EURASIP Journal on Audio, Speech, and Music Processing

EMPIRICAL RESEARCH

Open Access

Points2Sound: from mono to binaural audio using 3D point cloud scenes

Francesc Lluís*, Vasileios Chatziioannou and Alex Hofmann

Abstract

For immersive applications, the generation of binaural sound that matches its visual counterpart is crucial to bring meaningful experiences to people in a virtual environment. Recent studies have shown the possibility of using neural networks for synthesizing binaural audio from mono audio by using 2D visual information as guidance. Extending this approach by guiding the audio with 3D visual information and operating in the waveform domain may allow for a more accurate auralization of a virtual audio scene. We propose Points2Sound, a multi-modal deep learning model which generates a binaural version from mono audio using 3D point cloud scenes. Specifically, Points2Sound consists of a vision network and an audio network. The vision network uses 3D sparse convolutions to extract a visual feature from the point cloud scene. Then, the visual feature conditions the audio network, which operates in the waveform domain, to synthesize the binaural version. Results show that 3D visual information can successfully guide multi-modal deep learning models for the task of binaural synthesis. We also investigate how 3D point cloud attributes, learning objectives, different reverberant conditions, and several types of mono mixture signals affect the binaural audio synthesis performance of Points2Sound for the different numbers of sound sources present in the scene.

Keywords: Binaural synthesis, Neural network, Audio-visual learning, 3D point clouds

1 Introduction

People perceive the world through multiple senses that jointly collaborate to understand the environment. While visual stimuli are important for spatial cognition, auditory stimuli are particularly critical. For example, being capable of hearing instantly from all angles helps people orient themselves in space and influences their visual attention [1, 2]. As auditory stimuli are received by both ears, our brain locates sound sources in space by comparing the sound that our ears receive. This process, known as binaural hearing, relies mainly on two acoustic cues: interaural time difference (ITD) and interaural level difference (ILD). ITD is the difference in the arrival time of a sound between the ears and ILD is the difference in sound intensity. In the median plane, i.e., the vertical plane between the ears, ITD and ILD are both small, and we rely on spectral cues to locate sources [3]. All such acoustic cues can be described by the head-related transfer function (HRTF), which encodes the sound distortion caused by geometries of the head and the torso [4].

In immersive applications, the generation of accurate binaural acoustic cues that match the visual counterpart is key to providing people with meaningful experiences in the virtual environment. These acoustic cues, ITD and ILD, strongly rely on the 3D position between the receiver and the sound sources. Recently, several methods using neural networks have been proposed for generating binaural audio from mono audio, using 2D visual information as guidance [5–8]. However, using 2D visual information inherently restricts the neural network's ability to extract information about the 3D positions between the receiver and the sound sources present in a scene. Not having access to this potentially useful information entails the risk of finding a sub-optimal solution for the task of binaural generation. In addition, these recent methods extract visual information by applying 2D dense

*Correspondence: lluis-salvado@mdw.ac.at

Department of Music Acoustics, University of Music and Performing Arts Vienna, Vienna, Austria

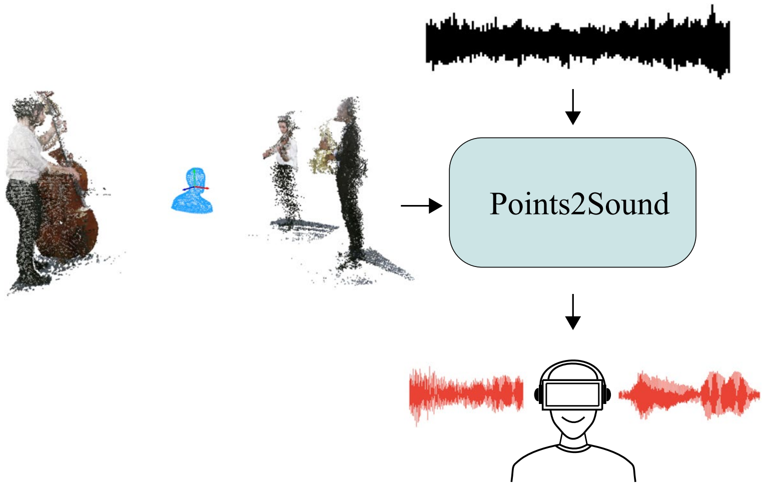

Fig. 1 Points2Sound is a deep learning model capable of generating a binaural version from mono audio that matches a 3D point cloud scene

convolutions to planar projections of the scene. This process forces the 2D convolutional filters to attend to local planar-projection regions with no relationship to physical space—possibly hindering the audio-visual learning necessary for the binaural synthesis task.

In this paper, we introduce Points2Sound, a multi-modal neural network that synthesizes binaural audio from mono audio using 3D visual information as guidance (see Fig. 1). For the visual learning, we propose the use of 3D point clouds as visual information as well as 3D sparse convolutions for extracting information from the 3D point clouds. This approach enables the model to extract information about the 3D position of the sound sources while convolutional filters attend to 3D data structures in local regions of the 3D space. For the audio learning, Points2Sound uses advancements in neural audio modeling in the waveform domain. We extend the Demucs architecture [9] and show that this model can be effectively conditioned by using visual information. Although Demucs was originally designed for source separation, we find it appropriate for binaural synthesis given that the model needs to intrinsically separate the sound of the sources in the mono mixture for further binaural rendering. This study thus analyzes the performance of Points2Sound for different 3D point cloud attributes, learning objectives, reverberant conditions, and types of mono mixtures.

The main contributions of this work are the following:

- We introduce the use of 3D point clouds to condition audio signals. By using 3D visual information and 3D sparse convolutions, the neural network can learn the correspondence between audio characteristics (e.g., spatial cues or timbre) and 3D structures found in local regions of the 3D space—a correspondence relevant for binaural audio synthesis.
- We tackle visually informed binaural audio generation directly in the waveform domain, thereby optimizing our model in an end-to-end fashion and allowing it to learn audio features that are not limited by a fixed resolution of a spectrogram representation.
- We evaluate how 3D point cloud attributes, i.e., depth or rgb-depth, several types of mono mixture signals, the effect of the room, and different learning objectives affect binaural audio synthesis performance for different numbers of sound sources in the scene.
- We provide a dataset of the captured 3D-video point clouds, videos with listening examples, and the source code for reproducibility purposes[1].

This paper is organized as follows: Section 2 provides a brief overview of the related work. Section 3 details the neural network architecture, the training procedure, and the data used. Section 4 presents the evaluation metrics and the obtained results. Section 5 discusses the results, and Section 6 concludes.

2 Related work

We provide a brief overview of related works in the field of audio-visual source separation and audio-visual spatial audio generation.

[1] https://github.com/francesclluis/points2sound

2.1 Audio-visual source separation

Source separation has been traditionally approached using only audio signals [10, 11] with methods such as independent component analysis [12], sparse coding [13], or non-negative matrix factorization [14]. Recently, audio source separation has experienced significant progress due to the application of deep learning methods [15–18]. Current trends include performing source separation in the waveform domain [9, 19, 20] or preserving binaural cues during the separation process [21, 22]. In addition, deep learning methods have facilitated the inclusion of visual information to guide the audio separation [23, 24]. In the case of music source separation using visual information, learning methods mainly use appearance cues from the 2D visual representations [25, 26], but have been enhanced also with motion information [27, 28]. Interestingly, audio-visual source separation models intrinsically learn to map audio to their corresponding position in the visual representation. This has encouraged the use of visual information for spatial audio generation [5].

2.2 Audio-visual spatial audio generation

With the recent advances on audio modeling using neural networks, end-to-end deep learning approaches using explicit information about the position and orientation of the sources in the 3D space have been proposed for binaural audio synthesis [29, 30]. These approaches require head tracking equipment to know the pose of the receiver and the sound source in the environment. Concurrently, audio-visual learning for spatial audio generation has gained interest. Several methods have been proposed to infer spatial acoustic cues to mono audio from leveraging visual information. Morgado et al. [31] proposed a learning method to generate the spatial version of mono audio guided by 360° videos. Their approach is to predict the spatial audio in ambisonics format which can be later decoded as binaural audio for reproduction through headphones. Gao et al. [7] show that directly predicting the binaural audio creates better 3D sound sensations. They propose a U-Net-like framework for mono-to-binaural conversion using normal field-of-view videos. Since then, binauralization models using 2D visual information have been enhanced using different approaches such as using an auxiliary classifier [8] or integrating the source separation task in the overall binaural generation framework [5]. In addition, it has been shown that features from pretrained models on audio-visual spatial alignment tasks are beneficial for audio binauralization [6]. Note that many of these approaches use audio spectrogram representations while considering mono mixture signals represented as the sum of the two spatial channels in order to train their networks [5–8]. It remains unclear how operating in the waveform domain and using other mono representations may affect the binaural synthesis performance.

