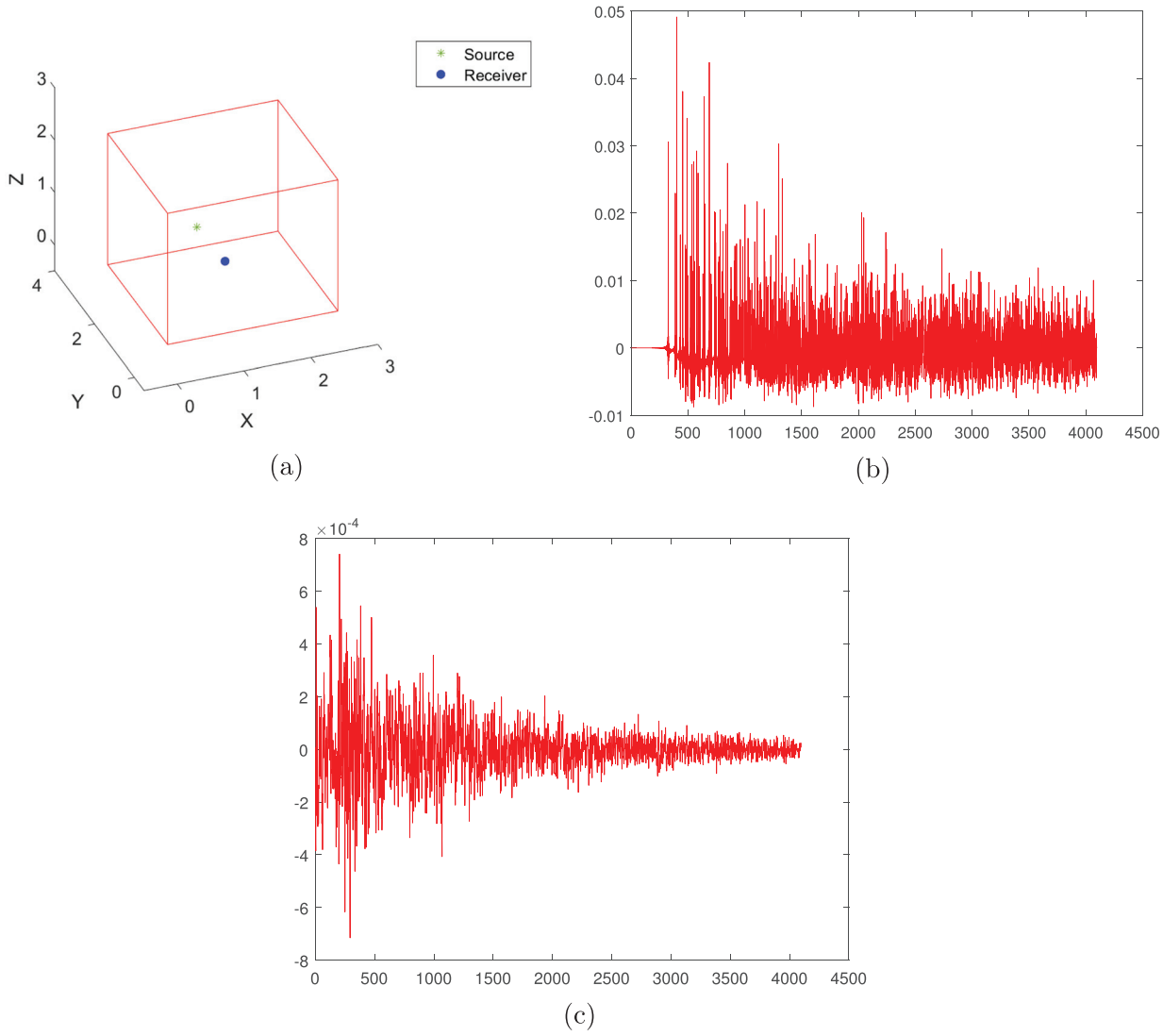

FIG. 4. (Color online) Receiver image method vs a measured image method. (a) Room setup, (b) receiver image method, (c) measured image method.

$$
\begin{aligned}
s_{xy} &= (x_v - x_s)L_y,\\
s_{xz} &= (x_v - x_s)L_z,\\
s_{yx} &= (y_v - y_s)L_x,\\
s_{yz} &= (y_v - y_s)L_z,\\
s_{zx} &= (z_v - z_s)L_x,\\
s_{zy} &= (z_v - z_s)L_y,
\end{aligned}
\tag{11a}
$$

$$
\begin{aligned}
a_{xyz} &= |(x_v - x_s)(y_v - y_s)(z_v - z_s)|,\\
a_{xy} &= |(x_v - x_s)(y_v - y_s)|,\\
a_{xz} &= |(x_v - x_s)(z_v - z_s)|,\\
a_{yz} &= |(y_v - y_s)(z_v - z_s)|,
\end{aligned}
\tag{11b}
$$

$$
\begin{aligned}
c_{xy} &= x_s y_v - x_v y_s,\\
c_{xz} &= x_s z_v - x_v z_s,\\
c_{yz} &= y_s z_v - y_v z_s.
\end{aligned}
\tag{11c}
$$

The conditions for each face of the room are summarized in Table II. Note that the condition cannot be met for both face 1 and face 2, and the same goes for any two parallel faces. In terms of efficiency, in the worst case scenario we go through 12 conditions with 4 additional comparisons per axis. Algorithm 1 utilizes the sources, the pre-calculated variables in Eq. (11), the virtual receiver locations, the room dimensions, and the conditions from Table II. Using these, algorithm 1 can determine if the wave between the receiver and the source crosses parallel faces respectively to an axis, and returns the face which is crossed first. Given the room's dimensions and the locations of both source and virtual receiver, algorithm 2 uses algorithm 1 to determine the face that is first crossed. An output of 0 from algorithm 2 represents that the wave does not go through the original room at all (not a viable path).

