ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10446118

SPATIAL SCAPER: A LIBRARY TO SIMULATE AND AUGMENT SOUNDSCAPES FOR SOUND EVENT LOCALIZATION AND DETECTION IN REALISTIC ROOMS

Iran R. Roman[1*] *Christopher Ick*[1*] *Sivan Ding*[1] *Adrian S. Roman*[2]
Brian McFee[1] *Juan P. Bello*[1]

[1] Music and Audio Research Laboratory, New York University, New York, USA
[2] Viterbi School of Engineering, University of Southern California, California, USA
[*]Equal contribution

ABSTRACT

Sound event localization and detection (SELD) is an important task in machine listening. Major advancements rely on simulated data with sound events in specific rooms and strong spatio-temporal labels. SELD data is simulated by convolving spatialy-localized room impulse responses (RIRs) with sound waveforms to place sound events in a soundscape. However, RIRs require manual collection in specific rooms. We present `SpatialScaper`, a library for SELD data simulation and augmentation. Compared to existing tools, `SpatialScaper` emulates virtual rooms via parameters such as size and wall absorption. This allows for parameterized placement (including movement) of foreground and background sound sources. `SpatialScaper` also includes data augmentation pipelines that can be applied to existing SELD data. As a case study, we use `SpatialScaper` to add rooms to the DCASE SELD data. Training a model with our data led to progressive performance improves as a direct function of acoustic diversity. These results show that `SpatialScaper` is valuable to train robust SELD models.

Index Terms— data augmentation, data simulation, room simulations, microphone arrays, spatial audio

1. INTRODUCTION

Sound event localization and detection (SELD) consists of two subtasks: localizing sound sources and determining their category (i.e. *music* vs *dog bark*) [1]. SELD is relevant for assistive technologies that improve the lifestyle and safety of low vision and audition individuals [2].

Training models requires strongly-labeled data collected in rooms using microphone arrays [3]. Curating data is labor-intensive and thus only a handful datasets exist [4, 5]. An alternative is to simulate data using room impulse responses (RIRs) [6]. Since an RIR's location is known, convolving it with a sound waveform results in a soundscape with an event whose location, class, and start/end time are perfectly known. This method allows for data simulation at scale. However, this method still assumes RIRs collected in real rooms.

We present `SpatialScaper`, an open-source library for SELD data simulation and augmentation. Existing methods assume a RIR database collected in real rooms and using specific microphone hardware. `SpatialScaper`, in constrast, emulates virtual rooms of any size to synthesize RIRs using a microphone array of any shape. This dramatically increases the range of acoustic diversity that `SpatialScaper` can simulate. Moreover, `SpatialScaper` utilizes many databases of real RIRs, allowing for simulations in real and synthetic rooms. `SpatialScaper`'s API parametrizes a soundscape's most important variables, such as the room size, wall absorption, the audio files that function as background and foreground events, their location in the room, etc. These can be user-defined or drawn from a distribution. Furthermore, `SpatialScaper` can also apply effects to individual sound events for data augmentation. Moreover, it can augment existing SELD datasets using techniques known to improve SELD metrics [7]. We include a case study to showcase how using `SpatialScaper` leads to improved SELD model performance. Our contributions:

1. A library to simulate SELD data with strong labels by using both real and synthetic RIRs.[1]
2. A study that highlights how increased acoustic diversity in SELD training data improves model performance.