3 Approach

We propose Points2Sound, a deep learning algorithm capable of generating a binaural version from a mono audio using the 3D point cloud scene of the sound sources in the environment.

3.1 Problem formulation

Consider an audio mono signal $s_m \in \mathbb{R}^{1\times T}$ of length T samples generated by N sources s_i, with

$$s_m(t) = \sum_{i=1}^{N} s_i(t) \tag{1}$$

along with the 3D scene of the sound sources in the 3D space represented by a set of I points $\mathcal{P} = \{\mathcal{P}_i\}_{i=1}^{I}$, and the corresponding binaural signal $s_b \in \mathbb{R}^{2\times T}$. We aim at finding a model f with the structure of a neural network such that $s_b(t) = f(s_m(t), \mathcal{P})$. The binaural version $s_b(t)$ generated by N sources s_i is defined as:

$$s_b^{L,R}(t) = \sum_{i=1}^{N} s_i(t) \circledast \mathrm{HRTF}(\varphi_i, \theta_i, \mathrm{d}_i)|_{\mathrm{left,right}} \tag{2}$$

where $\mathrm{HRTF}(\varphi_i, \theta_i, \mathrm{d}_i)|_{\mathrm{left,right}}$ is the head-related transfer function of the sound incidence at the specified i-source orientation (φ_i, θ_i) and distance d_i for both left (L) and right (R) ears. Throughout this work, orientation and distance are defined based on a head-related coordinate system, i.e., the center of coordinates is considered the head of the listener. During the training of the model, we consider generic HRTFs measured in an anechoic environment. However, during testing, we also evaluate the model performance under reverberant room conditions by using two binaural room impulse responses. Also, note that in this work only the sound source's contribution through the 3D point cloud is explicitly considered for binauralization. The contributions of the listener and the environment are implicitly considered via the choice of HRTF.

3.2 Points2Sound model architecture

We propose a multi-modal neural network architecture capable of synthesizing binaural audio in an end-to-end fashion: the architecture takes as inputs the 3D point cloud scene along with the raw mono waveform and outputs the corresponding binaural waveform. The architecture comprises a vision network and an audio network. Broadly, the vision network extracts a visual feature from the 3D point cloud scene that serves to condition the

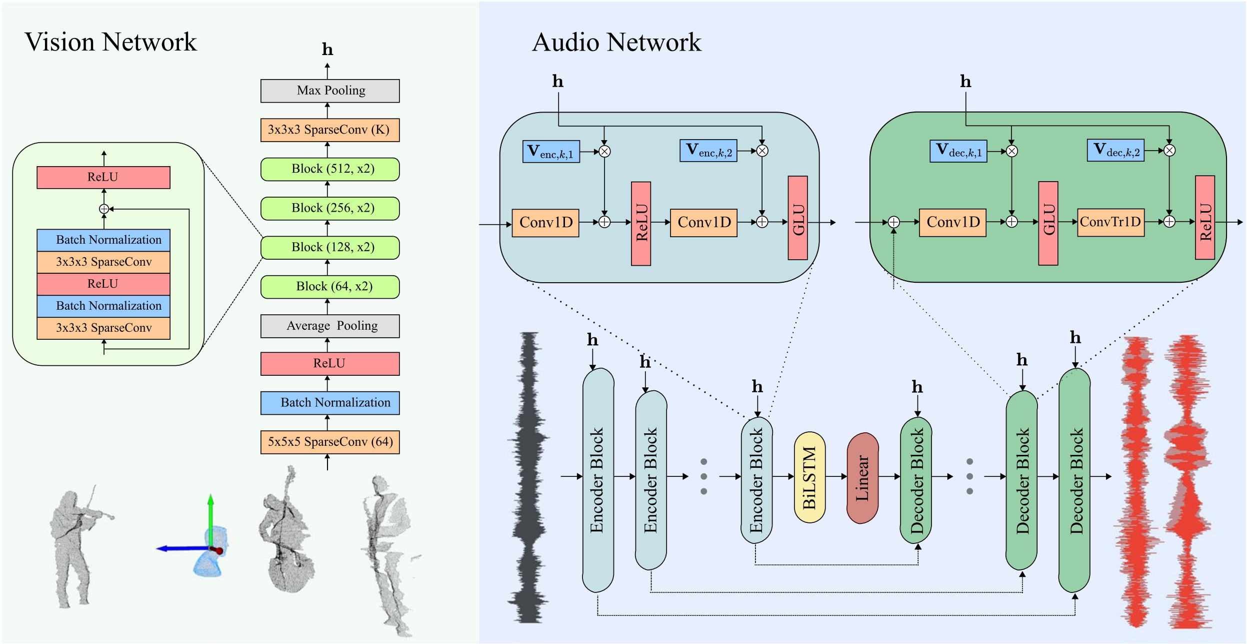

Fig. 2 Overview diagram of Points2Sound. It consists of a sparse Resnet18 network for visual analysis and a Demucs network for binaural audio synthesis. The vision network extracts a visual feature **h** from the 3D point cloud. Then, this visual feature serves to condition the audio network to generate a binaural version from the mono audio that matches the visual counterpart. Both networks are jointly optimized during the training of the model

audio network for binaural synthesis. Figure 2 shows a schematic diagram of the proposed model.

3.2.1 *Sparse tensor for 3D point cloud representation*

3D scenes captured by rgb-depth cameras or LiDAR scanners can be represented by a 3D point cloud. A 3D point cloud is a low-level representation of a 3D scene that consists of a collection of I points $\{\mathcal{P}_i\}_{i=1}^{I}$. The essential information associated to each point $\mathcal{P}_i$ is its location in space. For example, in a Cartesian coordinate system, each point $\mathcal{P}_i$ is associated with a triple of coordinates $\mathbf{c}_i = (x_i, y_i, z_i) \in \mathbb{R}^3$ in the x, y, and z-axes. In addition, each point can have associated features $\mathbf{f}_i \in \mathbb{R}^n$ like its color.

Extracting information from a 3D point cloud using neural networks requires non-standard operations that can handle the 3D data sparsity. It is common to represent the 3D point cloud information using a sparse tensor [32–34] and define operations on that sparse tensor, such as the 3D sparse convolution operation. Note that sparse tensors require a discretization step that enables point cloud coordinates to be defined in the integer grid of the sparse tensor. In this work, the 3D point cloud is represented with a third order tensor by first discretizing its coordinates using a voxel size v_s. The voxel size v_s denotes the discretization step size and allows to define point cloud coordinates in the integer grid of the tensor. The discretized coordinates of each point are given by $\mathbf{c}'_i = \lfloor \frac{\mathbf{c}_i}{v_s} \rfloor = (\lfloor \frac{x_i}{v_s} \rfloor, \lfloor \frac{y_i}{v_s} \rfloor, \lfloor \frac{z_i}{v_s} \rfloor)$. Then, the resultant tensor representing the point cloud is given by

$$\mathbf{T}[\mathbf{c}'_i] = \begin{cases} \mathbf{f}_i & \text{if } \mathbf{c}'_i \in C' \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where C' is the set of discretized coordinates of the point cloud and $\mathbf{f}_i$ is the feature associated to the point $\mathcal{P}_i$. Note that in the following we will evaluate how 3D point cloud attributes affect the binaural audio synthesis. Accordingly, we will consider two types of 3D point cloud scenes: when 3D point cloud scenes consist of depth-only data and when 3D point cloud scenes consist of rgb-depth data. In the cases where depth-only information is available, we use the non-discretized coordinates as the feature vectors associated to each point, i.e., $\mathbf{f}_i = \mathbf{c}_i$. When rgb-depth information is available, we use the rgb values as the feature vectors associated to each point, i.e., $\mathbf{f}_i = (\mathrm{r}_i, \mathrm{g}_i, \mathrm{b}_i)$.

3.2.2 *3D sparse convolution on a 3D sparse tensor*

The 3D sparse convolution is a generalized version of the conventional dense convolution designed to operate on a 3D sparse tensor [35].

The 3D sparse convolution on a 3D sparse tensor is defined as follows:

$$\mathbf{T}^{\text{out}}[x,y,z] = \sum_{p,j,k\in\mathcal{N}(x,y,z)} \mathbf{W}[p,j,k]\mathbf{T}^{\text{in}}[x+p,y+j,z+k] \tag{4}$$

for $(x,y,z) \in C'_{\text{out}}$. Where $\mathcal{N}(x,y,z) = \{(p,j,k) \mid |p| \leq \Gamma, |j| \leq \Gamma, |k| \leq \Gamma, (p+x, j+y, k+z) \in C'_{\text{in}}\}$. $\mathbf{W}$ are the weights of the 3D convolutional kernel and $2\Gamma + 1$ is the convolution kernel size. C'_{in} and C'_{out} are predefined input and output discretized coordinates of sparse tensors [35].

