Self-Supervised Monaural Audio Spatialization Using 2D Visual Correspondence

Anonymous ICCV submission

Paper ID 10151

Abstract

This work describes a method for converting monaural to 3D audio for an audio-visual data, using a visual correspondence from the 2D visual sequence. The main challenge faced by any deep learning based method is the availability of a dataset, which is extensive enough to ensure a good quality of the learning model. In this work, we propose a novel approach for synthetic scene construction for dataset generation, which otherwise is scarcely available for this problem statement. Following this, we train a modified U-NET *with* RESNET-18 *to perform a self-supervised audio-visual correspondence based learning, utilizing the synthetic dataset and apply the model on real-world videos. The quality of the model is evaluated with various types of scenes that are generated synthetically. We also show that our approach is more robust to aberrations in visual objects and has the ability to use the motion and position of video objects to predict a 3D/multi-channel audio from a mono audio that is given as input.*

1. Introduction

Audio recording and playback have gone through evolutionary and revolutionary changes from the first monophonic audio on mechanical devices in the late nineteenth century, through stereophonic sound on electronic devices, multi-channel audio, digital audio to 3D immersive audio experience of the present day. The use of immersive sound and directionally focussed audio, rendered through a multiplicity of acoustic transducers aims to envelop listeners in an experiential environment.

The importance of the quality of audio recording and rendering is obviously of great importance in enhancing the quality immersive experience for the audience. With the increasing proliferation of ultra high definition digital video at 4K and 8K resolutions on progressively larger screens aimed to give the audience an increasingly immersive visual experience and naturally, a matching enhancement in the quality of immersive sound playback has gained importance. Various immersive sound formats such as Dolby Atmos(R) [23] and Ambisonics [17] have emerged, which standardize the methods and formats for recording and authoring audio in a manner suitable for positioning audio sources accurately in 3D space, thereby enhancing the immersive watching experience for a user.

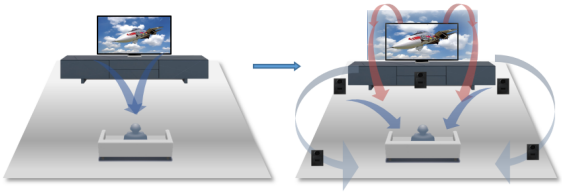

Figure 1. 3D and Multi channel audio remain to be extremely rare or involves high production costs and expertise to make it available for 2D Videos. The proposed approach infers a 3D visual sound by using visual correspondence from the 2D videos and monaural audio input.

In regular practice though, it is difficult to obtain audio-video content which has been authored for 3D audio due to the limited number of recording facilities available for such audio ,which results in audio-visual content for the ordinary home consumer being available with mono auditory audio or, typically, stereophonic audio. Moreover, playback of premium quality audio or audio-visual content mixed in specific 3D audio formats may require special hardware or premium solutions at the consumer's end.

In summary the problem is twofold. The first problem is one of availability of audio-visual content with audio authored and recorded in spatial audio (3D audio) formats, and second, the delivery of such content, if available, on bandwidth constrained streaming channels. The second problem is the lack of an automatic method of conversion of existing 2d audio-visual content, containing monaural or stereophonic audio, into spatial audio (3D audio), using a technique of upmixing from one or two audio channels to multiple accurately positioned audio sources.

In this work, we address these problems by proposing a novel deep learning approach, which uses an audio-visual stream with regular 2D Video and monaural or stereophonic audio to generate 3D audio. Our technique, naturally, needs a large number of 2D videos with 3D audio for the pur-

pose of training our deep learning model. This requirement poses its own challenge due to the extremely low availability of such audio-visual content and, therefore, we also propose a novel approach for synthetically generating such training data for the model. The trained model results in a completely automatic technique for conversion of conventional 2d audio-visual content to content with 3D audio. The trained model, in the prediction mode, can be utilized during various stages of audio-visual content delivery pipelines, e.g. during content creation, content streaming or on device playback.

The method proposed in this work is based on analysis of the motion or spatial arrangement in a video scene to compute the directionality and depth of video objects and associate computed information to the placement of the audio sources in 3D space. In general, a visual scene might include multiple visual objects with multiple sounds associated, mixed as a single source. We further propose a deep learning approach to overcome all the challenges of associating visual objects with different sounds and address complex scenarios like occlusion of objects.

In essence, we propose a deep learning based model pipeline, trained using a self-supervised learning method utilizing a synthetic dataset, to convert a mono sound to 3D sound using just a 2D visual correspondence.

2. Related work

Recent works describe deep learning for audio-visual separation of speech [4, 18, 1, 5], musical instruments [24], and other objects [7]. New problems are being explored, such as separation of on- and off-screen sounds [18], learning object sound models from unlabeled video [7], or predicting sounds per pixel [24]. All these methods exploit mono audio and visual cues to perform audio-visual source separation, whereas our requirement is to localize the audio sources in 3D space. In addition, rather than localization responsible for a given sound [14, 13, 25, 2, 24, 20, 21], our work aims to create 3D audio tracks for the given audio-video sequence.