2. RELATED WORK

While SELD is traditionally done via traditional signal-processing [8], modern approaches use deep neural networks [1, 3]. Major highlights over the past five years include: 1) the introduction of SELDnet [3], a model that is updated every year to serve as a SELD baseline [3], 2) the development of the muli-ACCDOA representation to detect multiple and overlapping sound sources even of the same class [9], 3) data augmentation techniques that can be applied to existing SELD datasets [7], and 4) robust SELD systems [7, 10]. These have been made possible in part due to SELD challenges, such as Detection and Classification of Acoustic Sound Events

[1]https://github.com/iranroman/SpatialScaper

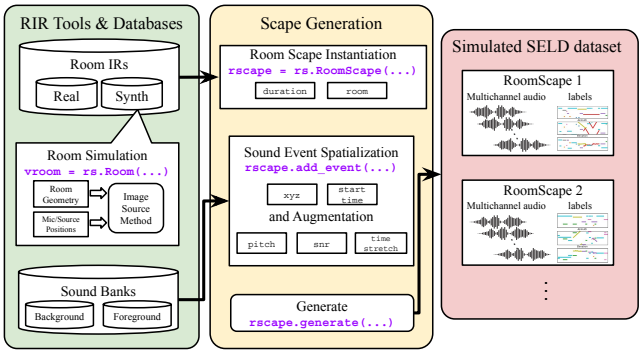

Fig. 1. `SpatialScaper` data generation pipeline.

(DCASE) [3] or LOCalization And TrAcking (LOCATA) [5], as well as the SELD datasets released for the 2019, 2020, and 2021 editions of DCASE consist of simulated data [6, 11, 12]. In 2022 and 2023, the DCASE challenge added a relatively small dataset of strongly-annotated recordings of sound events produced by human actors in the real world known as STARSS [4].

Several tools exist to simulate strongly-annotated soundscapes. One is `ambiscaper`, a library to simulate SELD data in the ambisonics spatial audio format [13]. Its limitations include the lack of dynamic sound sources (i.e. events that move), and no control of the distance between the microphone and the sound source, both of which are critical to develop "complete" SELD models [1, 14, 15]. It is also limited by the fact that it only simulates the first order ambisonics (FOA) format. Another code-base to simulate SELD data are scripts provided by the DCASE challenge organizers [16], which lack an API and thus offer limited possibilities to control simulation parameters. Furthermore, this codebase uses a fixed number of rooms in its RIR database with FOA and 4ch tetrahedral microphone (which we refer to as "MIC") formats. Other RIR databases that utililze other microphone formats are available [17, 18], but have not yet been used to generate SELD data. Furthermore, recent SELD literature shows promising results by utilizing geometric room simulations to create synthetic RIRs for generating SELD data [19, 20].

One of the most cited libraries for single-channel soundscape generation is `scaper` [21]. However it only generates sound event detection (SED) data, lacking localization cues. [21]'s importance is rooted in its full parameterization of effects (like pitch shifting or time stretching) and temporal placement of sound events. Parameters can be drawn from distributions to mitigate human bias data collection or simulation. We build upon `scaper` to add spatial control of rooms, delivering the first library for parameterized SELD data generation

3. SPATIAL SCAPER

`SpatialScaper`'s primary use-case is creating soundscapes using virtual or real RIRs. In the case of a virtual room, the room size, microphone array, and sound decay are controllable. Figure 1 gives an overview of the generation pipeline and 2 shows example code to simulate a soundscape. The next subsections explain the API functionalities.

```
1  import SpatialScaper as ss
2
3  # define a virtual room
4  vroom = ss.Room(
5     dims = [5,3,2], decay = 0.8, mic_type = 'em32',
6     mic_loc = [2.5,2.5,0.5]) # mic_type could be
7     # MIC, a list of capsule coordinates, etc.
8
9  # create a room scaper instance
10 ssc = ss.SScaper(duration = 60, room = vroom,
11    fg_path = '/path/to/fg_events',
12    bg_path = '/path/to/bg_events',
13    ref_db = -50, # in dB
14    )
15
16 # Add background noise
17 ssc.add_background(label = ('const', 'back'),
18    source_file = ('choose', []),
19    source_time = ('const', 0))
20    )
21
22 # Add a moving sound event
23 ssc.add_event(label = ('choose', []),
24    event_xyz = ('const',
25          [
26             [4.0,0.1,0.2], [4.5,0.1,1.9]
27          ] # inital and final position
28       )
29    )
30
31 # Genereate the audio and the annotation
32 ssc.generate(dest_path = '/path/out/rs1.wav')
```

Fig. 2. Instantiating a soundscape using a virtual room, microphone, background noise, and a moving foreground event.