3.2.3 Vision network

The vision network consists of a Resnet18 [36] architecture with 3D sparse convolutions [35] that extracts a visual feature from the 3D point cloud scene. Resnet18 with 3D sparse convolutions has been successfully used in several tasks such as 3D semantic segmentation [35] or 3D single-shot object detection [34]. Thus, we consider sparse Resnet18 suitable for our scenario, where extracting information about the position of the sources while recognizing the type of source is critical for reliable binaural synthesis. Sparse Resnet18 learns at different scales by halving the feature space after two residual blocks and doubling the receptive field by using a stride of 2. A key characteristic of residual blocks is their residual connections which allow to propagate the input data through later parts of the block by skipping some layers. Through the network, ReLU is used as an activation function and batch normalization is applied after sparse convolutions. At the top of the Resnet18 4^{th} block, we add a 3×3×3 sparse convolution with $K = 16$ output channels and apply a max-pooling operation to adequate the dimensions of the extracted visual feature $\mathbf{h}$.

3.2.4 Audio network

We adapt the Demucs architecture [9] to synthesize binaural versions $\hat{s}_b$ from mono audio signals s_m given the 3D scene of the sound sources $\mathcal{P}$. Although Demucs was originally designed for source separation, we find it appropriate for binaural synthesis because the model needs to intrinsically learn to separate the sound of the sources in the mono mixture for further rendering (see Eq. 2). Demucs works in the waveform domain and has a U-Net-like structure [37] (see Fig. 2). The encoder-decoder structure learns multi-resolution features from the raw waveform while skip connections allow low-level information to be propagated through the network. In the current case, skip connections allow later decoder blocks of the network to access information related to the phase of the input signal, which otherwise may be lost when propagated through the network. In this work, we keep the original six convolution blocks for both the encoder and decoder but extend the architecture so that the input and output channels match our mono and binaural signals.

3.2.5 Conditioning

We use a global conditioning approach on the audio network to guide the binaural synthesis according to the 3D scene. Global conditioning was introduced in Wavenet [38] and has been recently used in the Demucs architecture for separation purposes using one-hot vectors [39]. In a similar way, we use the extracted visual feature $\mathbf{h}$ from the vision network and insert it in each encoder and decoder block of the audio network. Specifically, the visual feature is inserted after being multiplied by a learnable linear projection $\mathbf{V}_{\cdot,q,\cdot}$. As in [39], Demucs encoder and decoder take the following expression:

$$\begin{aligned}\texttt{Encoder}_{q+1} = {} & \text{GLU}(\mathbf{W}_{\text{encoder},q,2} * \text{ReLU}(\mathbf{W}_{\text{encoder},q,1} * \\ & \texttt{Encoder}_{q} + \mathbf{V}_{\text{encoder},q,1}\mathbf{h}) + \mathbf{V}_{\text{encoder},q,2}\mathbf{h}),\end{aligned} \tag{5}$$

$$\begin{aligned}\texttt{Decoder}_{q-1} = {} & \text{ReLU}(\mathbf{W}_{\text{decoder},q,2} *^{\top} \text{GLU}(\mathbf{W}_{\text{decoder},q,1} * \\ & (\texttt{Encoder}_{q} + \texttt{Decoder}_{q}) + \mathbf{V}_{\text{decoder},q,1}\mathbf{h}) \\ & + \mathbf{V}_{\text{decoder},q,2}\mathbf{h}).\end{aligned} \tag{6}$$

where $\texttt{Encoder}_{q+1}$ and $\texttt{Decoder}_{q-1}$ are the outputs from the q-th level encoder and decoder blocks, respectively. $\mathbf{W}_{\cdot,q,\cdot}$ are the 1-D kernel weights at the q-th block. Rectified linear unit (ReLU) and gated linear unit [40] (GLU) are the corresponding activation functions. The operator $*$ denotes the 1-D convolution while $*^{\top}$ corresponds to a transposed convolution operation, as commonly defined in the deep learning frameworks [41].

3.2.6 Learning objective

During the training of Points2Sound, the parameters of both vision and audio networks are optimized to reduce the L1 loss function between the estimated binaural signal $\hat{s}_b^{L,R}$ and the ground truth binaural signal $s_b^{L,R}$. The L1 loss computes the absolute error between the estimated and the ground truth waveform samples. We refer to this learning objective as

$$\mathcal{L}_{\text{full}} = \| s_b^{L,R} - \hat{s_b}^{L,R} \|. \tag{7}$$

Note that in Section 4.3 we will investigate the effect of another learning objective on the performance of Points2Sound.

3.3 Data

While there are lots of audio datasets, there is data scarcity of 3D point cloud videos of performing musicians. For the purposes of this work, we capture 3D videos of the same twelve performers playing different instruments: cello, doublebass, guitar, saxophone, and violin. In addition, we separately collect audio recordings of these instruments from existing audio datasets. This data will serve later to generate 3D audio-visual scenes for supervised learning.

3.3.1 Point clouds

Recordings were conducted using an Azure Kinect DK (by Microsoft) placed 1 m above the floor and capturing a frontal view of the musician at a distance of 2 m. Azure Kinect DK comprises a depth camera and a color camera. The depth camera was capturing a 75° × 65° field of view with a 640 × 576 resolution while the color camera was capturing with a 1920 × 1080 resolution. Both cameras were recording at 15 fps, and Open3D library [42] was then used to align depth and color streams and generate a point cloud for each frame. The full 3D video recordings span 1 h of duration with an average of 12 performers for each instrument.

We increase our 3D point cloud video dataset collecting 3D videos from *small ensemble 3D-video database* [43] and *Panoptic Studio* [44]. In *small ensemble 3D-video database*, recordings are carried out using three RGB-Depth Kinect v2 sensors. LiveScan3D [45] and OpenCV libraries are then used to align and generate point clouds for each frame given each camera point of view and sensor data. The average video recording is 5 min per instrument and a single performer per instrument. In *Panoptic Studio*, recordings are carried out using ten Kinect sensors. In this case, recordings span two instrument categories: cello and guitar. The average time per video recording is 2 min per instrument for a single performer per instrument. As we gather 3D point cloud videos from different sources, we set the axes representing the point clouds to have the same meaning for all the collected 3D videos. This is as follows: the body face direction is the z-axis, the stature direction is the y-axis, and the side direction is the x-axis. Then, we split 75% of the data for training, 15% for validation, and the remaining 10% for testing. Data split is made ensuring that there is no overlap in identities between sets.

3.3.2 Audio

We collect 30 h of mono audio recordings at 44.1 kHz from *Solos* [46] and *Music* [25]. Both datasets gather music from YouTube which ensures a variety of acoustic conditions. In total, we gather 72 recordings per instrument with an average of 5 min per recording. We split 75% of the recordings for training, 15% for validation, and the remaining 10% for testing.

For further binaural auditory scene generation, we also create multiple binaural versions of each recording using the *Two!Ears* Binaural Simulator [47]. Specifically, for each audio recording, we simulate the binaural version at a discrete set of angular positions in the horizontal plane with no elevation, i.e., $(\varphi_k, \theta) := (\frac{k\pi}{4}, 0)$ for $k = 0, \dots, 7$. For binaural auditory modeling, we use the HRTFs at 1m of distance between source and receiver measured with a KEMAR manikin (type 45BA) at the anechoic chamber of the TU Berlin [48].

3.4 Audio-visual 3D scene generation

We synthetically create mono mixtures, 3D scenes, and the corresponding binaural version to train the model in a supervised fashion.

For each instance, we randomly select N sources and N angular positions with N chosen uniformly between 1 and 3. The binaural mixture, which serves as supervision, is created following Eq. 2. First, we select 3-s length binaural signals for each sound source in the mix based on its angular position, and then, we sum all selected binaural signals to create the binaural mixture.

For the 3D scene, we first select individual musician's point clouds corresponding to these sources. Then, musician point clouds are located at their corresponding angular position in a random distance ranging from 1 to 3 m from the listener's head in the 3D space. Finally, all N musician point clouds are merged to create a single 3D point cloud scene. Note that we generate binaural versions using HRTFs computed at 1-m distance from the listener's head but locate the sources in a distance ranging from 1 to 3 m. This assumption is based on the fact that distance has a smaller influence on the shape of HRTFs, for source-receiver distances greater than 1m [49].

During the training of the model, each individual musician point cloud is independently augmented in both coordinates and color. We randomly shear and translate the coordinates of each musician in the scene. Shearing is applied along all axes and the shear elements are sampled from a Normal distribution $\mathcal{N}(0, 0.1^2)$. Translation is applied along the stature direction and the translation offset is sampled from $\mathcal{N}(0, 0.2^2)$.