The work patented in [22] describes steps of providing source depth data indicative of distance from the listener of at least one audio source, and upmixing the input audio to generate the 3D output audio using the source depth data. The source depth data is extracted from a stereoscopic 3D video and depends on individual sources of sound. This is not applicable directly to the scenario of our work which aims to extract visual correspondence including depth from conventional 2D Video.

The approach proposed in [16] converts monaural audio recorded by a microphone attached to a 360◦ video camera into spatial audio with audio sources located over the viewing sphere of the camera. The proposed technique [16] consists of an end-to-end trainable neural networks that separate individual sound sources and localize them on the viewing sphere, trained on 360◦ video frame sequences and corresponding audio inputs. The paper demonstrates the possibility of inferring the spatial location of sound sources based only on 360◦ video and a mono audio track.

Figure 2. Representative source collection like bike show and ted talks.

This work shows a promising technique using 360◦ videos and 3D audio from youtube or recorded manually to train the neural network model and predict 3D audio for 360◦ videos. This is different from the requirements of our work, in which 2D videos with monaural audio are used as input and synthetically generated 3D audio as desired output, for training the neural network model and to predict the 3D sound when a mono audio is provided as an input along with corresponding 2D video.

Very recent work on 2.5D sound [8] proposes a method to convert monaural audio in audio-visual sequences into binaural audio by using spatial cues derived from video frames. The work uses a deep convolutional neural network that learns to decode the monaural(single-channel) soundtrack into its binaural audio through transformations calculated using visual information about object and scene configurations. The visual stream information helps to convert the flat single channel audio into spatialized sound.

The method uses a binaural mic to manually record the binaural(2-channel) audios of musical instruments for training the neural network model, but does not extend in to 3d spatial audio which is agnostic to the configuration of channels or acoustic output transducers(speakers). The use of self-supervised learning method used for training the convolutional neural network is useful, but requires voluminous training data. The work relies on the existence of large number of binaural audio streams for training the neural network model. The same is not realistic for the case dealt with in our work, which aims to spatialize the sounds in a real-world video, due to the paucity of 3D audio streams with 2D videos available for training.

3. Approach

The preceding research [8], [16] though self supervised learning in nature, still relies on specially recorded videos with binaural mic, for inferring the spatial audio, which would heavily restrict the scope of model to diverse visual and audio objects.

Added to the above limitation, real-world videos contain a lot of motion either due to the visual objects moving within the scene or due to the camera movements, which is

not considered in FAIR-PLAY Dataset [8]

Our key idea is to combine multiple, single source audio-visual data, to simulate a real-world video, with the objective of controlling both the visual positioning and audio spatiality together. We propose a novel approach for the creation of such a dataset, synthetically from a very few readily available 2D videos online, and further use it for training the 3D audio model(Sec. 3.3.6).

3.1. Problem complexity estimation

First and foremost, we have identified the complexity of converting a mono to 3D audio by analyzing many real-world videos, and segregated them into categories with the following criteria $P_1 \rightarrow$ single visual and audio source in the scene, $P_2 \rightarrow$ multiple visual and audio sources, but each audio-visual object belong to different classes as per [15] $P_3 \rightarrow$ multiple visual and audio sources, with one or more objects of same class as per [15]. Apart from the above three categories, we have identified other natural phenomena generally observed in real-world videos and labelled them as $P_4 \rightarrow$ Occlusion, background and ambient sounds, $P_5 \rightarrow$ Asymmetric and symmetric camera angles

3.2. Source video collection and annotation

3.2.1 Source video collection

We have collected 500 high quality video sources which are readily available on Youtube(Refer Fig. 2). All the sources are roughly equal in proportion, and belongs to the following classes *Person-Male, Person-Female, Vehicle-Car, Vehicle-Bus, Vehicle-Airplane, Vehicle-Boat, Vehicle-Train, Vehicle-Bike, Sentient-Bird, Sentient-Animal* .

3.2.2 Annotation

We developed a custom annotation tool, to support time segment validation that fits the criteria of single audio and visual source. The dashboard also supports drawing bounding boxes around the visual object, so that annotators can specifically point to the source of the sound in a scene type of P_3 or P_4(Refer 3.1). After the annotation, the final data for each video includes, one or more time segments which contain single audio-visual source and also $\geq$1 bounding boxes in a scene type of P_3 or P_4. Though the annotation of bounding boxes in not performed on all the frames, we employ interpolation techniques to estimate the bounding box for the remaining intermediate frames. Let us call this interpolated data as $I^T_{(l,t,r,b)}$, where I^T contains interpolated left, top, right, bottom co-ordinates of bounding box of the visual object at a particular time $t \in [0,T]$ of duration $\mathbf{T}$.

Let us call the sequences obtained after this processing as $V^n_{selected}$ where $n \in [0,N]$

3.3. Synthetic dataset generation

From the annotated sources, frames with fps=2, and the corresponding audio are extracted for further processing. All the frames are resized to a fixed width and height $W' \times H'$. A collection of N$\geq$1 sources from (Sec. 3.2.2) are selected for a synthetic video generation.