3.1. Instantiating a room scape

`SpatialScaper` integrates databases of RIRs and/or a virtual RIR simulation engine that uses the image source method implementation of `pyroomacoustics` [22]. Line 4 in Figure 2 shows how a virtual room is defined. Required parameters include the room dimensions, decay factor for sounds reflecting off of walls, and a microphone in a location. Next, line 10 shows the instantiation of the soundscape. Parameters include the duration of the output file, the room definition, and the paths to possible foreground and background sound files. While the example in Figure 2 uses a virtual room, the user could also specify one of the rooms in our curated database of RIRs (`METU` [18], for example).

3.2. Adding background noise

Sustained background noise (i.e. AC or traffic) can be added to the room soundscape at a specific location with a constant SNR (usually a low value). Assuming that the `/path/to/bg_events` contains files with sustained long

noises, one or more can be selected and placed in the room soundscape (files will be looped throughout the track), using a `ref_db` level. Line 17 randomly selects a file from the path with background events and places it in a specific location. Alternatively, the background noise can be linearly added to all RIR channels to remove the localization effect.

3.3. Spatializing target events

Placing a target event involves selecting a file, determining the time-point where it will start playing (both within itself and in the room soundscape), assigning it an SNR level, and determining its initial and final location in the room. Effects such as random pitch-shift and time stretching may be applied. Line 23 in Figure 2 shows how a sound event is randomly selected to transverse the room between two xyz coordinates (specified by a list of two xyz coordinates). By default, the sound event will move linearly (i.e. along the shortest path) throughout its duration. Users may also specify other possible trajectories (spline, random walks, etc.).[2] When spatializing events in a real room, the trajectory is adjusted to the nearest one possible given the RIR layout recorded in the room.

3.4. Triggering the room scape generation

Finally, the soundscape generation. See line 32 in Figure 2 specifying the destination path. The conversion between microphone and ambisonics formats is facilitated by `SpatialScaper`'s built-in ambisonics encoder, adapted and improved from the related `micarraylib` library [23].

3.5. Augmenting existing SELD recordings

Given an existing SELD dataset, `SpatialScaper` can be used to augment using techniques like channel swapping, soundscape rotation, time-domain remixing, and random time-frequency masking [7]. `SpatialScaper` generalizes these augmentations for any spherical array. Figure 3 shows an example augmentation pipeline. It assumes that the directory with the SELD dataset has a metadata subdirectory with csv files consistent with the DCASE SELD challenge format. The API searches for all wav files with names that match the csv filenames to apply the augmentations.

```
ss.apply_augmentation(data_path='path/to/dataset',
                      aug_type = 'channel swapping')
# outputs path is 'path/to/dataset_swapped'

ss.apply_augmentation(data_path='/path/to/data',
                      aug_type = 'time freq mask')
# output path is 'path/to/dataset_tfmasked'
```

Fig. 3. Using `SpatialScaper` to augment a SELD dataset via the augmentations recently proposed by Wang et al. [7]

4. CASE STUDY: IMPROVING SELDNET

4.1. Model, training procedure, and metrics

SELDnet is the baseline model that the organizers of the DCASE SELD challenge update every year [3]. It is a convolutional recurrent neural network that can take either MIC or FOA inputs. Its output is the multi-ACCDOA representation that can detect and classify multiple and overlapping sound sources, even of the same class [9]. We use the SELDnet that was released with the 2023 version of STARSS [4].