Regarding color, we distort the brightness and intensity of each sound source in the scene. Specifically, we apply color distortion to each point via adding Gaussian noise sampled from $\mathcal{N}(0, 0.05^2)$ on each rgb color channel. We also alter color value and color saturation with random amounts uniformly sampled ranging from −0.2 to 0.2 and −0.15 to 0.15, respectively. Figure 3 illustrates the different augmentation operations applied. After augmentation, each scene is represented as a sparse tensor by discretizing the point cloud coordinates using a voxel size of 0.02 m. We select a small voxel size as it has been shown to work better than bigger ones for several 3D visual tasks [35].

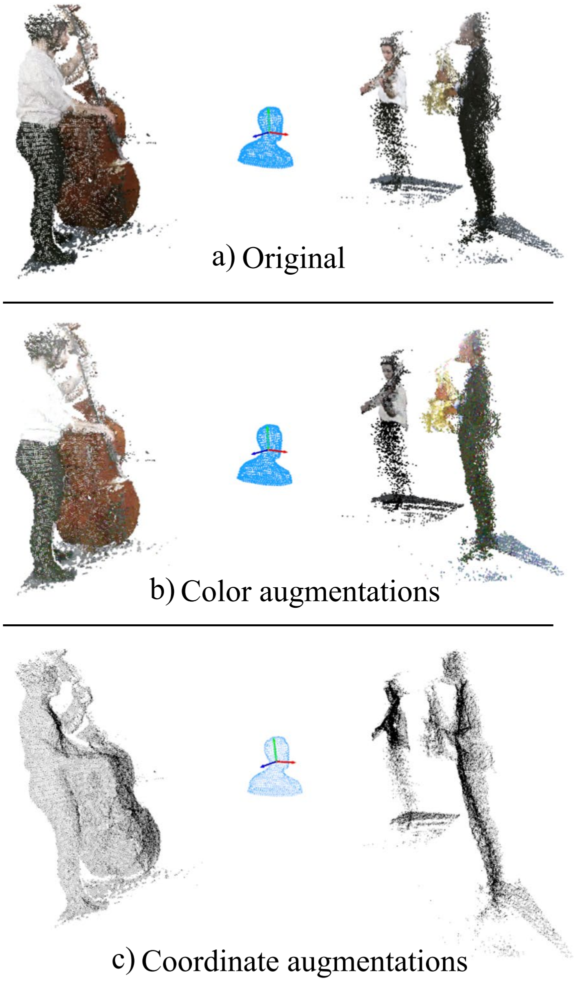

Fig. 3 Illustration of the augmentation operations applied to a 3D point cloud scene. **a** Original scene. **b** Color augmentations applied to the original scene. Color augmentations include modifying the brightness and intensity of the sound sources as well as distort the color of each point using Gaussian noise. **c** Coordinate augmentations applied to the original scene. Coordinate augmentations include random shearing along all axes and random translation along the stature direction

3.5 Implementation details

Initially, we pretrain the vision network to facilitate the future learning process. Pretraining is done on the 3D object classification task modelnet40 [50]. Since modelnet40 consists of 3D CAD models, we sample point clouds from the mesh surface of the objects shapes. For the pretraining, we also discretize the coordinates setting the voxel size to 0.02 m. Then, Points2Sound vision and audio networks are jointly trained for 120k iterations using the Adam [51] optimizer. We use a batch size of 40 samples and we set the learning rate to 1×10^{-4}. We select the weights corresponding to the lowest validation loss after the training process. Training and testing are conducted on a single Titan RTX GPU. The training stage takes about 72 h while the inference takes 0.115 s to binauralize 10 s of mono audio (value averaged from 300 samples). We use the Minkowski Engine [35] for the sparse tensor operations and PyTorch [41] for the other operations required.

4 Results

4.1 Evaluation metrics

As in previous work [31], we measure the quality of the predicted binaural audio assessing the short-time Fourier transform (STFT) Distance. Using the STFT Distance, we assess how similar the frequency components of each predicted binaural channel are to the ground truth. STFT Distance ($\mathrm{d_{STFT}}$) between a binaural signal s_b and its estimate $\hat{s_b}$ is defined as:

$$\begin{aligned}\mathrm{d_{STFT}} =&\|\mathrm{STFT}(s_b^L(t)) - \mathrm{STFT}(\hat{s}_b^L(t))\|_2\\ &+ \|\mathrm{STFT}(s_b^R(t)) - \mathrm{STFT}(\hat{s}_b^R(t))\|_2\end{aligned} \quad (8)$$

where $\|\cdot\|_2$ is the L2 norm and $\mathrm{STFT}(\cdot)$ is the short-time Fourier transform. The STFT is computed using a Hann window of 23 ms and a hop length of 10 ms.

We also assess the quality of the predicted binaural audio using the Envelope Distance. The Envelope Distance operates in the time domain and is intended to capture the perceptual similarity between two binaural signals in a better way than directly computing the loss between its waveform samples [31]. Envelope Distance ($\mathrm{d_{ENV}}$) between a binaural signal s_b and its estimate $\hat{s_b}$ is defined as:

$$\begin{aligned}\mathrm{d_{ENV}} =&\|E[s_b^L(t)] - E[\hat{s}_b^L(t)]\|_2\\ &+ \|E[s_b^R(t)] - E[\hat{s}_b^R(t)]\|_2\end{aligned} \quad (9)$$

where $E[s(t)]$ corresponds to the envelope of the signal $s(t)$. The envelope is given by the magnitude of the analytical signal computed using the Hilbert transform.

Note that we report the performance depending on the number of sources ($N = 1, 2, 3$), based on the average of the evaluation metrics. Average values for any number of sources are given by $\overline{\mathrm{d_{ENV}}}$ and $\overline{\mathrm{d_{STFT}}}$. Also, before computing the distances, the predicted and the ground truth signals are normalized according to their maximum absolute value.

4.2 Baselines

We use two baselines to assess the quality of the predicted binaural versions:

Rotated-Visual Rotated-Visual baseline assesses the performance of Points2Sound ($\mathcal{L}_{\mathrm{full}}$) when wrong visual

Table 1 Quantitative results of baselines and Points2Sound considering different mono input audio. For each method, we use rgb-depth 3D point cloud attributes and report the performance depending on the number of sources ($N = 1, 2, 3$), based on the average of the evaluation metrics. Average values for any number of sources are given by$\overline{\mathrm{d}_{\mathrm{ENV}}}and\overline{\mathrm{d}_{\mathrm{STFT}}}$

	$\mathrm{d}_{\mathrm{ENV}}$ ↓			$\mathrm{d}_{\mathrm{STFT}}$ ↓			$\overline{\mathrm{d}_{\mathrm{ENV}}}$ ↓	$\overline{\mathrm{d}_{\mathrm{STFT}}}$ ↓
	1	**2**	**3**	**1**	**2**	**3**		
s_m(true mono)								
Mono-Mono	0.387	0.403	0.388	26.719	26.414	26.747	0.392	26.626
Rotated-Visual	0.232	0.285	0.305	9.002	10.588	12.016	0.274	10.535
Points2Sound ($\mathcal{L}_{\mathrm{full}}$)	**0.173**	**0.248**	**0.280**	**3.297**	**6.645**	**9.080**	**0.233**	**6.340**
$s_m = s_b^L$								
Mono-Mono	0.148	0.155	0.159	7.472	6.997	6.951	0.154	7.14
Rotated-Visual	0.165	0.166	0.165	7.610	6.808	6.345	0.165	6.921
Points2Sound ($\mathcal{L}_{\mathrm{full}}$)	**0.054**	**0.103**	**0.130**	**0.636**	**1.820**	**2.604**	**0.095**	**1.686**
$s_m = s_b^L + s_b^R$								
Mono-Mono	0.142	0.166	0.178	4.046	4.112	4.058	0.162	4.072
Rotated-Visual	0.166	0.192	0.209	5.663	5.918	6.031	0.189	5.870
Points2Sound ($\mathcal{L}_{\mathrm{full}}$)	**0.015**	**0.073**	**0.114**	0.099	**0.762**	**1.521**	**0.067**	**0.794**
Points2Sound ($\mathcal{L}_{\mathrm{full}}$) (only-depth)	0.016	0.080	0.122	**0.082**	0.885	1.736	0.072	0.901
Points2Sound ($\mathcal{L}_{\mathrm{diff}}$)	0.015	0.090	0.125	0.153	1.205	1.832	0.076	1.063

The lowest errors are highlighted using bold font

information is provided. To this end, during testing, we rotate the 3D scene by $\pi/2$ in the horizontal plane of the listener's head.