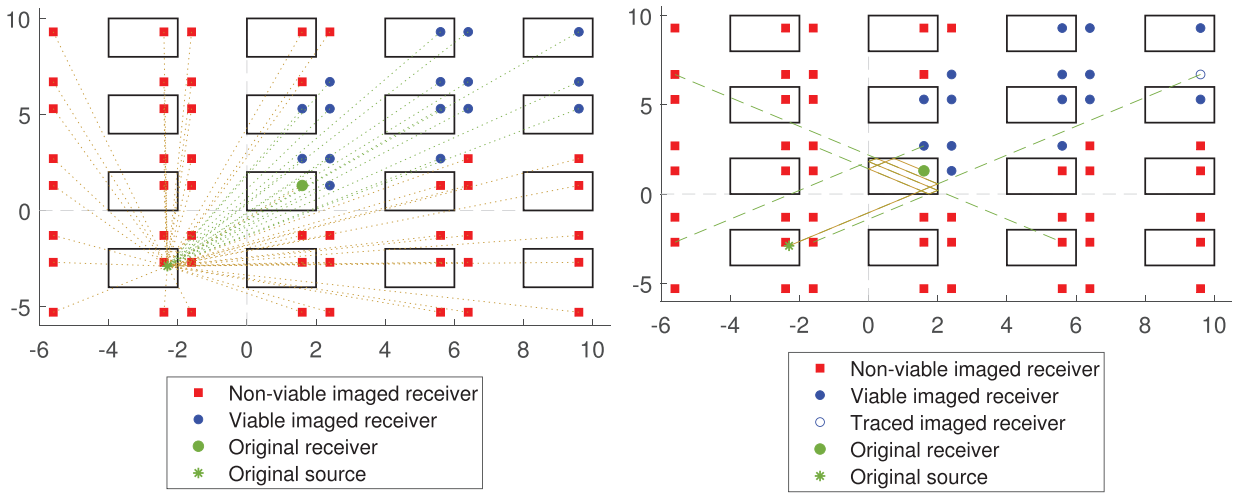

FIG. 5. (Color online) High order receiver imaging with a source outside a shoe-box room. (a) High order mapping of viable sources using waves. (b) Illustration of a reflection path for the virtual imaged source at $(m_x, m_y) = (2, 2)$ and $(q, j) = (0, 1)$. The real path in solid orange and the tracing in shredded green line.

ALGORITHM 1. Determines penetration through original room, with respect to 1-D.

Input: $a, L, x_s, x_v, face, face\_num, face\_u\_bar$
Output: $face\_num, face\_u\_bar$
  if $x_s < 0 < x_v$ **then**
    $u\_bar \leftarrow -x_s a$
    if $u\_bar < face\_u\_bar$ **then**
      if first and second conditions of $face$ **then**
        $face\_u\_bar \leftarrow u\_bar$
        $face \leftarrow face + 1$
      end if
    end if
  else if $x_s < L < x_v$ **then**
    $u\_bar \leftarrow (x_s - L)a$
    if $u\_bar < face\_u\_bar$ **then**
    if first and second conditions of $face + 1$ **then**
        $face\_u\_bar \leftarrow u\_bar$
        $face\_num \leftarrow face + 1$
      end if
    end if
  end if

ALGORITHM 2. Determines through which of the faces does the wave between the source and receiver penetrate (if any).

Input: L, s, v
Output: $face\_num$
    Initialize:
    calculate all 13 variables
    $face\_num \leftarrow 0$
    $face\_u\_bar \leftarrow a_{xyz}$
    $face\_num, face\_u\_bar \leftarrow Alg1(a_{yz}, L_x, x_s, x_v, 1, face\_u\_bar)$
    $face\_num, face\_u\_bar \leftarrow Alg1(a_{xz}, L_y, y_s, y_v, 3, face\_u\_bar)$
    $face\_num, face\_u\_bar \leftarrow Alg1(a_{xy}, L_z, z_s, z_v, 5, face\_u\_bar)$

E. StIM

We now have to take into account the fact that the source is also located within a room. Let the receiver room be defined by $L_1 = (L_{x,r}, L_{y,r}, L_{z,r})$ and the source room be defined by $L_2 = (L_{x,s}, L_{y,s}, L_{z,s})$. Let the source and receiver be located at $\vec{s} = (x_s, y_s, z_s)$ and $\vec{r} = (x_r, y_r, z_r)$, respectively. Figure 6 shows such a scenario in 2-D, where the source's room is replicated. The original source is indicated by green

TABLE I. Intersection of the parameterized line $l(u)$ with each of the room's faces.

Face	u	x	y	z
1	$\frac{x_v}{x_v - x_s}$	0	$\frac{x_v y_s - x_s y_v}{x_v - x_s}$	$\frac{x_v z_s - x_s z_v}{x_v - x_s}$
2	$\frac{x_v - L_x}{x_v - x_s}$	L_x	$\frac{(L_x - x_s)y_v - (L_x - x_v)y_s}{x_v - x_s}$	$\frac{(L_x - x_s)z_v - (L_x - x_v)z_s}{x_v - x_s}$
3	$\frac{y_v}{y_v - y_s}$	$\frac{x_s y_v - x_v y_s}{y_v - y_s}$	0	$\frac{z_s y_v - z_v y_s}{y_v - y_s}$
4	$\frac{y_v - L_y}{y_v - y_s}$	$\frac{(L_y - y_s)x_v - (L_y - y_v)x_s}{y_v - y_s}$	L_y	$\frac{(L_y - y_s)z_v - (L_y - y_v)z_s}{y_v - y_s}$
5	$\frac{z_v}{z_v - z_s}$	$\frac{x_s z_v - x_v z_s}{z_v - z_s}$	$\frac{y_s z_v - y_v z_s}{z_v - z_s}$	0
6	$\frac{z_v - L_z}{z_v - z_s}$	$\frac{(L_z - z_s)x_v - (L_z - z_v)x_s}{z_v - z_s}$	$\frac{(L_z - z_s)y_v - (L_z - z_v)y_s}{z_v - z_s}$	L_z

https://doi.org/10.1121/10.0006781

TABLE II. Face intersection conditions.

Face	$\bar{u}$	Points on both sides of the face	Face limits first dimension	Face limits second dimension
1	$-x_v a_{yz}$	$x_v < 0 < x_s$	$0 \le c_{xy} \le -s_{xy}$	$0 \le c_{xz} \le -s_{xz}$
2	$(x_v - L_x)a_{yz}$	$x_v < L_x < x_s$	$s_{yx} - s_{xy} \le c_{xy} \le s_{yx}$	$s_{zx} - s_{xz} \le c_{xz} \le s_{zx}$
3	$-y_v a_{xz}$	$y_v < 0 < y_s$	$s_{yx} \le c_{xy} \le 0$	$0 \le c_{yz} \le -s_{yz}$
4	$(y_v - L_y)a_{xz}$	$y_v < L_y < y_s$	$-s_{xy} \le c_{xy} \le s_{yx} - s_{xy}$	$s_{zy} - s_{yz} \le c_{yz} \le s_{zy}$
5	$-z_v a_{xy}$	$z_v < 0 < z_s$	$s_{zx} \le c_{xz} \le 0$	$s_{zy} \le c_{yz} \le 0$
6	$(z_v - L_z)a_{xy}$	$z_v < L_z < z_s$	$-s_{xz} \le c_{xz} \le s_{zx} - s_{xz}$	$-s_{yz} \le c_{yz} \le s_{zy} - s_{yz}$

asterisks. The receiver's room is not replicated, and the original receiver is indicated by a green circle. First, we simply apply the receiver image method using the original source location. Next, we continue by applying the original IM to the source's room, in order to generate virtual source locations. Similar to the problem presented in Sec. IV A, we need to decide which of the virtual sources has a viable path. Mathematically, this is the same problem of finding out which virtual receiver is viable, when the original source is outside the room, only now, the receiver is the one outside the room. Hence, the solution is also similar to Sec. IV C, with the exception that here we stay loyal to the imaging of the sources. For each virtual source, we can use algorithm 1 and algorithm 2 to determine if the path connecting the virtual source and the original receiver goes through the original source room. If it does, we apply the receiver image method on the receiver room, using that virtual source's location (blue asterisks in Fig. 6). Otherwise, we omit the virtual source as the path is not viable (red × symbols in Fig. 6).