3.3.1 Visual object extraction

As the first step of synthetic video generation, we first use a pre-trained Mask-RCNN [11], to extract the location of the visual objects of specific category(Sec. 3.2.1) from each frame in the annotated visual sequence.

For solo visual-audio source in the scene(P_1), we directly use the Mask-RCNN bounding boxes $M^t_{(l,t,r,b)}$ of the intended category with highest score for Visual Object extraction. For multi visual but single audio sources, Intersection over union(**IOU**) is performed between I^t(Sec 3.2.2) and M^t to estimate the visual source of the audio.

The extracted visual object at any time **t** in the visual sequence, is padded and centered on a transparent background, which has same dimensions of $W' \times H'$.

3.3.2 Random path generation

For each visual sequence of the N audio-visual sources to be mixed, a corresponding random path is generated. To simulate the motion, various parameters like variable step size, additional environment boundaries for occlusion, are introduced to make the motion vector more realistic as a real-world video.

Base randomness estimation Based on the fps of extracted frames and the time T' of the Synthetic scene to be exported, the total random points P to be calculated are identified(Fig. 4). Only a fraction F of P points are calculated using random walk in x, y, z dimensions independently and are confined within standard image dimensions $W' \times H'$ along with a defined Z' limits. For simplicity we assume scene depth Z' as $\pm \max(W', H')/2$. While constructing the above 3D walk, a random step size within a threshold is used, at each time instance to control the variable step length, which introduces the variable pace of movement. Calculating all the data points using random walk is avoided to minimize too much randomness which is highly unlikely of any moving object in an actual video. Let us call this base random vectors as $\vec{B}^n_{x,y,z}$, where N is total sequences to be mixed, x,y,z are the axis of the 3D random walk and $n \in [0,N]$.

Handling occlusion Occlusion is generally caused when visual object leaves and enters the scene, or when overlapped by a different object. The former case of occlusion

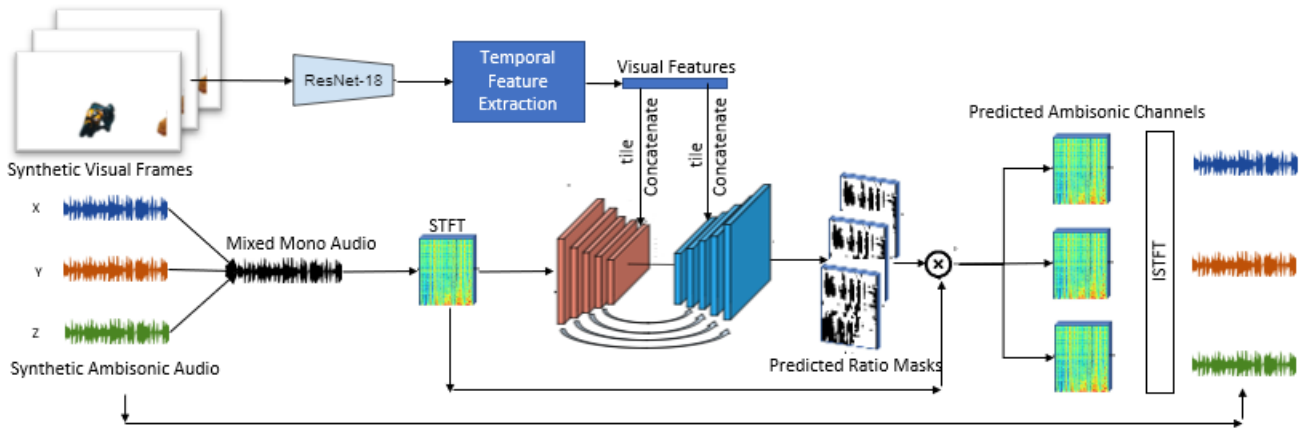

Figure 3. Our MONOTO3D deep network takes a mixed monaural audio and its accompanying visual frame as input, and predicts ambisonic audio channels as output that satisfies the visual spatial configurations. A pre-trained ResNet-18 network, on ImageNet is used to extract visual features, which is followed by a temporal feature extraction network for motion estimation, and a U-NET to extract audio features and perform joint audio-visual analysis. We predict a complex ratio mask for the each ambisonic audio signal, then multiply it with the complex STFT of input mono audio to restore the ambisonic audio channels. During prediction on real-world videos, the input is single-channel monaural audio and visual frames of the scene for which the background is removed.

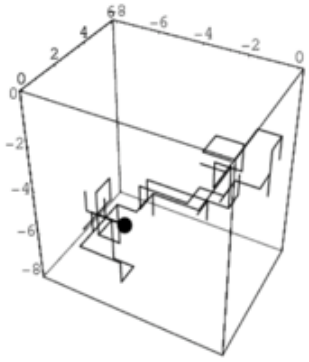

Figure 4. Representative image for 3D random walk.

is handled by introducing a deviation factor D, which is additional boundary that random walk point can move to. The latter is handled in Sec. 3.3.5

3.3.3 Motion vector estimation

For visual frames First the motion vector $\vec{V}^{N}_{x,y,z}$ for each sequence $V^{n=0...N}_{selected}$, is calculated by interpolating for remaining fraction $F' = P - F$ from $\vec{B}^{N}_{x,y,z}$. The motion vector thus obtained, provides x,y,z positions for the $V^{t}_{selected}$ for each time instant $t \in [0, T']$ where T' is the duration of the synthetic scene.