Mono-Mono Mono-Mono baseline simply copies the mono input audio to both binaural predicted channels.

A quantitative comparison with a similar method for mono to binaural synthesis using visual information is provided in the Appendix.

4.3 Evaluation

For evaluation, we use 504 audio-visual 3D scenes with $N = 1, 2, 3$ sound sources. Audio-visual 3D scenes are generated using the test data set and following the procedure explained in Section 3.4. During the evaluation, augmentation operations are not applied and 10-s audio clips are selected.

Points2Sound input audio We consider three different types of mono mixture input signals for Points2Sound. Table 1 shows quantitative results of Point2Sound for the different types of mono mixture signals and number of sources.

First, we consider true mono mixture signals which come from the audio dataset, detailed in 3.3.2, where no HRTFs have been applied. In this case, Points2Sound improvement over the baselines is notable. This is especially observed in the $\overline{\mathrm{d}_{\mathrm{STFT}}}$ metric, where the Mono-Mono baseline achieves 26.626 while Points2Sound achieves 6.340. Despite the improvement, the binaural predictions are degraded and contain time-frequency artifacts that change the timbre characteristics of the original sound.

Second, we consider mono mixture signals which contain only the left binaural channel, i.e., $s_m = s_b^L$. Note that in this case, ITDs are not preserved in the mono mixture and the network has to shift differently between left and right channels in the binaural prediction. Results show that Points2Sound improves the evaluation metrics for all baselines especially when few sources are present. When $N = 1$ source, Points2Sound achieves a $\mathrm{d}_{\mathrm{ENV}}$ of 0.054 and a $\mathrm{d}_{\mathrm{STFT}}$ of 0.636, as opposed to 0.165 and 7.610 achieved by the Rotated-Visual baseline. It is important to remark that in comparison to the true mono input signal, the absolute values in $\mathrm{d}_{\mathrm{STFT}}$ show how much information is given to the model when the HRTFs are already applied in the mono mixture.

Third, following previous work [5–8], we consider mono mixture signals represented as the sum of the two spatial channels, i.e., $s_m = s_b^L + s_b^R$. In this case, the mixing of the channels creates a mono signal that loses spatial properties. But, the resultant mono signal preserves the correct ITDs from the binaural version. Results show that Point2Sound achieves the best results using this mono representation with a $\overline{\mathrm{d}_{\mathrm{ENV}}}$ of 0.095 and a $\overline{\mathrm{d}_{\mathrm{STFT}}}$ of 1.686. Note that the obtained quantitative results are similar to the ones achieved using the above approach, where mono mixture signals are represented as $s_m = s_b^L$.

For the following, when we refer to Points2Sound, we assume it has been trained using mono mixture signals represented as $s_m = s_b^L + s_b^R$.

3D point cloud attributes We evaluate how 3D point cloud attributes affect the binaural audio synthesis. Accordingly, we consider two types of 3D point cloud scenes: when 3D point cloud scenes consist of depth-only data and when 3D point cloud scenes consist of rgb-depth data. We use the term "only-depth" in Table 1 to report the performance when 3D point cloud scenes consist of depth-only data. Otherwise, we report the performance when 3D point cloud scenes consist of rgb-depth data.

We observe that Points2Sound benefits from the rgb information especially when multiple sound sources are present. For example with $N = 3$ sources, Points2Sound using rgb-depth features achieves a $\mathrm{d_{ENV}}$ of 0.114 and a $\mathrm{d_{STFT}}$ of 1.521, as opposed to 0.122 and 1.736 achieved by only depth features. But with $N = 1$ source, Points2Sound using depth features slightly outperform rgb-depth features, providing a $\mathrm{d_{STFT}}$ of 0.082 and a $\mathrm{d_{ENV}}$ of 0.016, as opposed to 0.099 and 0.015 We suspect that with a single source, depth features already provide straight information about the position of the source for accurate binaural synthesis. However, when multiple sources are present, rgb-depth features better distinguish each source which facilitates the binaural synthesis of the audio network. Informal listening corroborates that rgb-depth features help the model to recognize and locate multiple sources, as it provides more stable auditory images of the sound sources for the whole 10-s clips.

Number of sources in the 3D scene We are also interested in evaluating the quality of the predicted binaural audio depending on the number of sound sources present in the 3D scene. We observe that when $N = 1$ sources, Points2Sound provides perceptually convincing binaural predictions with a consistent performance across almost all examples.

When $N = 2$ sources, informal listening reveals that binaural audio predictions are convincing especially when the sources are located at the same side of the listener's head. Quantitative results show that Points2Sound using rgb-depth features achieves a $\mathrm{d_{ENV}}$ of 0.044 and a $\mathrm{d_{STFT}}$ of 0.284 when both sources are in the same side and a $\mathrm{d_{ENV}}$ of 0.084 and a $\mathrm{d_{STFT}}$ of 0.944 for other source position configurations. We also observe that in some cases, Points2Sound has difficulties when one of the two sources is located in front or behind the listener's head. This results in binaural predictions where the auditory image of the front/back source is not stable for the whole 10-s clip.

When N=3 sources, Points2Sound has difficulties to provide stable auditory images for every sound source for the whole 10-s clip.

Effect of the room We evaluate how Points2Sound performs under reverberant room conditions. To this end, we use binaural room impulse responses (BRIRs) to generate two test sets. Note that Points2Sound is trained using HRTFs measured in an anechoic room. The first test set is generated using the BRIRs measured in the studio room Calypso at TU Berlin [52]. The Calypso room has a volume of 83 m^3 and a reverberation time RT60 of 0.17 s at a frequency of 1 kHz. The second test set is generated using the BRIRs measured in the meeting room Spirit at TU Berlin [53]. The Spirit room has a rectangular shape with an estimated reverberation time RT60 of 0.5 s. In both rooms, BRIRs were measured using a KEMAR manikin (type 45BA) and a loudspeaker (Genelec 8250A) placed in front of the manikin at 2 m of distance. As the head movements of the manikins were measured from $-\pi/2$ to $\pi/2$, we simulate the binaural versions at the following discrete set of angular positions in the horizontal plane with no elevation $(\varphi_k, \theta) := (\frac{k\pi}{4}, 0)$ for $k = -2, \ldots, 2$. Finally, 504 audio-visual 3D scenes with $N = 1, 2, 3$ are generated using the same procedure explained in Section 3.4. Figure 4 shows a visual comparison of the performance of Points2Sound for different room acoustic conditions. The results are reported for a different number of sources using rgb-depth point cloud features. We observe that Points2Sound performance decreases as the testing of acoustic conditions diverge from the anechoic training. In this case, better binaural predictions are achieved in the room Calypso with a RT60 of 0.17 s as opposed to predictions in the room Spirit with a RT60 of 0.5 s. It is in high reverberant rooms where it becomes evident that Points2Sound has not been trained to model room effects such as reflections, especially for the low number of sources in the scene. For example with $N = 1$ sources, Points2Sound achieves an average of $\mathrm{d_{STFT}}$ of 4.491 in the Spirit room in contrast with 1.081 achieved in the Calypso room. Points2Sound performance in the dry room, i.e., the Calypso room, is closer to the anechoic performance, in particular when multiple sound sources are present. With $N = 3$ sources, Points2Sound achieves an average of $\mathrm{d_{STFT}}$ 2.180 in the Calypso room and 1.521 in the anechoic setting.