We are now left with the last consideration of adjusting the reflection attenuation, to account for the wall transitions. In the implementation by Habets,[30] the impulse response for such virtual source s, given the receiver location r was already introduced by Eq. (8). We now adjust these equations for the case of StIM.

We note that for adjacent rooms, the waves can only be transferred through the joint wall. This can happen in two ways as presented in Fig. 6 with the light blue lines. The wave either penetrates through the cross section between the joint wall, in which case the transition coefficient is with respect to that wall, or it travels out of the source's room and into the receiver's room through another wall. Let us split the impulse response into two factors h_1 and h_2, where h_1 contains all the waves penetrating the inner wall and h_2 contains the waves that travel out of the source's room. For the h_2 case, we have to use algorithms 1 and 2 to find the face of the receiver's room, through which the wave enters. The transition coefficient is then the multiplication of the transition coefficient for that face, and the joint face. Higher order cases, of a wave which travels more than once between the rooms, exist as well. We do not handle these cases in this paper, as they are usually attenuated below 60 dB. To adjust for the attenuation of a joint room impulse response, we gather the reflection coefficients from the source room $\beta_{i,s}$ and from the receiver room $\beta_{i,r}$. We denote the transition and reflection coefficients of the face through which the wave exits the source room and enters the receiver room $\bar{\beta}_s$, $\bar{\alpha}_s$, $\bar{\beta}_r$, and $\bar{\alpha}_r$, where β represents reflection and α represents transition. The impulse response $h_1(\vec{s},\vec{r},t)$ for the first case, for a virtual source s given $p_s \in \mathcal{P}_s = (q_s, j_s, k_s)$, $m_s \in \mathcal{M}_s = (m_{x,s}, m_{y,s}, m_{z,s})$ and a receiver r given $p_r \in \mathcal{P}_r = (q_r, j_r, k_r)$, $m_r \in \mathcal{M}_r = (m_{x,r}, m_{y,r}, m_{z,r})$ can be written by

$$\begin{aligned} &h_1(\vec{s},\vec{r},t) \\ &= \frac{\bar{\alpha}_s}{\bar{\beta}_s \bar{\beta}_r} \sum_{p_s \in \mathcal{P}_s} \sum_{m_s \in \mathcal{M}_s} \sum_{p_r \in \mathcal{P}_r} \sum_{m_r \in \mathcal{M}_r} \\ &\times \left[\beta_{1,s}^{|m_{x,s}-q_s|} \beta_{2,s}^{|m_{x,s}|} \beta_{3,s}^{|m_{y,s}-j_s|} \beta_{4,s}^{|m_{y,s}|} \beta_{5,s}^{|m_{z,s}-k_s|} \beta_{6,s}^{|m_{z,s}|}\right] \\ &\times \left[\beta_{1,s}^{|m_{x,r}-q_r|} \beta_{2,s}^{|m_{x,r}|} \beta_{3,s}^{|m_{y,r}-j_r|} \beta_{4,s}^{|m_{y,r}|} \beta_{5,s}^{|m_{z,r}-k_r|} \beta_{6,s}^{|m_{z,r}|}\right] \frac{\delta(t-\tau)}{4\pi d}. \end{aligned} \tag{12}$$

Here, we note that $\bar{\beta}_s = \bar{\beta}_r$. We simply divide by both to omit a single reflection for each method, as it becomes a transition. We then represent the transition by multiplying by the respective transition coefficient. For the second case, the impulse response $h_2(\vec{s},\vec{r},t)$, we are only required to account for the additional multiplication by the transition coefficient of the receiver room's face through which the wave enters. Hence, the impulse response can be written by

$$h_2(\vec{s},\vec{r},\tau) = \bar{\alpha}_r h_1(\vec{s},\vec{r},\tau). \tag{13}$$

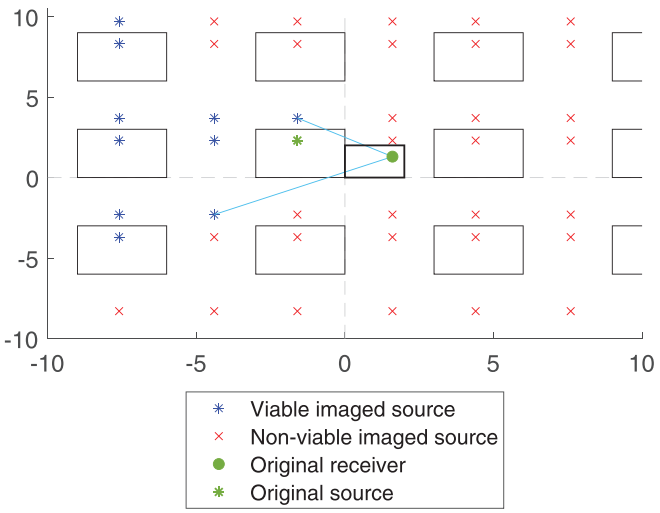

FIG. 6. (Color online) Structure image method illustration.

This complete analysis is dubbed the structure image method (StIM). StIM can produce a structural RIR with a relatively small overhead, while maintaining the capability of producing an inside room RIR using the receiver image method, without any overheads. It enables the generation of a large number of RIRs, describing sources arriving from an adjacent room with a low computational complexity. StIM enables the training and evaluation of DNN models for classification of sound, with a higher diversity of environmental representation.

Figure 7 depicts an example of a StIM RIR, compared with a measured RIR. The measured RIR, in Fig. 7(c), was measured in two identical coupled rooms, both of size $(L_x, L_y, L_z) = (2.55, 3.3, 2.5)$. Figure 7(a) details the measuring setup, where the source's room is allocated between –2.55 and 0 on the x axis, while the receiver's room is allocated between 0 and 2.55 on the x axis. The source and receiver locations are $(x_s, y_s, z_s) = (-0.4, 1.3, 0.9)$ and $(x_r, y_r, z_r) = (1.28, 1.75, 0.81)$, respectively. Both rooms are real room with various objects and furniture. Figure 7(b) is the StIM measured RIR, of the same setup. Similar to the receiver and original IM, StIM is not an accurate representation of a real environment.

