SPRING

GRANT AGREEMENT N. 871245

Deliverable D2.3

Audio-Visual Data Simulator

Due Date: 31/01/2022
Main Author: INRIA
Contributors: BIU, CVUT
Dissemination: Public Deliverable / Confidential Deliverable

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

DOCUMENT FACTSHEET

Deliverable no.	D2.3: Audio-visual data simulator
Responsible Partner	INRIA
Work Package	WP2: Environment Mapping, Self-localisation and Simulation
Task	T2.2: Audio-Visual Data Simulator
Version & Date	VFinal, 31/01/2022
Dissemination level	[X] PU (public) [ ] CO (confidential)

CONTRIBUTORS AND HISTORY

Version	**Editor**	**Date**	**Change Log**
V1	INRIA, BIU	17/01/2022	First Draft
V2	CVUT	19/01/2022	Vision simulator update and review
V3	CVUT	27/01/2022	Illustrations and references added
VFinal	INRIA	31/01/2022	Adapted structure

APPROVALS

Authors/editors	INRIA, BIU, CVUT, ERM
Task Leader	LEADER
WP Leader	CVUT

TABLE OF CONTENTS

EXECUTIVE SUMMARY

EXECUTIVE SUMMARY

This deliverable aims to report the progress on the audio-visual data simulator. There are three main challenges when simulating AV data within the SPRING project:

1. To generate realistic data that can be used to train, e.g., the audio-visual tracker;
2. To generate data that corresponds to the AV sensing capabilities of ARI, especially regarding audio.
3. To be able to generate large amounts of data.

Since the microphone array of ARI is positioned and oriented so that the recorded audio does not correspond to any available large-scale datasets, we emphasize generating realistic audio data than generating video data.

We organized this into two different modules so that they could be independently replaced in the future or be adapted in other projects or to other robotic platforms.

First, we developed an AV data generator based on the images and annotations of a publicly available dataset. With clean speech signal input, it generates the binaural signals at a sensor's position/orientation.

Secondly, we developed an audio generator that, given a clean speech signal, it can provide sound signals that correspond to ARI's microphones.

Finally, we implemented simulations of visual input, 3D mapping using AI Habitat, and training data generation in MyGym.

The software is available at: https://gitlab.inria.fr/spring/wp2_mapping_localization/av-sim.

AV BINAURAL SIMULATOR

*Illustration SEQ Illustration * ARABIC 1: Stiched image of the JackRabbot dataset [1].*

AUDIO-VISUAL BINAURAL SIMULATOR

Given a video sequence and 3D annotations of the speakers, the audio-visual binaural simulator (AV simulator) provides simulated binaural audio sequences, which correspond to the trajectories of the speakers in space and time.

To give an example, our AV simulator simulates binaural audio sequences for the 360-degree stitched images (see Illustration 1) in the JackRabbot dataset [1] captured by a robot in an indoor scenario. For the visual part, the JackRabbot dataset provides 3D positions *(x, y, z)* and orientations of the people in the video. The projected 2D positions (i.e., bounding boxes with the upper left corner *(x, y)*, box width *w*, and height *h*) are also given. Importantly, it provides identity labels of the speakers, which allows us to provide a unique audio identity for each speaker.

For the audio part, since the JackRabbot dataset does not contain audio recordings, the AV simulator explores clean speeches from the TIMIT dataset [2]. The TIMIT dataset provides clean (i.e., no background noise) narrations from more than 400 narrators. Concretely, the AV simulator first generates mono-channel audio sequences by synchronizing the audio and video sequences using audio samples from the TIMIT dataset and 3D labels from the JackRabbot dataset. Then, the AV simulator explores the deep learning-based binaural audio generation networks from [3] to provide binaural audio outputs. The listener is considered in the center of the 3D space of the video sequence, and the outputs create a spatial and temporal correspondence between the locations (relative to the listener) of the (moving or immobile) speakers and the generated synchronized mono audio. The binaural audio generation networks are pre-trained by [3], and the weights are frozen during our binaural audio generation. It inputs the positions of speakers and their generated mono audio sequences from our AV simulator and outputs the corresponding binaural audio sequences.