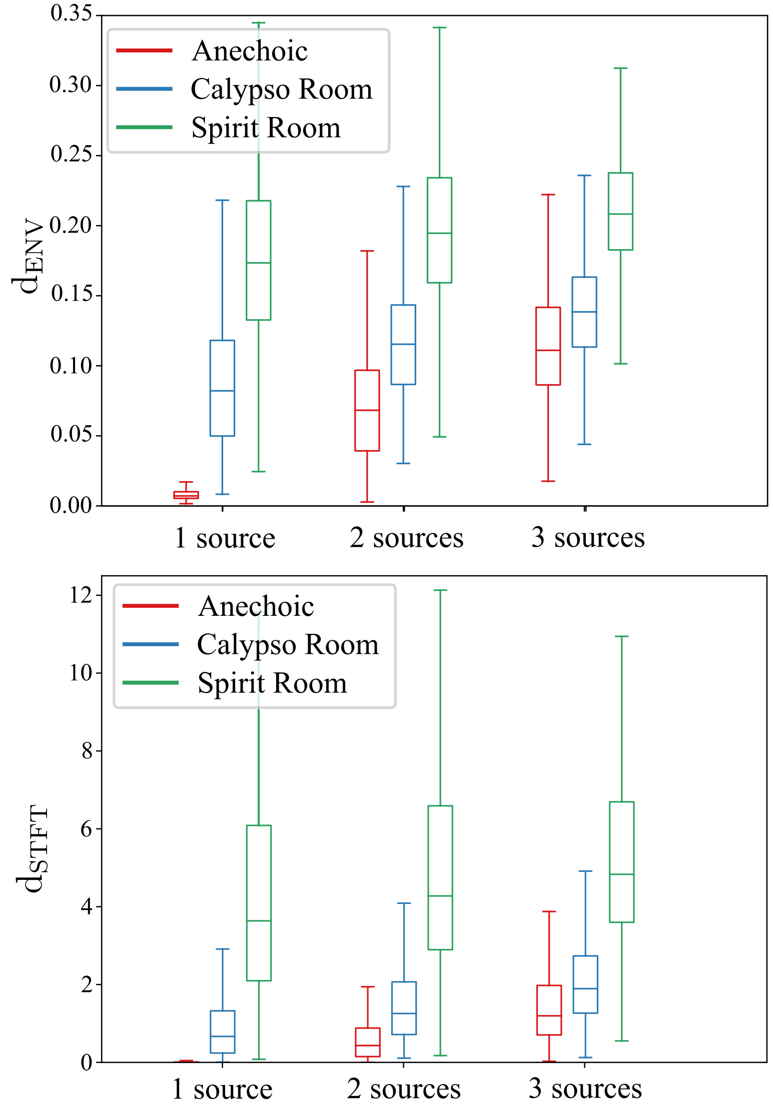

Fig. 4 Envelope Distance (d_{ENV}) and STFT Distance (d_{STFT}) of Points2Sound ($\mathcal{L}_{\text{full}}$), trained in anechoic conditions, for different rooms and number of sources in the 3D scene. The results are reported using rgb-depth point cloud features. For each number of sources, the box extension shows the first and third quartile of the data with a line at the median. The whiskers extending from the box show the range of the data

This suggests that Points2Sound potential applicability should consider dry rooms that resemble the anechoic training conditions.

Points2Sound loss function. We analyze Points2Sound performance depending on the learning objective. The initial proposed learning objective, referred to as $\mathcal{L}_{\text{full}}$, optimizes the parameters of the model to reduce the L1 loss between the estimated binaural $\hat{s_b}^{L,R}$ and the ground truth binaural $s_b^{L,R}$, i.e.

$$\mathcal{L}_{\text{full}} = \left\| s_b^{L,R} - \hat{s_b}^{L,R} \right\| \tag{10}$$

Note that in this case, Points2Sound predicts the full binaural signal, i.e., predicts both left and right binaural channels. Several methods in the literature propose to optimize the models by predicting the difference of the two binaural channels [5, 7]. To this end, we consider another loss function for Points2Sound, i.e., $\mathcal{L}_{\text{diff}}$, which optimizes the parameters to reduce the L1 loss between the estimated binaural difference channels $\hat{s_b}^{\text{diff}}$ and the ground truth binaural difference channels s_b^{diff}, i. e.

$$\mathcal{L}_{\text{diff}} = \left\| s_b^{\text{diff}} - \hat{s_b}^{\text{diff}} \right\| \quad (11)$$

where $s_b^{\text{diff}}(t) = s_b^L - s_b^R$. Note that when using $\mathcal{L}_{\text{diff}}$, Point2Sound is forced to learn the differences between the left and right binaural channels and predicts a one-channel signal $\hat{s_b}^{\text{diff}}$. Then, considering the mono signal represented as $s_m = s_b^L + s_b^R$, both predicted binaural channels are recovered as follows:

$$\hat{s_b}^L = \left(s_m + \hat{s_b}^{\text{diff}}\right)/2, \quad \hat{s_b}^R = \left(s_m - \hat{s_b}^{\text{diff}}\right)/2 \quad (12)$$

Results in Table 1 show that Points2Sound benefits from directly predicting the binaural signal using the $\mathcal{L}_{\text{full}}$ loss function as opposed to predicting the difference between binaural channels with the $\mathcal{L}_{\text{diff}}$ loss function. Using rgb-depth point cloud features, Points2Sound ($\mathcal{L}_{\text{full}}$) achieves a $\overline{\text{d}_{\text{ENV}}}$ of 0.067 and a $\overline{\text{d}_{\text{STFT}}}$ of 0.794 while Points2Sound ($\mathcal{L}_{\text{diff}}$) achieves 0.076 and 1.063, respectively. The poor performance obtained with the rotated-visual baseline indicates that Points2Sound strongly relies on the 3D scene to synthesize binaural audio and incorrect predictions are expected when using wrong visual information. In the following, we refer to Points2Sound assuming it has been trained using the $\mathcal{L}_{\text{full}}$ loss function.

4.4 Listening examples

We provide a Supplementary video with four listening examples where Points2Sound is applied to real-world data we record from expert musicians. We consider four challenging audio-visual scenes of $N = 2$ sources performing simultaneously in the same room. Specifically, two audio-visual scenes contain guitar and violin as sound sources while the other two contain doublebass and violin. The recorded audio fragments cover a variety of music styles (classical and jazz), tempi (vivace and lento), and dynamics (forte and piano). The 3D scenes of musicians are captured using Azure Kinect DK cameras while mono audio is captured using a Google Pixel 4 smartphone at a static position in the middle of the room. For each scene, the video shows the raw data first and then demonstrates the binaural predictions of Points2Sound. Despite the discrepancy between training data and real-world scenarios, the binaural predictions of Points2Sound show promising extrapolation ability.

5 Discussion

The work presented in this paper indicates the potential of using 3D visual information to guide multi-modal deep learning models for the synthesis of binaural audio from mono audio. By using 3D point clouds as visual information, the vision network has the ability to extract information about the 3D positions between the receiver and the sound sources in a scene to guide the binaural synthesis. By using 3D sparse convolutions, the network learns the correspondence between 3D structures found in local regions of the 3D space and audio characteristics.

When Points2Sound is trained using true mono signals that do not contain HRTF information, our proposed method introduces time-frequency artifacts that lead to degraded binaural predictions. This suggests that a significant amount of Points2Sound's capacity is needed to model the HRTF information. As a result, the model has more difficulties to synthesize accurate binaural sound. Considering one of the two binaural channels as input mono, i.e., $s_m = s_b^L$, Points2Sound achieves similar quantitative results as when considering the mono audio as the sum of the two binaural channels, i.e., $s_m = s_b^L + s_b^R$. Interestingly, the model trained using the sum of the two binaural channels as mono input provides encouraging extrapolation results when applied to real mono recordings, as demonstrated by the provided sound examples.

Results suggest that waveform-based approaches can provide convincing performance for the task of visually informed spatial audio generation without the need to rely on hand-crafted spectrograms as input. In addition, by operating in the waveform domain, our model synthesizes the signal directly. This is in contrast to spectrogram-based models which predict a mask to overcome the difficulties of directly predicting the spectrum, due to the large dynamic range of STFTs [5–8].

Our proposed model benefits from predicting the full binaural signal as opposed to the difference between binaural channels. This might be of relevance for other applications where visually informed models operating in the waveform domain are used to generate spatial audio.

Our proposed model benefits from using visual features extracted from rgb-depth point clouds to improve the binaural synthesis when multiple sources are present, in comparison with features extracted from depth-only point clouds. However, the fact that Points-2Sound can work with only-depth information may be beneficial in cases of low ambient light, where RGB sensors would fail to capture the scene, in contrast to LiDAR sensors that are still able to capture depth information. As mentioned above, we observe that in some cases Points2Sound predicts binaural versions where the auditory image of the sources is not stable. As this effect is mainly observed for cases with the number of sources $N > 1$, we suspect that this problem is related to the source separation capability of the audio network. To further investigate this phenomenon, a separate study on channel bleeding in source separation for different types of musical instruments would be required.

After analyzing the performance of Points2Sound under reverberant conditions, it is shown that the method could be applied to dry rooms that resemble the anechoic training conditions. However, a decreased performance is expected when the room acoustics conditions diverge from the anechoic training. The performance of Points2Sound in highly reverberant rooms, after retraining or fine-tuning the model using binaural room impulse responses that contain the influence of the room, remains to be studied.

6 Conclusion and future work

This work introduced Points2Sound, a multi-modal deep learning model capable of generating a binaural version from mono audio using a 3D point cloud scene as guidance. Points2Sound shows that 3D visual information can successfully guide the binaural synthesis while demonstrating that waveform-based approaches can provide convincing performance for the task of visually informed spatial audio generation.

Such models see increased interest for the generation of spatial audio in immersive applications. Recent portable devices, like smartphones, have the ability to capture 3D visual data from the environment using LiDAR or rgb-depth cameras. However, such devices have limited capabilities to record spatial audio from the sound sources. Having a recorded rgb-depth environment and its corresponding mono audio, our approach is a step towards synthesizing proper acoustic stimuli for the users navigating the virtual environment depending on their location and head position.