We have measured the generation time of IM RIR. The average over three RIR measurements, is 0.3743 s. Measuring the StIM RIR generation, the average of three measurements yielded 2.73 s for a single RIR. It is important to note that if we ignore the reflections from the sources room and simply account for the reflections from the source's room, the StIM RIR is generated in an average of 0.3726 s. This shows that the longer calculation time is only due to taking the reflections from the receiver's room into account.

V. EXPERIMENTAL FRAMEWORK AND RESULTS

A. RIR simulation

For simulation purposes, we have generated 1000 location samples. Each location is composed of two adjacent rooms of random sizes. For simplicity, all the reflection and

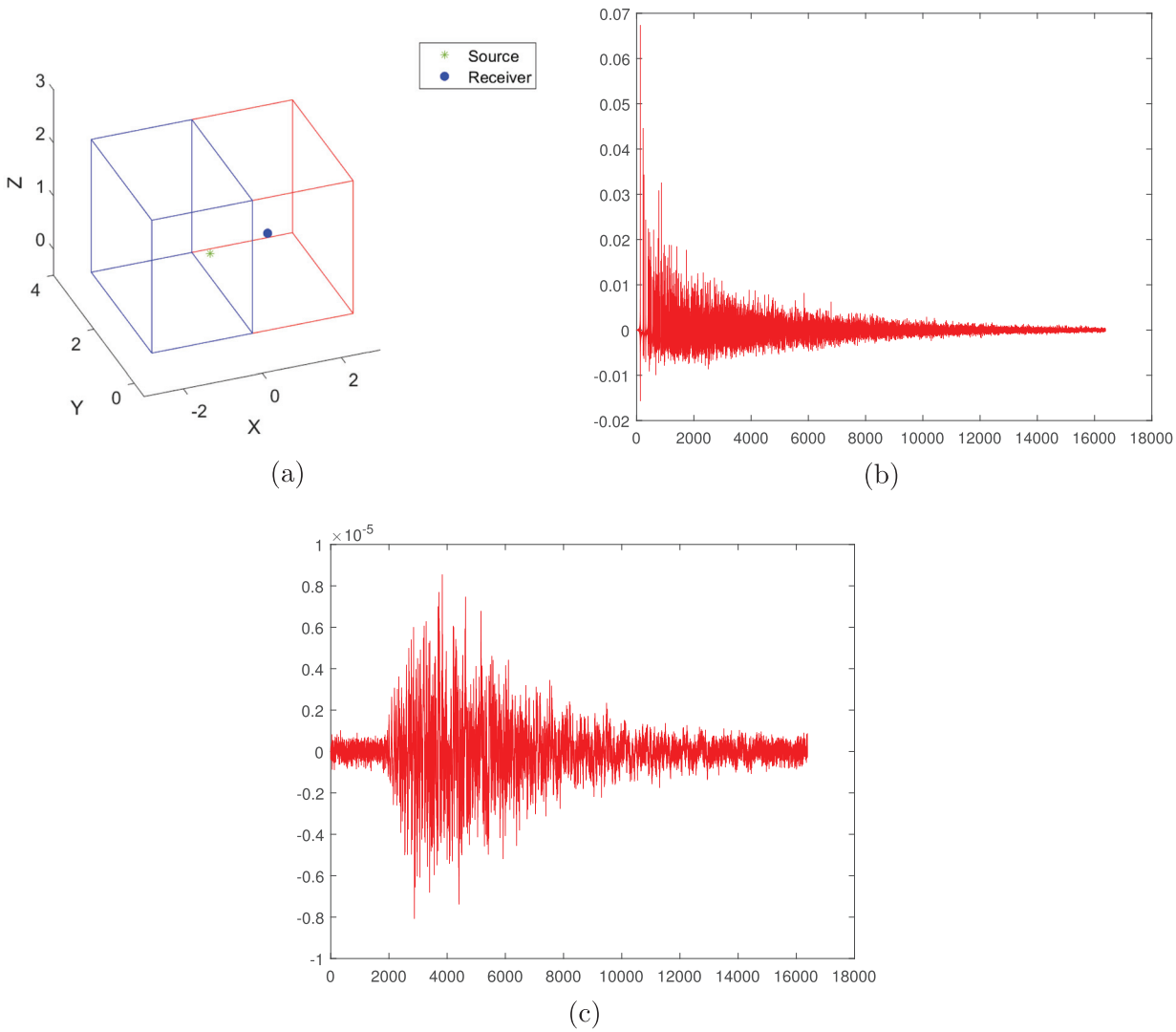

FIG. 7. (Color online) An example of StIM with a coupled room audio scene vs a measured image method. (a) Coupled rooms setup, (b) StIM, (c) measured image method.

https://doi.org/10.1121/10.0006781

transition coefficients are equal on all faces. The height, depth, and width of the rooms (H, D, W) are randomized under the constraints $2 \leq H \leq 6$, $1.5 \leq D \leq 4$, $1.5 \leq W \leq 4$. The ceiling and floor of both rooms are aligned to create the same height in both rooms, as well as an alignment with respect to a single wall, as presented in Fig. 6. The height and wall alignment assumptions are often also the case with common structures. In each location simulated, we randomize a receiver location in one of the rooms. Two sources locations are then randomized, one in each room. Thus, every location contains two RIRs, one inside the receiver's room and the second in the adjacent room. The first RIR represents the existing IM and the second RIR is calculated using StIM. All RIRs are of length 4096, with 44.1 KHz sample rate.