For audio The motion vector $\vec{A}^{n=0...N}_{x,y,z}$ for the audio is obtained from $\vec{V}^{N}_{x,y,z}$ by a second interpolation, because the samples defined by the sampling rate in audio is different from the samples defined by frame rate in video. The motion vector thus obtained, provides x,y,z positions for audio of $V_{selected}$ at each time instant $t \in [0, T']$.

3.3.4 Motion vector projection

Visual frames For a source n of N with the corresponding motion vector $\vec{V}^{n}_{x,y,x}$, the visual object $V^{n,t}_{selected}$ is centered at $\vec{V}^{n,t}_{x,y}$, where $t \in [0, T']$, x, y represents x and y co-ordinates of n^{th} visual object at t and T' is the duration of synthetic scene to be exported. To project depth on to the visual object we resize the $V^{n,t}_{selected}$ by calculating new width and height of each frame of n $\in [0, N]$ at each time instant t $\in [0, T']$ as following,

$$W^t = W' \times tan^{-1}(\vec{V}^{n,t}_{z}), \quad H^t = H' \times tan^{-1}(\vec{V}^{n,t}_{z}) \tag{1}$$

Audio For spatializing the audio, we use Ambisonic encodings [17] to construct the spatial audio channels $X_{ambi}, Y_{ambi}, Z_{ambi}$, using $\vec{A}^{n}_{x,y,z}$. First $\vec{A}^{n}_{x,y,z}$ is normalized and transformed to the range of $\{-1, 1\}$ in all the x,y,z dimensions. Assuming $W' \approx H' \approx Z'$, the normalized spatial vectors for audio lie within the range of unit cube, circumscribing a unit omni-sphere that the ambisonic encoding equations rely on. To normalize the motion to the unit omni-sphere, we further divide $\vec{A}^{n}_{x,y,z}$ by $\sqrt{3}$, which transorms the points to the cube inscribed in the omni-sphere. Refer Fig. 5

$$\vec{A}^{n}_{d\in\{x,y,z\}} = \frac{A^{n}_{d\in\{x,y,z\}}}{\sqrt{3}} \tag{2}$$

The final motion vectors thus obtained, after Eq. 2 can be used directly for obtaining as suggested by the authors of

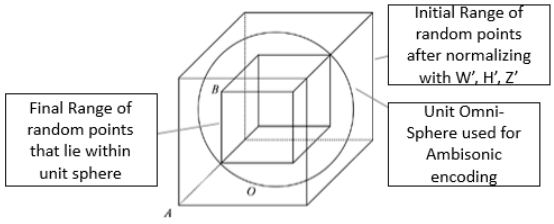

Figure 5. Transformation of random walk points to unit-onmisphere for ambisonic audio compatibility.

[17] as following

$$\begin{aligned} X^n_{ambi} &= \vec{A}^n_x \otimes A^n_{mono} \\ Y^n_{ambi} &= \vec{A}^n_y \otimes A^n_{mono} \\ Z^n_{ambi} &= \vec{A}^n_z \otimes A^n_{mono} \end{aligned} \tag{3}$$

where $\vec{A}^n_x, \vec{A}^n_y, \vec{A}^n_z$ are audio object positions of n^{th} source from Eq. 2, A^n_{mono} is the mono audio of the n^{th} source and $\otimes$ is the element-wise multiplication.

3.3.5 Synthetic video and audio mixing

The N sources of visual frames and audios to be mixed are of varying durations. To mix such misaligned sequences, we follow the following approach. For each video sequence, we first identify the length of the sequence L^n. Let the duration of the Synthetic scene to be exported is T' and that of the n^{th} source be T^n.

If $T^n > T'$, frames and audio with duration T' is randomly sampled from the visual and audio sequence of the n^{th} source. $\tilde{T}^n$ is the new duration of n^{th} source. A padding is then calculated to add the empty sequence at the beginning and the end of both visual and audio sequences.

$$\begin{aligned} S^n &= rand(0, T' - \tilde{T}^n) \\ E^n &= T' - (S^n + \tilde{T}^n) \end{aligned} \tag{4}$$

where S^n, E^n is the duration of empty visual and audio samples to be padded at the beginning and end of the n^{th} visual and audio sequences to obtain a duration T'.