Future work could involve adding loudness into the learning process via predicting a reference sound level for each source. This would allow to infer also sound attenuation in 3D dynamic scenes.

Appendix

In the appendix, we show a quantitative comparison of Points2Sound ($\mathcal{L}_{\text{full}}$) with a recent spectrogram-based Mono2Binaural model from Gao et al. [7]. Mono2Binaural was designed to generate a binaural version from mono audio at 16 kHz using 2D visual information as guidance. In this case, we adapt it for audio recordings sampled at 44.1 kHz and 3D visual information as guidance. The original Mono2Binaural extracts visual features using a Resnet18 with dense convolutions while audio features and audio-visual analysis is performed using a U-Net. We use the same sparse Resnet18 from Points2Sound to extract the visual feature from the 3D scene. In addition, in order to resemble its original model, the last Resnet18 3×3×3 sparse convolution is implemented with $K = 512$ channels. Then, as in Mono2Binaural, the visual feature vector is replicated to match the spatial feature dimensions of the U-Net bottleneck and concatenated along the channel dimension. During training, we select 0.63-s clips of audio and compute the STFT using a Hann window of 23 ms and a hop length of 10 ms. Mono2Binaural considers mono inputs represented as $s_m = s_b^L + s_b^R$ and the learning objective is to predict the complex-valued spectrogram of the difference of the two binaural channels. Then, both predicted binaural channels are recovered as follows:

$$\hat{s_b}^L = \left(s_m + \hat{s_b}^{\text{diff}}\right)/2, \quad \hat{s_b}^R = \left(s_m - \hat{s_b}^{\text{diff}}\right)/2 \quad (13)$$

where $s_b^{\text{diff}}(t) = s_b^L - s_b^R$. We use the Adam optimizer and minimize the mean squared error loss function. During testing, Mono2Binaural uses a sliding window with a hop size of 50 ms to binauralize the 10-s audio clips.

Figure 5 shows a visual comparison of the performance of Points2Sound and Mono2Binaural for different sources when rgb-depth point cloud features are used. Table 2 shows quantitative results of both learning methods for different types of 3D point cloud attributes and number of sources.

In addition, we provide a second Supplementary video with listening examples where three audio-visual scenes from the test set with $N = 2$ sources are present. For each listening example, we first show the 3D point cloud scene and then provide the input mono audio, the Points2Sound and Mono2Binaural predicted binaural audios, and the ground truth binaural audio. The audio-visual scenes are selected to contain sound sources that are not located in the same side of the listener's head. Also, the scenes contain a variety of sound sources which play in the same frequency range in some fragments.

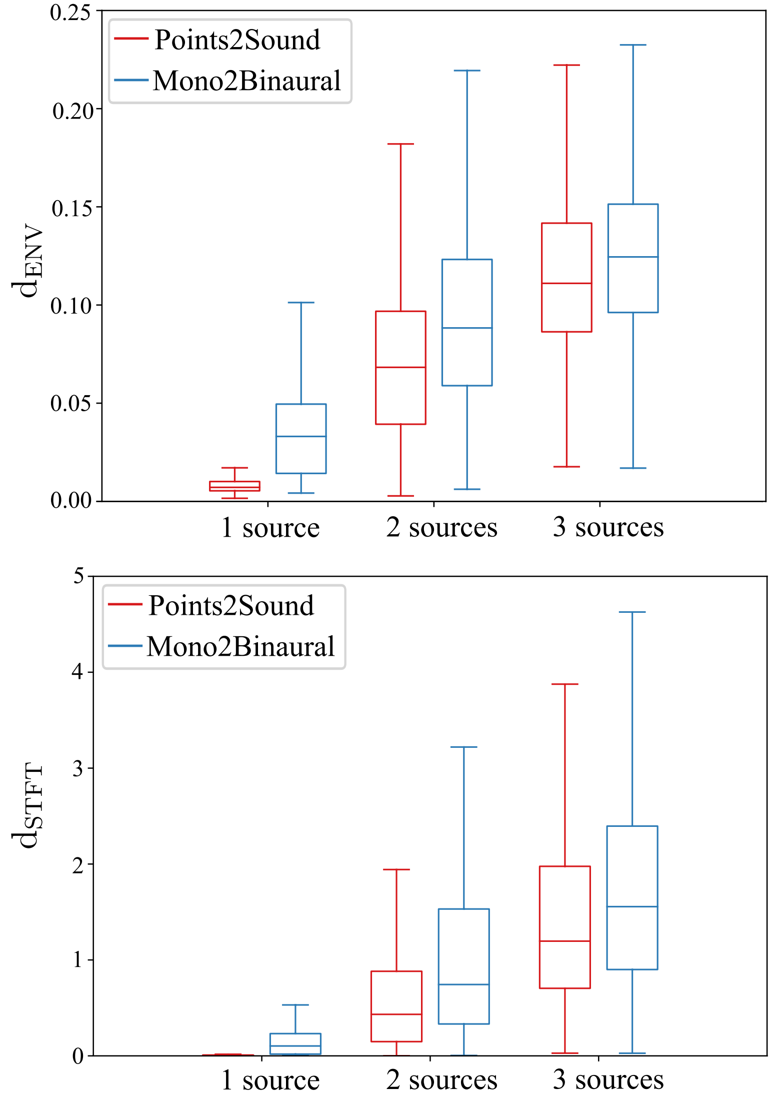

Fig. 5 Envelope Distance (d_{ENV}) and STFT Distance (d_{STFT}) of Points2Sound using $\mathcal{L}_{full}$ loss function and Mono2Binaural for different numbers of sources in the 3D scene. The results are reported using rgb-depth point cloud features. For each number of sources, the box extension shows the first and third quartile of the data with a line at the median. The whiskers extending from the box show the range of the data

Table 2 Quantitative results of Points2Sound and Mono2Binaural. For each method, we report the performance depending on the number of sources ($N = 1, 2, 3$) and the type of 3D point cloud attributes (depth or rgb-depth), based on the average of the evaluation metrics. Average values for any number of sources are given by$\overline{d_{ENV}}$and$\overline{d_{STFT}}$

	Visual features	d_{ENV} ↓			d_{STFT} ↓			$\overline{d_{ENV}}$ ↓	$\overline{d_{STFT}}$ ↓
		1	**2**	**3**	**1**	**2**	**3**		
Mono2Binaural [7]	Depth	0.038	0.101	0.132	0.192	1.300	2.029	0.090	1.174
	rgb-depth	0.036	0.094	0.126	0.213	1.142	1.858	0.085	1.071
Points2Sound	Depth	0.016	0.080	0.122	**0.082**	0.885	1.736	0.072	0.901
	rgb-depth	**0.015**	**0.073**	**0.114**	0.099	**0.762**	**1.521**	**0.067**	**0.794**

The lowest errors are highlighted using bold font

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13636-022-00265-4.

Additional file 1. First supplementary video

Additional file 2. Second supplementary video

Acknowledgements

The authors thank Alexander Mayer for technical support. We also thank the reviewers for comments leading to significant improvements.

Authors' contributions

FL proposed the idea, wrote the code, ran the experiments, and wrote the paper. VC and AH supervised the research. The authors read and approved the final manuscript.

Funding

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 812719.

Availability of data and materials

The captured 3D-video point cloud musicians dataset is available at https://zenodo.org/record/4812952.