B. Dataset, features, and pre-processing

For audio classification, we used the ESC-50 dataset.[40] We focused on three classes associated with indoors, namely, "crying_baby," "coughing," and "toilet_flush," yielding a dictionary of size $|\mathcal{C}| = 3$. This dataset contains very clean samples, which were recorded in an interference free, quiet environment. Each sound is a 5-s segment and was sampled with 44.1 KHz sample rate. In the pre-processing phase, we used three different processes to represent three different scenarios, using the RIRs produced in Sec. IV A. The first scenario is the plain sound, as it is (clean). The second scenario includes convolving samples with IM RIRs from Sec. IV A that represent the source and receiver being inside the same room (inside_room). The third scenario simulates the sound arriving to the receiver from an adjacent room (outside_room) by convolving with the StIM RIRs from Sec. IV A. After simulating the scenario by convolving with the respective RIR, each sample is padded with zeros up to 6 s. Mel-frequency cepstral coefficient (MFCC) features are extracted for each sample using 40 coefficients calculated over mel-spectrogram, with 2048 FFT bins, a 2048 Hanning window, and 512 samples hop length. We have tested STFT and mel-spectogram features, as well. MFCC features gave the best representation out of the three, however, this may not be the best feature for other SEC tasks, such as polyphonic audio or a time-varying label. Finally, a random WGN $w(t)$, with signal to noise ratio of 30 dB is added, as in Eq. (1), and the sample is normalized with respect to root mean square (RMS) of 1. Ko *et al.*,[48] who studied the effects of IM RIRs simulation on DNN models training for a speech recognition task, showed that the transition to real environment when training with IM RIRs can be problematic. In order to mitigate this problem, they have developed a full algorithm, which uses randomized WGN point sources and ambient noise. Adding such randomness to the data deals with the fact that StIM and IM are not accurate representations of a real environment with objects, as shown in Figs. 4 and 7. Though we do not follow their full algorithm, we show that simply adding a WGN $w(t)$ to each sample is enough, in this case, to make a smooth transition. Adding randomized WGN point sources helps to better represent a real environment, as reflections from stationary objects such as a furniture and moving objects such as people are not well represented in the IM. From a deep-learning point of view, the addition of WGN can be considered as a form of vicinal risk minimization. We train on similar but different examples by adding the random vector, which is also known as data augmentation.[49,50] Following this pre-processing methodology, we augment the dataset with K RIR instances for each audio example in the cases of inside_room and outside_room. The value of K was empirically tested with the values $\{1, 5, 10, 20, 50, 100, 200\}$.

C. Deep neural network model and training

For a classification model, we used three different models. All models are CNN, as the label is constant throughout each sample, and we treat the input as a single image. One could use an RNN\CRNN model and treat the input MFCC as a time series. We leave this for future study. We chose a simple, generic, CNN as a baseline. We compare the results against an Alexnet classifier[51] and a VGG16 classifier.[52] Both Alexnet and VGG16 are common classifiers and we show that our data-augmentation method improves the results for all three methods. All models are implemented using KERAS.

1. Baseline

The input to the baseline model is a tensor of size $1 \times 40 \times 517$, where 1 is the number of channels (a gray-scale image), 40 is the number of MFCC coefficients, and 517 is the resulting time bins from Sec. IV B. The baseline method is composed of 4-layered CNN blocks followed by a dense soft-max layer. Each CNN block has a first CNN layer with a fixed kernel size of 2, relu activation functions, and $2^i \cdot 16$ filers, where $i \in 0, 1, 2, 3$, is the index of the layer. The CNN layer is followed by a max-pooling layer with a pool-size of 2. The block is concluded with a dropout layer with a drop-rate of 0.1. After all 4 blocks, we add a 2D global-average-pooling, followed by a dense, soft-max classifier of size $1 \times c$, with an l_2 regularization, where c is the number of classes to be classified. The model was trained using an Adam optimizer, with a learning rate of 0.001, and categorical cross-entropy loss.

2. Alexnet

For the Alexnet classifier, we used the original implementation with 3 alterations. We have adjusted the input to the size of $1 \times 65 \times 479$ tensor. Here, 1 is the number of channels (a gray-scale image), 65 is the number of MFCC coefficients, and 479 represents a time segment of 5.77 s. The cropping in time is done in order to fit the image to the Alexnet classifier, and is still within the zero-padding limit of Sec. IV B. The second alteration is the step size of the first layer alone. We set the step-size to (1, 9) so that the second layer of Alexnet receives the expected size. Finally, we

have altered the last soft-max layer to the required number of classes c. The model was trained using an Adam optimizer, with a learning rate of 0.001, and categorical cross-entropy loss.

3. VGG16

For the VGG16 classifier, we have also changed the size of the input size to the first layer, so that we avoid further changes to the architecture. The input is a tensor of size $1 \times 65 \times 479$. In this model we had to lower the learning rate to 0.0001. Otherwise, the model did not learn. We used an Adam optimizer, with categorical cross-entropy loss.

Prior to training, we divided the dataset into train, validation, and test sets, with equal distribution of classes in each set. For the cases of inside_room and outside_room, the 1000 RIRs produced in Sec. IV A are also divided into train, validation, and test sets. This division allows us to test the robustness of the resulting system, with respect to the location. We randomise K RIRs for each sample with respect to the division, such that RIRs in the test set were never seen in the training phase (and the same for validation). The samples are augmented by convolving with each of the K RIRs. We then proceed to extract features, according to the process described in Sec. IV B. We trained each model for 100 epoches, using early stopping with patience of 5. The number of epoches and patience were empirically chosen. We assess the results using Eq. (2).

D. Experimental results analysis

We used the scenarios described in Sec. IV B, in order to train and asses numerous models of each network. The name of each model represents the network used, the scenario which was used to train it, and the respective chosen K for inside_room and outside_room. For example, a model named "Baseline_inside_room_K_20" represents the base-model trained using IM RIRs with $K = 20$ impulse response augmentation, while "VGG_outside_room_K_100" represents the VGG16 model trained using StIM RIRs with $K = 100$ impulse response augmentation. We started by training a clean model for each network and evaluate the results. For the evaluation, we use the test set as well as real sound recorded by us in a real environment. We have recorded a total of 30 recordings, 10 for each class, using two laptops and one cellphone. The recordings were made in two of the research' residential environment due to Covid-19 isolation limitations. "Toilet_flush" was recorded in two different locations. For the first set, the source's room sizes where approximately $H = 2.5_m$, $D = 1.1_m$, $W = 0.7_m$, and for the second location $H = 2.75$, $D = 1.2_m$, $W = 1.75_m$. In both cases, the receiver's location was in a Hall, with many openings to many adjacent rooms. The source room's door was kept close and other doors from the receiver's hall were either close or open. The "coughing class" was also recorded in two sets of adjacent rooms. In the first set, the source's room is of size $H = 2.5_m$, $D = 2.5_m$, $W = 3.1_m$, and the receivers room is $H = 2.5_m$, $D = 2.5_m$, $W = 3.4_m$. In the second set, the source's room is of size $H = 2.75_m$, $D = 5.2_m$, $W = 3.3_m$, and the receivers room is $H = 2.75_m$, $D = 3.3_m$, $W = 3_m$. In both cases, the doors where kept closed. The "crying_baby" was recorded in a third, single location, where the source's room dimensions were $H = 2.1_m$, $D = 3_m$, $W = 2.75_m$ and the receiver's room is $H = 2.1_m$, $D = 3_m$, $W = 2.5_m$. These locations, present rectangular rooms (excluding halls) with various furniture, objects, and openings (doors and windows). The receiver location was altered between recordings, as was the coughing class's source location. Table III presents the training, validation, and testing results for all three networks on the clean datasets, as well as performances on the real recordings. The clean models seem to achieve very good results on the simulation phase, without the help of any augmentation. However, Table III reveals that these results can be degraded when environmental features are present.