Video mixing After performing the transformations in Eq. 1, and further random sampling and padding in Eq. 4, all the visual sequences from N sources are superimposed on one another framewise, where frame with highest Z'_t is on the top. Frames with negative Z'_t value are ignored, indicating a scenario of Occlusion when the visual object is behind the viewer and is out of the camera scope. Let the final sequence of the synthetic frames obtained be

$$V^T_s \rightarrow Synthetic\ Visual\ Frames \tag{5}$$

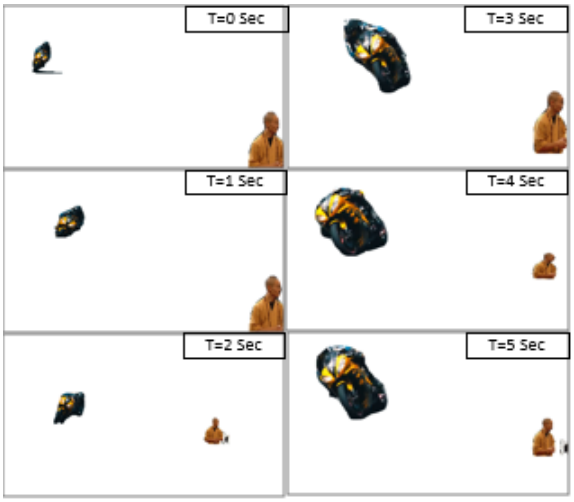

Figure 6. Example of synthetically created scene.

3D audio mixing After random sampling and padding the ambisonic channels of each source n in Eq. 3, using the Eq. 4, the final ambisonic channels are as following

$$X^s_a = \sum_{n=1}^{N} X^n_a, \quad Y^s_a = \sum_{n=1}^{N} Y^n_a, \quad Z^s_a = \sum_{n=1}^{N} Z^n_a \tag{6}$$

Mono audio mixing Similar to the above, random sampling and padding the monaural audio of each source A^n_{mono}, using the Eq. 4 is performed. The final monaural audio of the synthetic scene is

$$A^s_a = \sum_{n=1}^{N} A^n_{mono} \tag{7}$$

Finally visual frames from Eq. 5, mono audio from Eq. 7 are used as inputs and STFT of ambisonic channels in Eq. 6 as targets for our MONOTO3D(Sec. 3.3.6) network in training phase.

3.3.6 MONOTO3D network

The preceding research in Audio source Separation [9][24] and binaural audio generation [8], clearly proved that U-NET [19] based encoder decoder networks perform best for sound localization, separation and generation. So we too adopt, modified UNET for 3DAudio Visual Model for 3D Audio generation. The network takes visual frames and mono audio as input and predict *First Order Ambisonic(FOA)* channels.

Unlike the MONO2BINAURAL network [8], our network is targeted to predict 3D audio for long durations, and to support problem complexity P_4(Sec 3.1). Also our synthetic scene generation(Sec 3.3) approach is capable of

generating moving audio-visual objects as well. So, in order to guide the model better, we use multiple visual frames for visual attention and longer audio duration.

For the Audio network, we adopt 7-layered U-NET architecture.

Visual correspondence network We use the first seven layers of pretrained RESNET-18 [12] as image feature extraction network, followed by a series of convolution and batch normalization units for temporal feature extraction from the image frames. The feature extraction layers takes visual frames as input of size $B * S_{temporal} \times C \times H' \times W'$, where $B, S_{temporal}$ are the batch size and Number of temporal frames for the visual sequence, and C, H', W' are the channels, height and width of each image.

The output of the feature extraction module is then reshaped from $B * S_{temporal} \times 256 \times 19 \times 19$ to $B \times S_{temporal} * 256 \times 19 \times 19$ and passed to temporal feature extraction module to get the temporal spatial information for the visual objects.

The temporal feature extraction module consists of three convolutions and batch normalization units. The input channels for these units are configured as $256 \times S_{temporal}$, $128 \times S_{temporal}$, $64 \times S_{temporal}$, where $S_{temporal}$ is the number of frames in visual sequence. The output of the temporal feature extraction module has $B \times 1024 \times 19 \times 19$. which is passed through adaptive pooling for tile and concat operation for the Audio Network.

Audio network Similar to the prior research [8], on the audio side, we adopt a 7 layered U-NET [19] style architecture which has 7 convolutions (or down-convolutions) and 7 de-convolutions (or up-convolution) with skip connections in between. The U-NET encoder-decoder is adopted as it is ideal for dense-prediction task and maintaining the shape integrity of the input mono audio and output Ambisonic channels.We pass the **STFT** $\tilde{ST_{mono}}$ of Mono Audio from Eq. 7 as the input to the Audio network. Unlike the tile and concat approach used in [8], we use adaptive pooling implemented in PyTorch, to make the layers compatible for concatenation.To better guide the Audio Network using visual correspondence, our tile and concatenation happens at two layers of the U-NET, the first at ConvLayer-7 and the second at UpConvLayer-5. For pooling strategy we use adaptive max-pooling for first concatenation, and adaptive average-pooling for the second concatenation. The input channels of the U-NET are adjusted accordingly to make the tile and concat compatible. The targets for the model are the corresponding complex ratio masks for the ambisonic channels, which when multiplied with the input audio STFT gives the ambisonic channel STFT in complex domain.