To replicate SELDnet's training procedure, one must use data that is both simulated and recorded in the real world. The simulated data comes from the "DCASE 2022 SELD mixtures for baseline training" [24], which we refer to as "DCASE" dataset. The real data comes from the STARSS dataset [4]. SELDnet is trained with "DCASE" and the "dev-train" files in STARSS. In its official implementation the model is cross-validated on "dev-test" files in STARSS, and is "tested" on the same "dev-test" files. This is done because the annotations for the STARSS evaluation files are not publicly available. To ensure that we cross-validate and test on annotated recordings carried out in separate rooms, we divide the STARSS "dev-test" by recording location. We use the rooms that were recorded in Finland ("dev-test-tau") for cross-validation and the rooms that were recorded in Japan ("dev-test-sony") for the final test. This ensures that the sounds and rooms used for the final evaluation are not seen during model development.

We use the DCASE SELD metrics of location-dependent F-score (F) and error rate (ER) for classification, and localization error (LE) and recall (LR) [3]. SELDnet shows optimal performance on FOA, so our experiments use that format.

4.2. Exp 1: Adding acoustic diversity to the training data

The "DCASE" dataset consists of simulated SELD data using RIRs collected in nine different rooms (150 1min soundscapes per room). We use `SpatialScaper` to simulate more room soundscapes using real and synthetic RIRs, and we add these to SELDnet's training split for a total of up to 29 rooms (only 18 if not using `SpatialScaper`). This process adds acoustic diversity to SELDnet's training data. Consistent with the "DCASE" dataset, we generate one-minute-long soundscapes (150 per added room), and use the same sound categories (sourced from the FSD50K dataset [25]; our "music" tracks come from the FMA dataset [26]). We evaluate model performance on test (i.e. STARSS "dev-test-sony").

4.3. Exp 2: Replicating "DCASE" with augmentations

We also use `SpatialScaper` to recreate the "DCASE" dataset "from scratch" using real or synthetic RIRs. We refer

[2]Presently, free spatialization is only available in the virtual rooms.

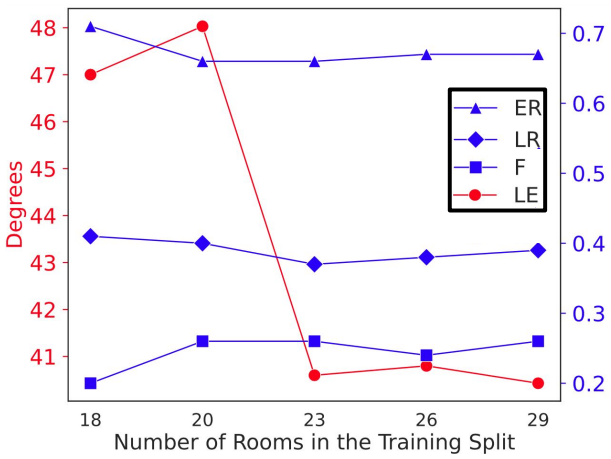

Fig. 4. Performance on the test split (STARSS23 "dev-test-sony") as a function of adding rooms (i.e. increasing acoustic diversity) to the training split.

to these datasets as "R real" and "R virt", respectively.

To showcase `SpatialScaper`'s ability to apply effects, we create a "R pitch" dataset where sound events are randomly pitch-shifted half an octave (up or down) before being spatialized. Furthermore, we apply the channel swapping augmentation to the "R real" dataset, resulting in the "R swap" dataset. We train separate SELDnet models using the generated data in either the "DCASE", "R real", "R pitch", "R swap", or "R virt" datasets. The use of the STARSS23 dataset splits remains consistent with Exp. 1.

5. RESULTS

Figure 4 shows the results of Exp. 1. Adding acoustic diversity to the training data improves SELDnet's LE. Other metrics such as ER, F, and LR are not significantly affected. This makes sense, since LE is determined by a model's ability to differentiate between a sound's direct path versus early wall reflections. Adding rooms increases the distribution of sound trajectories the model sees during training. Interestingly, the improved LE is observed after adding five rooms and seems to plateau. Also note that adding only two room impacted LE, which can be explained by the fact that all new rooms come from domains and collection procedures different than the original DCASE rooms. However, the benefit of adding rooms is unquestionable after more rooms are added.