In order to improve the environmental robustness, we trained seven additional models for each network with RIR augmentations. At the first stage, we generated 1000 such location RIRs and divided them into train, validation, and test sets. Then we chose a different number of RIRs, K, to represent the location for each of the seven models, where $K = \{1, 5, 10, 20, 50, 100, 200\}$. Out of the respective set, we have randomized K locations for each sample. This procedure represents the sound over a large variety of locations while avoiding representing each sample with all location examples, which reduces the complexity while maintaining performances. Figure 8 shows the accuracy results for the testing set as a function of K for all three classifiers for both inside_room and outside_room scenarios. The results show that using a higher order of augmentation, K, can mitigate over-fitting towards the training set's RIRs. This generally makes sense, as a higher number of K better represents the distribution of source and receiver allocations and structure of the room. The results justify the requirement for a large number of RIRs. StIM is essential in order to produce such a large number of simulated RIRs quickly. It can produce both structure RIRs, while maintaining the ability to generate an IM RIR using the receiver imaging.

In order to study the performances of each classifier when presented with a different scenario than the one presented in the training process, we proceed to evaluate the models using specific pre-processing scenarios. Table IV is divided into two parts, the first one shows the results of classifiers when presented with data from the "inside_room

TABLE III. Evaluation of architectures' performances, trained without any augmentation. The training, validation, and testing results are presented for each architecture. The furthest right column is the performance on real sound recorded by us in a real environment.

Network	Training	validation	Test	Real data
Baseline	100%	93.33%	100%	56.66%
Alexnet	100%	93.33%	96.66%	40%
VGG	100%	96.66%	100%	53.33%

https://doi.org/10.1121/10.0006781

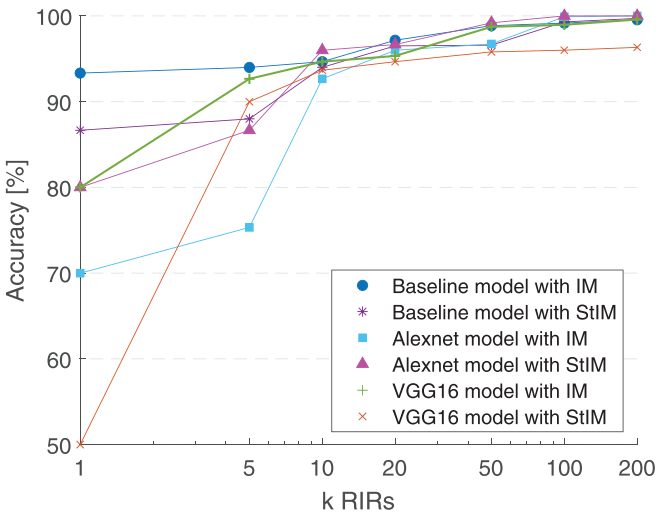

FIG. 8. (Color online) Accuracy of all three classifiers with respect to K randomly chosen RIRs.

training scenario," and the second part show the results of classifiers when presented with data from the "outside_room training scenario." We have evaluated only the $K = 200$ models, as these are the best performing models for each training scenario.

At first glance, the results in Table IV can be deceiving. The "clean" models seem to achieve similar performances when tested against scenarios they did not train with. However, as was shown in Table III, these results are only true for clean samples, recorded in a quiet and clean-of-interference environment. When we introduce the effect of reverberations to these models, the results are not as promising. Table IV shows that models trained in different incompatible scenarios can cope with other scenarios to some extent. On the first part of Table IV, the models trained under the outside_room scenario achieves below 90% for all networks. On the second part, the models trained under the inside_room scenario, also achieves below 90% for all networks. Not surprisingly, the models trained to fit the specific scenario achieve the best results.

These results show the benefits of training with generators which are capable of simulating many structural examples to augment the data, such as adjacent rooms. Since both IM and StIM are not an accurate representations of a real room, the transition from simulation into a real environment is not trivial. In order to study this transition, we proceeded to evaluate the three different models for each network, with our real recorded samples. Table V details the performances of all nine models on the real-recorded examples. The middle column in Table V shows the results for models that were trained without adding $w(t)$ in the training process. Such models achieved similar performances to the ones where $w(t)$ was added on the simulation phase. However, when transitioning to real examples, recorded in a real structure, there is an obvious degradation in the performances of all models trained without $w(t)$ (middle column of Table V).

TABLE V. Evaluation of architectures' performances, trained without augmentation (clean), with IM augmentation (inside_room), or with StIM augmentation (outside_room). The performance are with respect to a test set recorded in a real environment of coupled rooms. The middle column represents the results of models trained without the addition of a WGN $w(t)$, while the left column, represents the results of models trained with the addition of a WGN $w(t)$.

Model	Accuracy without $w(t)$	Accuracy with $w(t)$
Baseline_Clean	66.66%	56.66%
Baseline_inside_room_K_200	63.33%	83.33%
Baseline_outside_room_K_200	60%	93.33%
AlexNet_Clean	53.33%	40%
AlexNet_inside_room_K_200	63.33%	69.99%
AlexNet_outside_room_K_200	76.66%	83.33%
VGG16_Clean	50%	53.33%
VGG16_inside_room_K_200	63.33%	76.66%
VGG16_outside_room_K_200	76.66%	90%

TABLE IV. Evaluation of architectures' performances, trained without augmentation (clean), with IM augmentation (inside_room), or with StIM augmentation (outside_room). The performance in the higher table, "inside room" are with respect to a test set, pre-processed using IM. The performance in the lower table, "outside room" are with respect to a test set, pre-processed using StIM.

Inside room					
Baseline		Alexnet		VGG	
Training procedure	Accuracy	Training procedure	Accuracy	Training procedure	Accuracy
Inside_room_K_200	99.71%	Inside_room_K_200	99.94%	Inside_room_K_200	99.58%
Outside_room_K_200	73.03%	Outside_room_K_200	86.88%	Outside_room_K_200	86.53%
Clean	94.53%	Clean	84.78%	Clean	98.81%
Outside room					
Baseline		Alexnet		VGG	
Training procedure	Accuracy	Training procedure	Accuracy	Training procedure	Accuracy
Inside_room_K_200	75.43%	Inside_room_K_200	78.01%	Inside_room_K_200	89.51%
Outside_room_K_200	98.8%	Outside_room_K_200	99.81%	Outside_room_K_200	95.85%
Clean	94.96%	Clean	89.3%	Clean	96.68%