Loss function We use a combination of spectral convergence loss [3] and log-scale STFT magnitude loss[3]. For spectral convergence loss as well as log STFT loss, we use only the STFT magnitudes for both predicted and target ambisonic channels to calculate the loss.

In addition to the above loss functions we also propose a third loss called Spatial Loss which is spectral convergence loss for STFT in complex domain.

$$L_{\tilde{S}} = \| \, |S_r| - |\tilde{S}_r| \, \|_F \; / \; \| \, |S_r| \, \|_F + \| \, |S_i| - |\tilde{S}_i| \, \|_F \; / \; \| \, |S_i| \, \|_F, \quad (8)$$

where $\| \cdot \|_F$ is Frobeniuos normalization, S_r and $\tilde{S}_r$ are real part of STFT of target and predicted audio signals in complex domain, whereas S_i and $\tilde{S}_i$ are the imaginary part.

$$L_{combined} = \sum_{n}^{n \in [X,Y,Z]} L_S^n + L_L^n + L_{\tilde{S}}^n, \quad (9)$$

where $L_{combined}$ is the total loss across the three ambisonic channels X, Y, Z. L_S is the spectral convergence loss, L_L is the log-scale STFT loss and $L_{\tilde{S}}$ is the Spatial loss of each channel.

4. Implementation Details

4.1. Experiments

Around 0.3 Million scenes were created satisfying multiple criteria discussed below. We chose the number of sources N to be mixed, in the range [1,3] randomly for each synthetic scene.This would make sure the input data is diverse with both single and multiple audio-visual sources. Though for training phase we restrict the scope of synthetic scenes to only P_1&P_2(refer 3.1), we analyse the performance on P_3 scene types as well using this pre-trained model in evaluation phase.

Our MONOTO3D network is implemented in PyTorch. We used 3 $\times$ Nvidia Tesla V100 GPUs, with model size occupying about 16Gb on each GPU. Our model is trained with a batch size of 64, with 8 visual frames per each sample for visual correspondence. For all experiments, we resample the audio at 16kHz and **STFT** with Hann window of duration 25ms, hop duration of 10ms, and FFT size of 512. For MONOTO3D training, we sample audio segments randomly, of duration 4s from each 10s Synthetic Scenes. Unlike very small audio samples chosen in [8], [16], we process 4 seconds of audio and the corresponding visual frames at a time.

A learning rate of 1×10^{-5},1×10^{-4},1×10^{-4} is used for feature extraction, temporal feature extraction and the audio network respectively. The model was trained for 45 hrs.

4.2. Prediction on real-world videos

For real-world videos, we pre-process the video by extracting the frames from the videos and remove the

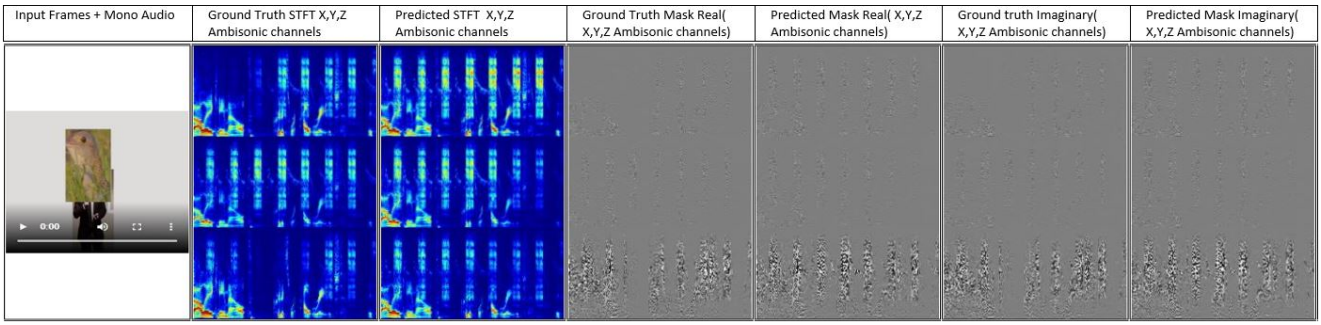

Figure 7. Visualization of input and target STFT magnitude heat map, ratio masks of the ambisonic channels in complex domain. Our model is able to predict the spatial audio using visual correspondence, represented by the X,Y,Z channels of ambisonic audio.

a. b.

Figure 8. Background removed from real-world videos before predicting 3D audio. [Please refer supplementary material for results on real-world videos with mono audio].

background. To remove the background we use Mask-RCNN [11], to identify and filter the specific visual objects defined in Sec. 3.1. The sequences thus obtained will be visually similar to the synthetic scenes that were used to train the model. The model predicts the complex ratio masks for each ambisonic channel. Each of these ratio masks are multiplied element wise with the input STFT to get the STFT of the corresponding ambisonic channel. The waveform for each ambisonic channel can then be generated by performing ISTFT [10].

Using the Ambisonic decoding techniques, we can further convert the ambisonic audio to standard 7.1 or 5.1 sound layouts [17]. The model can also be used for upmixing stereo audio, by first down-mixing the stereo to mono.