Something important to highlight is that SELDnet's performance before adding rooms (i.e. with its original data and training procedure) is worse than it was reported when it was released [4]. This is explained by the fact that we divided the STARSS23 "dev-test" split into "dev-test-tau" for cross-validation and "dev-test-sony" for testing. Therefore, we cross-validated SELDnet with a smaller dataset, and evaluated it on what seems to be a challenging data split[3].

Data	ER	F	LE	LR
DCASE	0.71	20.2	47.0	40.5
R real	0.71	19.8	46.5	34.0
R swap	0.59	31.7	29.5	31.2
R pitch	0.67	17.8	48.4	36.9
R virt	0.75	7.4	96.7	19.1

Table 1. SELDnet performance on the STARSS23 "dev-test-sony" (our test split) for different versions of training data used.

Table 1 shows results for Exp. 2. Comparing the first ("DCASE") and second ("R real") rows, we observe that `SpatialScaper` reproduces the "DCASE" dataset. A noticeable difference is the worse LR score. This can be attributed to the fact that we reproduced RIR databases and sound sources used in the original "DCASE" data, but authors did not share the music files they used. This is why we used the FMA dataset, which introduced a shift in the data distribution and could be the cause for the differences that we observe. It is worth noting that these differences are only observed in the localization metrics (LE and LR), likely due to the fact that changing the music in a room can have drastic changes in the resulting soundscape reverberation patterns.

We also see that using `SpatialScaper` to apply the channel-swapping augmentation leads to improvements across all metrics (except LR), an effect already described [7]. Furthermore, we observe the pitch-shift effect seems to have minimal effects. Finally, training with the simulated "R virt" data, which uses virtual RIRs, greatly impacts all metrics. This is not surprising, as the real RIRs better capture the reverberation properties observed in the test set data, which was collected in real rooms with human actors. This effect has been recently reported in the literature [20].

6. CONCLUSIONS

The presented work underscores the significance of acoustic diversity in training SELD models. `SpatialScaper` is an open-source library for parametric simulation and augmentation of SELD data at scale. Its capability to emulating varied acoustic environments proves valuable for model robustness and experimentation. We encourage the broader community to explore, use, and contribute to `SpatialScaper`, fostering advancements in the SELD domain.

7. ACKNOWLEDGEMENTS

This work is supported by the National Science Foundation grant no. IIS-1955357. The authors thank the funding source and their grant collaborators.

[3] We could replicate the results in [4] by using all "dev-test" files for cross-validation and testing, as written in the code implementation by the authors.

8. REFERENCES

[1] Pierre-Amaury Grumiaux, Sran Kitić, Laurent Girin, and Alexandre Guérin, "A survey of sound source localization with deep learning methods," *JASA*, vol. 152, no. 1, pp. 107–151, 2022.

[2] Sharnil Pandya and Hemant Ghayvat, "Ambient acoustic event assistive framework for id, detection, and recognition of unknown acoustic events of a residence," *Advanced Engineering Informatics*, vol. 47, 2021.

[3] Archontis Politis, Annamaria Mesaros, Sharath Adavanne, Toni Heittola, and Tuomas Virtanen, "Overview and evaluation of seld in dcase 2019," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, 2020.

[4] Kazuki Shimada, Archontis Politis, et al., "Starss23: An audio-visual dataset of spatial recordings of real scenes with spatiotemporal annotations of sound events," *arXiv:2306.09126*, 2023.

[5] Heinrich W Löllmann, Christine Evers, et al., "The locata challenge data corpus for acoustic source localization and tracking," in *IEEE 10th SAM Workshop*, 2018.

[6] Sharath Adavanne, Archontis Politis, and Tuomas Virtanen, "A multi-room reverberant dataset for seld," *arXiv:1905.08546*, 2019.