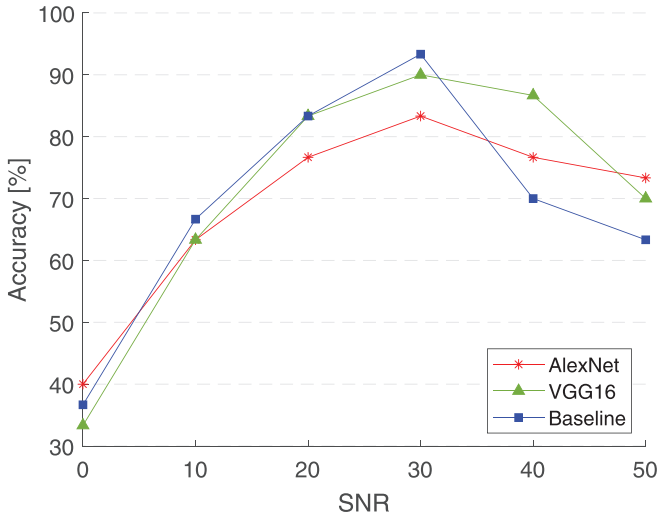

FIG. 9. (Color online) Accuracy of all three classifiers, trained using StIM, with $K = 200$, with respect to different SNR levels. The performance are with respect to a test set recorded in a real environment of coupled rooms.

From the right-most column in Table V, it is clearly visible that the models trained with RIRs of any kind, when adding $w(t)$, are performing better in real life recordings. The introduction of some randomization to an RIR can improve the results. We have added $w(t)$, follow the procedure in Sec. V B. Alternatively, it is also possible to implement on StIM the full algorithm proposed by Peddinti *et al.*[48] to deal with the transition to real-environment. Table V shows that all three classifiers benefit from StIM, when presented with a recording from another room.

Adding too much noise (low SNR) can degrade the performances simply by masking the original signal with noise. Too little sound, however, may result in an inaccurate representation of a real environment. Thus, a very high SNR is also a problem. Hence, we proceed to test each of the three networks, using $K = 200$, StIM RIRs augmentation, while altering the SNR of the $w(t)$ randomization factor. Figure 9 shows the results for different SNRs. When we look at the higher end of the SNR, where the randomization is relatively lower and has a lower impact, we observe a decrease in the performances. This is to be expected, since the results converge to the results where we do not add $w(t)$ in Table V. On the lower end, we also see a drop in performances, converging into the region of 30% to 40%. These results are close to guessing when dealing with a model, classifying between three classes. On $SNR = 0$, the signal and the random noise contribute the same mean square power to the total signal. This means that the noise obscures the signal. Hence, the classifiers are converging to guessing.

The fine-tuning of the SNR level is, probably, classifier and task dependent. Figure 9 shows that for VGG, the optimal SNR is somewhere between 30 dB and 40 dB, where for the baseline, it is much closer to 30 dB. It is possible that the tuning is also dependent on the dataset and involved classes. Broad spectrum signals will probably need more caring in the additive noise level as they are more similar in nature to the noise. Colored noises can be considered in such cases to differentiate the noise's spectral characteristics from the target class's spectral characteristics. We leave this subject for future study.

VI. CONCLUSIONS

The ability to generate simulations of sound travelling between rooms opens the possibility to describe diverse environments, and therefore improves the results for SEC, specifically, and other audio processing tasks in general. Other extensions of the IM that take frequency and incident-angle dependent factors into account should also be considered when implementing this method to achieve the best performances. The line of thought which leads to this method can be generalized to a higher number of adjacent rooms to represent even more complex structures. This method can be further expanded to describe openings between rooms, such as doors, by changing the transfer coefficient on a section of the wall to 1 and adjusting the usage of the algorithms, respectively. We showed that when utilizing a simulation to augment many environments into the dataset, it is important to have a large number of environmental examples, as well as to incorporate these examples correctly into the dataset. In addition, this work proposed a framework for a training process, using simulated RIR. Our experiments show promising results on real-world recorded data with accessible devices, such as laptops and cellphones. Products intended to utilise an SEC system will probably have similar components with such accessible devices, rendering their recordings a good representation for real-life data.

ACKNOWLEDGMENTS

The authors thank the associate editor and anonymous reviewers for their constructive comments and useful suggestions. This research was supported by the Pazy Research Foundation and the ISF-NSFC joint research program (Grant No. 2514/17).

[1]M. J. Bianco, P. Gerstoft, J. Traer, E. Ozanich, M. A. Roch, S. Gannot, and C.-A. Deledalle, "Machine learning in acoustics: Theory and applications," J. Acoust. Soc. Am. **146**(5), 3590–3628 (2019).
[2]Y. Alsouda, S. Pllana, and A. Kurti, "A machine learning driven iot solution for noise classification in smart cities," arXiv:1809.00238 (2018).
[3]G. Ciaburro and G. Iannace, "Improving smart cities safety using sound events detection based on deep neural network algorithms," Informatics **7**(3), 1–6 (2020).
[4]S. Krstulović, "Audio event recognition in the smart home," in *Computational Analysis of Sound Scenes Events* (Springer Verlag, Cham, 2018), pp. 335–371.
[5]L. Yang, H. Cheng, J. Hao, Y. Ji, and Y. Kuang, "A survey on media interaction in social robotics," in *Proceedings of the Pacific Rim Conference on Multimedia* (Springer, Berlin, 2015), pp. 181–190.
[6]C. Clavel, T. Ehrette, and G. Richard, "Events detection for an audio-based surveillance system," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, IEEE (2005), pp. 1306–1309.
[7]J. T. Geiger and K. Helwani, "Improving event detection for audio surveillance using Gabor filterbank features," in *Proceedings of the 23rd European Signal Processing Conference (EUSIPCO)*, IEEE (2015), pp. 714–718.
[8]Y. Arslan and H. Canbolat, "Performance of deep neural networks in audio surveillance," in *Proceedings of the 6th International Conference*