The implementation of the proposed algorithm is available at https://github.com/francesclluis/points2sound. The repository provides the instructions to install the environment as well as scripts to train and test Points2Sound and Mono2Binaural. Furthermore, the repository includes the weights of the trained models.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 25 November 2021 Accepted: 14 December 2022
Published online: 29 December 2022

References

1. J.F. Culling, M.A. Akeroyd, Spatial hearing. Oxf. Handb. Audit. Sci. Hear. **3**, 123–144 (2010)
2. C.W. Robinson, V.M. Sloutsky, When audition dominates vision.Experimental psychology **60**(2), 113 (2013)
3. J. Blauert, *Spatial Hearing: the Psychophysics of Human Sound Localization* (MIT press, Cambridge, 1997)
4. E. Shaw, External ear response and sound localization. Localization of sound: Theory Appl. **3**, 30–41 (1982)
5. H. Zhou, X. Xu, D. Lin, X. Wang, Z. Liu, Sep-stereo: visually guided stereophonic audio generation by associating source separation. in *European Conference on Computer Vision* (Springer, Cham, 2020), pp. 52–69
6. K. Yang, B. Russell, J. Salamon, Telling left from right: learning spatial correspondence of sight and sound. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2020), pp. 9932–9941
7. R. Gao, K. Grauman, 2.5 d visual sound. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2019), pp. 324–333
8. Y.D. Lu, H.Y. Lee, H.Y. Tseng, M.H. Yang, Self-supervised audio spatialization with correspondence classifier. in *2019 IEEE International Conference on Image Processing (ICIP)* (IEEE, 2019), pp. 3347–3351
9. A. Défossez, N. Usunier, L. Bottou, F. Bach, Music source separation in the waveform domain. arXiv preprint arXiv:1911.13254 (2019)
10. J.F. Cardoso, Blind signal separation: statistical principles. Proc. IEEE **86**(10), 2009–2025 (1998)
11. S. Haykin, Z. Chen, The cocktail party problem. Neural Comput. **17**(9), 1875–1902 (2005)
12. A. Hyvärinen, E. Oja, Independent component analysis: algorithms and applications. Neural Netw. **13**(4–5), 411–430 (2000)
13. B.A. Olshausen, D.J. Field, Sparse coding with an overcomplete basis set: a strategy employed by v1? Vis. Res. **37**(23), 3311–3325 (1997)
14. D. Lee, S. Sebastian, Algorithms for non-negative matrix factorization, advances in neural information processing systems. in *Proceedings of the 2000 Conference* (MIT Press, Cambridge, 2000), pp. 556–562
15. N. Zeghidour, D. Grangier, Wavesplit: end-to-end speech separation by speaker clustering. IEEE/ACM Trans. Audio Speech Lang. Process. **29**, 2840–2849 (2021)
16. Y. Luo, N. Mesgarani, Conv-TasNet: surpassing ideal time-frequency magnitude masking for speech separation. IEEE/ACM Trans. Audio Speech Lang. Process. **27**(8), 1256–1266 (2019)
17. D. Samuel, A. Ganeshan, J. Naradowsky, Meta-learning extractors for music source separation. in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, 2020), pp. 816–820
18. N. Takahashi, Y. Mitsufuji, Densely connected multi-dilated convolutional networks for dense prediction tasks. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 993–1002
19. D. Stoller, S. Ewert, and S. Dixon, Wave-u-net: A multi-scale neuralnetwork for end-to-end audio source separation. in Proc. Int. Soc. MusicInf. Retrieval, 2018, pp. 334–340
20. F. Lluís, J. Pons, X. Serra, End-to-end music source separation: is it possible in the waveform domain? in *Interspeech* (ISCA, 2019)
21. C. Han, Y. Luo, N. Mesgarani, Real-time binaural speech separation with preserved spatial cues. in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, 2020), pp. 6404–6408
22. K. Tan, B. Xu, A. Kumar, E. Nachmani, Y. Adi, SAGRNN: self-attentive gated RNN for binaural speaker separation with interaural cue preservation. IEEE Signal Process. Lett. **28**, 26–30 (2020)
23. A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W.T. Freeman, M. Rubinstein, Looking to listen at the cocktail party: a speaker-independent audio-visual model for speech separation. ACM Trans. Graph. (TOG) **37**(4), 1–11 (2018)
24. A. Owens, A.A. Efros, Audio-visual scene analysis with self-supervised multisensory features. in *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), pp. 631–648
25. H. Zhao, C. Gan, A. Rouditchenko, C. Vondrick, J. McDermott, A. Torralba, The sound of pixels. in *Proceedings of the European conference on computer vision (ECCV)* (Springer, Cham, 2018), pp. 570–586
26. R. Gao, R. Feris, K. Grauman, Learning to separate object sounds by watching unlabeled video. in *Proceedings of the European Conference on Computer Vision (ECCV)* (Springer, Cham, 2018), pp. 35–53
27. H. Zhao, C. Gan, W.C. Ma, A. Torralba, The sound of motions. in *Proceedings of the IEEE International Conference on Computer Vision* (IEEE, 2019), pp. 1735–1744
28. C. Gan, D. Huang, H. Zhao, J.B. Tenenbaum, A. Torralba, Music gesture for visual sound separation. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2020), pp. 10478–10487
29. A. Richard, D. Markovic, I.D. Gebru, S. Krenn, G.A. Butler, F. Torre, Y. Sheikh, Neural synthesis of binaural speech from mono audio. in *International Conference on Learning Representations* (2020)
30. I.D. Gebru, D. Marković, A. Richard, S. Krenn, G.A. Butler, F. De la Torre, Y. Sheikh, Implicit HRTF modeling using temporal convolutional networks. in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, 2021), pp. 3385–3389
31. P. Morgado, N. Vasconcelos, T. Langlois, O. Wang, Self-supervised generation of spatial audio for 360° video. in *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Curran Associates Inc., Red Hook, 2018), pp. 360–370
32. C. Choy, J. Lee, R. Ranftl, J. Park, V. Koltun, High-dimensional convolutional networks for geometric pattern recognition. in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (IEEE, 2020), pp. 11227–11236
33. S. Xie, J. Gu, D. Guo, C.R. Qi, L. Guibas, O. Litany, Pointcontrast: unsupervised pre-training for 3d point cloud understanding. in *European conference on computer vision* (Springer, Cham, 2020), pp. 574–591
34. J. Gwak, C. Choy, S. Savarese, Generative sparse detection networks for 3d single-shot object detection. in *Computer Vision–ECCV 2020: 16th*

European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16 (Springer, Cham, 2020), pp. 297–313

35. C. Choy, J. Gwak, S. Savarese, 4d spatio-temporal convnets: Minkowski convolutional neural networks. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2019), pp. 3075–3084
36. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition. in *Proceedings of the IEEE conference on computer vision and pattern recognition* (IEEE, 2016), pp. 770–778
37. O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation. in *International Conference on Medical image computing and computer-assisted intervention* (Springer, Cham, 2015), pp. 234–241
38. A.V.D. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, Wavenet: a generative model for raw audio. arXiv preprint arXiv:1609.03499 (2016)
39. T. Jenrungrot, V. Jayaram, S. Seitz, I. Kemelmacher-Shlizerman, The cone of silence: speech separation by localization. in *Advances in Neural Information Processing Systems* (Curran Associates Inc., Red Hook, 2020)
40. Y.N. Dauphin, A. Fan, M. Auli, D. Grangier, Language modeling with gated convolutional networks. in *International conference on machine learning* (PMLR, 2017), pp. 933–941
41. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: an imperative style, high-performance deep learning library. in *Advances in neural information processing systems* (Curran Associates, Inc., Red Hook, 2019), pp. 8026–8037
42. Q.Y. Zhou, J. Park, V. Koltun, Open3D: a modern library for 3D data processing. arXiv:1801.09847 (2018)
43. D. Thery, B. Katz, Anechoic audio and 3d-video content database of small ensemble performances for virtual concerts. in *23rd International Congress on Acoustics* (German Acoustical Society (DEGA), 2019), pp. 739–46
44. H. Joo, T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Banerjee, T. Godisart, B. Nabbe, I. Matthews et al., Panoptic studio: a massively multiview system for social interaction capture. IEEE Trans. Patt. Anal. Mach. Intell. **41**(1), 190–204 (2017)
45. M. Kowalski, J. Naruniec, M. Daniluk, Livescan3d: a fast and inexpensive 3d data acquisition system for multiple kinect v2 sensors. in *2015 international conference on 3D vision* (IEEE, 2015), pp. 318–325
46. J.F. Montesinos, O. Slizovskaia, G. Haro, Solos: a dataset for audio-visual music analysis. in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)* (IEEE, 2020), pp. 1–6
47. F. Winter, H. Wierstorf, A. Raake, S. Spors, The two! ears database. in *Audio Engineering Society Convention 142* (Audio Engineering Society, 2017)
48. H. Wierstorf, M. Geier, S. Spors, A free database of head related impulse response measurements in the horizontal plane with multiple distances. in *Audio Engineering Society Convention 130* (Audio Engineering Society, 2011)
49. M. Otani, T. Hirahara, S. Ise, Numerical study on source-distance dependency of head-related transfer functions. J. Acoust. Soc. Am. **125**(5), 3253–3261 (2009)
50. Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3d shapenets: a deep representation for volumetric shapes. in *Proceedings of the IEEE conference on computer vision and pattern recognition* (IEEE, 2015), pp. 1912–1920
51. D.P. Kingma, J.L. Ba, Adam: a method for stochastic gradient descent. in *ICLR: International Conference on Learning Representations* (2015), pp. 1–15
52. H. Wierstorf. Binaural room impulse responses of a 5.0 surround setup for different listening positions (2016). https://doi.org/10.5281/zenodo.160761
53. N. Ma, T. May, H. Wierstorf, G.J. Brown, A machine-hearing system exploiting head movements for binaural sound localisation in reverberant conditions. in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, 2015), pp. 2699–2703

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.