4.3. Mono To 3D Generation Accuracy

Following[8], we prepare the same baselines but for 3D Ambisonic Audio as following.

- **Audio-Only:** To determine if visual information is essential to perform MONOTO3D conversion, we remove the visual frames and implement a baseline using only audio as input. All other settings are the same except that only blank visual frames are passed to the network instead of the the original frames.
- **Flipped-Visual:** During testing, we flip the accompanying visual frames of the mono audio to perform prediction using the wrong visual information.
- **Mono-Mono:** A straightforward baseline that copies the mixed monaural audio onto the 4 channels to create a fake Ambisonic audio.

Method	$\mathcal{D}_{\{\text{STFT}\}}$	$\mathcal{D}_{\{\text{ENV}\}}$	Loss
Audio-Only	0.85	0.15	7.96
Flipped-Visual	0.81	0.12	7.08
Mono-Mono	1.29	0.27	-
MonoTo3D (Ours)	**0.35**	**0.11**	**4.94**

Table 1. Our results compared to the baselines when maximum number(N) of Audio-Visual objects in Synthetic scene are 2 and are of P_1 or P_2 scene types of equal proportion. Our model significantly outperforms all the baselines with a standard error of 1×10^{-2}

We evaluate the quality of our 3D audio by using the following common metrics used in both the prior researches [8]and [16]

1) **STFT Distance:** The euclidean distance between the ground-truth and predicted complex spectrograms Ambisonic channels X, Y, Z:

$$\mathcal{D}_{\{\text{STFT}\}} = |\mathbf{S}^X - \tilde{\mathbf{S}}^X|^2 + |\mathbf{S}^Y - \tilde{\mathbf{S}}^Y|^2 + |\mathbf{S}^Z - \tilde{\mathbf{S}}^Z|^2 \quad (10)$$

2) **Envelope (ENV) Distance:** Following [8] and [16], we take the envelope of the signals, and measure the euclidean distance between the envelopes of the ground-truth Ambisonic channels and the predicted Ambisonic signals. Let $E_{x(t)}$ denote the envelope of signal $x(t)$. The envelope distance is defined as:

$$\mathcal{D}_{\{\text{ENV}\}} = |E_t^x - \tilde{E}_t^x|^2 + |E_t^y - \tilde{E}_t^y|^2 + |E_t^z - \tilde{E}_t^z|^2 \quad (11)$$

Results. Table 1, 2, 3, 4 shows the performance of our models vs the baselines.

We created a separate set of synthetic scenes to perform the evaluations. Table 1 shows our evaluation results on synthetic scenes containing 1<=N<=2 visual objects at

Method	$\mathcal{D}_{\{STFT\}}$	$\mathcal{D}_{\{ENV\}}$	Loss
Audio-Only	0.86	0.16	8.15
Flipped-Visual	0.83	0.12	8.32
Mono-Mono	1.21	0.26	-
MonoTo3D (Ours)	**0.38**	**0.11**	**5.40**

Table 2. Our results compared to the baselines when maximum number(N) of Audio-Visual objects in Synthetic scene are 2 and are of either P_1, P_2 or P_3 scene types of equal proportion.

Method	$\mathcal{D}_{\{STFT\}}$	$\mathcal{D}_{\{ENV\}}$	Loss
Audio-Only	0.62	0.14	8.07
Flipped-Visual	0.55	0.12	7.19
Mono-Mono	0.75	0.14	-
MonoTo3D (Ours)	**0.33**	**0.11**	**5.68**

Table 3. Our results compared to the baselines when maximum number(N) of Audio-Visual objects in Synthetic scene are 3 and are of either P_1, P_2 scene types of equal proportion.

Method	$\mathcal{D}_{\{STFT\}}$	$\mathcal{D}_{\{ENV\}}$	Loss
Audio-Only	0.63	0.14	8.01
Flipped-Visual	0.57	0.12	7.30
Mono-Mono	0.89	0.19	-
MonoTo3D (Ours)	**0.36**	**0.12**	**5.82**

Table 4. Our results compared to the baselines when maximum number(N) of Audio-Visual objects in Synthetic scene are 3 and are of either P_1, P_2 or P_3 scene types of equal proportion.

any time and also of P_1 or P_2 scene type only. Table 2 shows our evaluation results on synthetic scenes containing 1<=N<=2 visual objects at any time and the Synthetic scene is either of P_1 or P_2 or P_3 scene type. Table 3 shows our evaluation results on synthetic scenes containing 1<=N<=3 visual objects at any time and the Synthetic scene is of P_1 or P_2 scene type only. Table 4 shows our evaluation results on synthetic scenes containing 1<=N<=3 visual objects at any time and the Synthetic scene is either of P_1 or P_2 or P_3 scene type.

In all the experiments, our model significantly outperformed the baselines(lower the values, better the spatialization) in both $\mathcal{D}_{\{STFT\}}$ & $\mathcal{D}_{\{ENV\}}$.