on Control Engineering & Information Technology (CEIT), IEEE (2018), pp. 1–5.
[9]B. U. Töreyin, Y. Dedeoğlu, and A. E. Çetin, "HMM based falling person detection using both audio and video," in *Proceedings of the International Workshop on Human-Computer Interaction* (Springer, Berlin, 2005), pp. 211–220.
[10]A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: Outcome of the DCASE 2016 challenge," IEEE/ACM Trans. Audio Speech Lang. Process. **26**(2), 379–393 (2018).
[11]Ç. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulović, "A framework for the robust evaluation of sound event detection," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE (2020), pp. 61–65.
[12]E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, IEEE (2015). pp. 1–7.
[13]I.-Y. Jeong, S. Lee, Y. Han, and K. Lee, "Audio event detection using multiple-input convolutional neural network," in *Detection and Classification of Acoustic Scenes and Events (DCASE)* (2017).
[14]S. Adavanne, G. Parascandolo, P. Pertilä, T. Heittola, and T. Virtanen, "Sound event detection in multichannel audio using spatial and harmonic features," arXiv:1706.02293 (2017).
[15]S. Adavanne, A. Politis, and T. Virtanen, "A multi-room reverberant dataset for sound event localization and detection," arXiv:1905.08546 (2019).
[16]S. Adavanne, A. Politis, and T. Virtanen, "A multi-room reverberant dataset for sound event localization and detection," in *Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)* (2019).
[17]M. R. Bai, S.-S. Lan, J.-Y. Huang, Y.-C. Hsu, and H.-C. So, "Audio enhancement and intelligent classification of household sound events using a sparsely deployed array," J. Acoust. Soc. Am. **147**(1), 11–24 (2020).
[18]J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE (2017), pp. 776–780.
[19]P. Smith, Jr., "Response and radiation of structural modes excited by sound," J. Acoust. Soc. Am. **34**(5), 640–647 (1962).
[20]R. H. Lyon and G. Maidanik, "Power flow between linearly coupled oscillators," J. Acoust. Soc. Am. **34**(5), 623–639 (1962).
[21]A. Craggs, "The use of simple three-dimensional acoustic finite elements for determining the natural modes and frequencies of complex shaped enclosures," J. Sound Vib. **23**(3), 331–339 (1972).
[22]G. Gladwell, "A variational formulation of damped acousto structural vibration problems," J. Sound Vib. **4**(2), 172–186 (1966).
[23]A. Burton and G. Miller, "The application of integral equation methods to the numerical solution of some exterior boundary-value problems," R. Soc. London, Ser. A **323**(1553), 201–210 (1971).
[24]D. Colton and R. Kress, *Integral Equation Methods in Scattering Theory* (SIAM, Philadelphia, 2013).
[25]T. Walsh, L. Demkowicz, and R. Charles, "Boundary element modeling of the external human auditory system," J. Acoust. Soc. Am. **115**(3), 1033–1043 (2004).
[26]L. Savioja, J. Backman, A. Järvinen, and T. Takala, "Waveguide mesh method for low-frequency simulation of room acoustics," in *Proceedings of the 15th International Conference on Acoustics (ICA-95)*, Trondheim, Norway (1995). pp. 637–640.
[27]A. Krokstad, S. Strom, and S. Sørsdal, "Calculating the acoustical room response by the use of a ray tracing technique," J. Sound Vib. **8**(1), 118–125 (1968).
[28]S. Siltanen, T. Lokki, and L. Savioja, "Rays or waves? understanding the strengths and weaknesses of computational room acoustics modeling techniques," in *Proceedings of the International Symposium on Room Acoustics* (2010).
[29]J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," J. Acoust. Soc. Am. **65**(4), 943–950 (1979).
[30]E. A. Habets, "Room impulse response generator," Techn. Univ. Eindhoven Tech. Rep **2**(2.4), 1–21 (2006), see https://www.researchgate.net/profile/Emanuel-Habets/publication/259991276_Room_Impulse_Response_Generator/links/5800ea5808ae1d2d72eae2a0/Room-Impulse-Response-Generator.pdf.
[31]J. Borish, "Extension of the image model to arbitrary polyhedra," J. Acoust. Soc. Am. **75**(6), 1827–1836 (1984).
[32]M. Vorländer, "Simulation of the transient and steady-state sound propagation in rooms using a new combined ray-tracing/image-source algorithm," J. Acoust. Soc. Am. **86**(1), 172–178 (1989).
[33]J. H. Rindel, "Modelling the angle-dependent pressure reflection factor," Appl. Acoust. **38**(2-4), 223–234 (1993).
[34]Y. W. Lam, "Issues for computer modelling of room acoustics in non-concert hall settings," Acoust. Sci. Technol. **26**(2), 145–155 (2005).
[35]H. Sinha, V. Awasthi, and P. K. Ajmera, "Audio classification using braided convolutional neural networks," IET Sign. Process. **14**(7), 448–454 (2020).
[36]I. Rodomagoulakis, N. Kardaris, V. Pitsikalis, E. Mavroudi, A. Katsamanis, A. Tsiami, and P. Maragos, "Multimodal human action recognition in assistive human-robot interaction," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE (2016), pp. 2702–2706.
[37]X. Wu, H. Gong, P. Chen, Z. Zhong, and Y. Xu, "Surveillance robot utilizing video and audio information," J. Intell. Robotic Syst. **55**(4), 403–421 (2009).
[38]A. Mesaros, A. Diment, B. Elizalde, T. Heittola, E. Vincent, B. Raj, and T. Virtanen, "Sound event detection in the DCASE 2017 challenge," IEEE/ACM Trans. Audio Speech Lang. Process. **27**(6), 992–1006 (2019).
[39]Q. Kong, I. Sobieraj, W. Wang, and M. Plumbley, "Deep neural network baseline for DCASE challenge 2016," in *Proceedings of DCASE 2016* (2016).
[40]K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*, ACM (2015), pp. 1015–1018.
[41]A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," in *Proceedings of the 18th European Signal Processing Conference*, IEEE (2010), pp. 1267–1271.
[42]X. Zhuang, X. Zhou, M. A. Hasegawa-Johnson, and T. S. Huang, "Real-world acoustic event detection," Pattern Recogn. Lett. **31**(12), 1543–1551 (2010).
[43]J. F. Gemmeke, L. Vuegen, P. Karsmakers, B. Vanrumste *et al.*, "An exemplar-based NMF approach to audio event detection," in *Proceedings of the IEEE Workshop Applications Signal Processing to Audio Acoustics*, IEEE (2013), pp. 1–4.
[44]A. Temko, R. Malkin, C. Zieger, D. Macho, C. Nadeu, and M. Omologo, "Clear evaluation of acoustic event detection and classification systems," in *Proceedings of the International Evaluation Workshop on Classification of Events, Activities and Relationships*, Springer (2006), pp. 311–322.
[45]H. Kuttruff, *Room Acoustics* (CRC Press, Boca Raton, FL, 2016).
[46]L. E. Kinsler, A. R. Frey, A. B. Coppens, and J. V. Sanders, *Fundamentals of Acoustics* (Wiley, New York, 1999).
[47]A. Farina, "Advancements in impulse response measurements by sine sweeps," in *Audio Engineering Society Convention 122*, Audio Engineering Society (2007).
[48]T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE (2017), pp. 5220–5224.
[49]O. Chapelle, J. Weston, L. Bottou, and V. Vapnik, "Vicinal risk minimization," in *Advances in Neural Information Processing Systems 13 (NIPS 2000)* (2001), pp. 416–422.
[50]H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," arXiv:1710.09412 (2017).
[51]A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Adv. Neural Inf. Process. Syst. **25**, 1097–1105 (2012).
[52]K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv:1409.1556 (2014).