[7] Qing Wang, Jun Du, et al., "A four-stage data aug. approach to resnet-conformer based acoustic modeling for seld," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1251–1264, 2023.

[8] Taras Butko, Fran González Pla, et al., "Two-source acoustic event detection and localization: Online implementation in a smart-room," in *2011 19th European Signal Processing Conference*, 2011, pp. 1317–1321.

[9] Kazuki Shimada, Yuichiro Koyama, et al., "Multi-accdoa: Localizing and detecting overlapping sounds from the same class with aux. duplicating permutation invariant training," in *ICASSP*, 2022.

[10] Jinbo Hu, Yin Cao, Ming Wu, Qiuqiang Kong, Feiran Yang, Mark D. Plumbley, and Jun Yang, "A track-wise ensemble event independent network for polyphonic seld," in *ICASSP*, 2022, pp. 9196–9200.

[11] Archontis Politis, Sharath Adavanne, and Tuomas Virtanen, "A dataset of reverberant spatial sound scenes with moving sources for seld," *arXiv:2006.01919*, 2020.

[12] Archontis Politis, Sharath Adavanne, Daniel Krause, Antoine Deleforge, Prerak Srivastava, and Tuomas Virtanen, "A dataset of dynamic sound scenes with directional interferers for seld," *arXiv:2106.06999*, 2021.

[13] Andres Perez-Lopez, "Ambiscaper: A tool for automatic generation and annotation of reverberant ambisonics sound scenes," in *2018 16th IWAENC*, 2018.

[14] Saksham S Kushwaha, Iran R Roman, Magdalena Fuentes, and Juan P Bello, "Sound source distance estimation in diverse and dynamic acoustic conditions.," in *WASPAA*, 2023.

[15] Benjamin S Liang, Andrew S Liang, Iran Roman, et al., "Reconstructing room scales with a single sound for augmented reality displays," *Journal of Information Display*, vol. 24, no. 1, pp. 1–12, 2023.

[16] Daniel Krause and Archontis Politis, "github.com/danielkrause/dcase2022-data-generator," .

[17] Thomas McKenzie, Leo McCormack, and Christoph Hold, "Dataset of Spatial Room Impulse Responses in a Variable Acoustics Room for Six Degrees-of-Freedom Rendering and Analysis," Nov. 2021.

[18] Orhun Olgun and Huseyin Hacihabiboglu, "METU SPARG Eigenmike em32 Acoustic Impulse Response Dataset v0.1.0," Apr. 2019.

[19] Prerak Srivastava, Antoine Deleforge, Archontis Politis, and Emmanuel Vincent, "How to (virtually) train your speaker localizer," August 2023.

[20] Christopher Ick and Brian McFee, "Leveraging geometrical acoustic simulations of spatial room impulse responses for improved seld," in *DCASE*, September 2023, pp. 56–60.

[21] Justin Salamon, Duncan MacConnell, Mark Cartwright, Peter Li, and Juan Pablo Bello, "Scaper: A library for soundscape synthesis and augmentation," in *WASPAA*, 2017, pp. 344–348.

[22] Robin Scheibler, Eric Bezzam, and Ivan Dokmanic, "Pyroomacoustics: A python package for audio room simulations and array processing algorithms," *CoRR*, vol. abs/1710.04196, 2017.

[23] Iran R Roman and Juan Pablo Bello, "Micarraylib: Software for the reproducible aggregation, standardization, and signal processing of microphone array datasets," in *DCASE*, 2021.

[24] Archontis Politis, "[DCASE2022 Task 3] Synthetic SELD mixtures for baseline training," Apr. 2022.

[25] Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and X Serra, "Fsd50k: an open dataset of human-labeled sound events," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, 2021.

[26] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson, "Fma: A dataset for music analysis," *arXiv preprint arXiv:1612.01840*, 2016.