Figure 7 shows the visualizations of the expected and predicted STFT Magnitude heat maps along with the Ratio Masks generated by the model. As we can observe the results are quite convincing with clear and accurate similarity between the ground truth and predicted components across all the three Ambisonic channels. This shows that the model was able to use the visual correspondence to independently predict the X,Y,Z channels indicating an omni-directional audio as per the specifications of Ambisonic [17].

Though the model was only trained with P_1 and P_2 type scenes during training, our results in Table 2 and Table 4 proves that model was able to generalize the dynamics of visual correspondence to audio in an unknown P_3 scenes as well.

Limitations. Apart from the above experiments, we have also conducted additional experiments for P_4 scenes as well, particularly for background and ambient sounds. Though the model was able to handle the occlusion smoothly, the placement of background and ambient sounds was not accurate given the unavailability of the visual correspondence. One particular improvement that can be made is using ambient background music and sounds during synthetic dataset generation, and place all these ambient sounds behind the listener assuming the listener is at the center of the soundfield.

5. Future work

As future work, we wish to extend our work in the following areas.

For the scope of this research we haven't covered the P_5 category scenes involving symmetric and asymmetric camera angles. We wish to explore image warping and perspective shift techniques, while creating the synthetic scenes. Inspired from [6], we wish to explore optical flow to improve our P_3 type scenes in future.

3D audio generation research is still in its early stage and significant methodologies have yet to be identified, that can properly capture the spatial perception of the generated audio. We would like to employ more efficient strategies for estimating the audio-visual perception, be it algorithmic or user based evaluations.

6. Conclusion

We presented an approach to convert single channel audio into First Order 3D Ambisonic audio with visual correspondence from a 2D video. The predicted 3D ambisonic sound offers a more immersive audio experience and is also speaker agnostic. To the best of our knowledge we believe we are the first to propose an end to end system for 3D or multi channel audio for a 2D video. We believe our MONOTO3D network combined with Synthetic scene generation approach opens up wide possibilities for research, thereby providing a better viewing experience of the content in any device, which till date has been restricted to Multiplexes or costly solutions.

References

[1] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. The conversation: Deep audio-visual speech enhancement. *CoRR*, abs/1804.04121, 2018. 2

[2] R. Arandjelovic and A. Zisserman. Look, listen and learn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 609–617, 2017. 2

[3] Sercan Ömer Arik, Heewoo Jun, and Gregory F. Diamos. Fast spectrogram inversion using multi-head convolutional neural networks. *CoRR*, abs/1808.06719, 2018. 6

[4] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *arXiv preprint arXiv:1804.03619*, 2018. 2

[5] Aviv Gabbay, Asaph Shamir, and Shmuel Peleg. Visual speech enhancement using noise-invariant training. *CoRR*, abs/1711.08789, 2017. 2

[6] Chuang Gan, Hang Zhao, Peihao Chen, David D. Cox, and Antonio Torralba. Self-supervised moving vehicle tracking with stereo sound. *CoRR*, abs/1910.11760, 2019. 8

[7] Ruohan Gao, Rogério Schmidt Feris, and Kristen Grauman. Learning to separate object sounds by watching unlabeled video. *CoRR*, abs/1804.01665, 2018. 2

[8] Ruohan Gao and Kristen Grauman. 2.5d visual sound. In *CVPR*, 2019. 2, 3, 5, 6, 7

[9] Ruohan Gao and Kristen Grauman. Co-separating sounds of visual objects. In *ICCV*, 2019. 5

[10] D. Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. In *ICASSP '83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 8, pages 804–807, 1983. 7

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 3, 7

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[13] John Hershey and Javier Movellan. Audio vision: Using audio-visual synchrony to locate sounds. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 2000. 2

[14] E. Kidron, Y. Y. Schechner, and M. Elad. Pixels that sound. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 88–95 vol. 1, 2005. 2

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. 3

[16] Pedro Morgado, Nuno Vasconcelos, Timothy R. Langlois, and Oliver Wang. Self-supervised generation of spatial audio for 360 video. *CoRR*, abs/1809.02587, 2018. 2, 6, 7

[17] FirstName F. LastName Ortolani. Introduction to ambisonics: A tutorial for beginners in 3d audio, rev 2015a. 1, 4, 5, 7, 8

[18] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. *European Conference on Computer Vision (ECCV)*, 2018. 2

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. 5, 6

[20] Arda Senocak, Tae-Hyun Oh, Junsik Kim, Ming-Hsuan Yang, and In So Kweon. Learning to localize sound source in visual scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[21] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *ECCV*, 2018. 2

[22] US Patent US9094771B2. Method and system for upmixing audio to generate 3d audio. 2004. 2

[23] Wikipedia contributors. Dolby atmos — Wikipedia, the free encyclopedia, 2021. [Online; accessed 16-March-2021]. 1

[24] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels, 2018. 2, 5

[25] A. Zunino, M. Crocco, S. Martelli, A. Trucco, A. Del Bue, and V. Murino. Seeing the sound: A new multimodal imaging device for computer vision. In *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 693–701, 2015. 2