Overall, the AV simulation can be summarized into two steps: i) AV simulator first collects 3D positions and orientations from a given dataset and generates synchronized mono audio (as described in Algorithm 1). ii) Given the speakers' positions, orientations, and synchronized audio as inputs, the AV simulator in the second step leverages the binaural audio neural generator from [3] to output the final binaural audio.

Algorithm 1 3D-Position and Mono-Audio Generation. The linear interpolation is necessary if the binaural audio generation requires different label sampling frequency than the video frame frequency.

```
Require: video_frames, audio_seqs, video_3Dlabels
  mono_audios ← [ ]
  labels_3d ← [ ]
  $candidate\_audio$ ← audio_seqs
  $num\_chunks \leftarrow audio\_freq/video\_freq$        ▷ define the number audio chunks per frame.

  for frame_idx in video_frames do
      if $length$(candidate_audio) $\leq num\_chunks$ then
          $candidate\_audio$ ← audio_seqs
      end if
      mono_audios ← mono_audios ∪ candidate_audio[: $num\_chunks$]    ▷ collect audio
      candidate_audio ← candidate_audio[$num\_chunks$ :]

      labels ← video_3Dlabels[$frame\_idx$]        ▷ pos. and ori. of multiple trajectories.
      $label$ ← labels                             ▷ collect pos. and ori.
      $label \leftarrow linear\ interpolation$
      labels_3d ← labels_3d ∪ $label$
  end for
```

BEYOND BINAURAL SIMULATION

BEYOND BINAURAL NEURAL SIMULATION

As an alternative to the DNN-based binaural rendering, we also provide a "physical" acoustic simulator. This simulator is based on the image method [4,5]. An efficient C++ implementation with a Matlab wrapper is available.[1]

When a sound source propagates in an echoic enclosure, it is reflected from the room facets (walls, floor, and ceiling) and objects within the environment. For each reflection, the sound is partially absorbed (the level of the energy loss depends on the material of the corresponding room facet).

The image method [4,5] is an efficient method for calculating room impulse responses. The reflections of the sound on the room facets are realised by multiple images of the source beyond the room facets (similarly to multiple reflecting mirrors).

The properties of the obtained room impulse response (RIR) are widely analysed in the literature. A detailed description is beyond the scope of this summary. The interested reader is referred to [6,7,8,9]. Typically, this RIR comprises the direct-arrival, sparse reflections corresponding to the early arrivals and a dense tail of reflections, with exponentially decaying amplitude, corresponding to the reverberation time of the room (referred to as late arrivals).

There are two main drawbacks of this approach. First, it fits only "shoebox" rooms with a rectangular shape. Moreover, no objects can be added to the room (namely, no furniture, etc.). The second problem is the microphone installation. The image method and the respective implementation assume that the microphones are mounted in free space, i.e., no reflections from the mounting device are considered. In our case, the microphones are mounted inside ARI (below the chest).

It is well known from the binaural rendering literature[2] [10] that the sound wave interacts with the so-called head-related transfer function (HRTF) of the hearing aid wearer. In our case, the device-related transfer function (DRTF) should be considered instead.

We have therefore decided to incorporate the DRTF into the image method. For that, we have calculated for each relevant reflection its angle-of-arrival and superimposed the corresponding DRTF picked from a pre-recorded database. To reduce computational complexity, we implemented the procedure up to a user-defined reflection order, typically set to 2, i.e., the direct-arrival plus six first-order reflections plus 36 second-order reflections.
Further reflections can be either implemented as in the regular image method (with amplitude matching procedure to the early reflections) or even more efficiently, as an exponentially decaying Gaussian process, according to the RIR statistical models. In the latter case, we should also consider the spatial coherence of the late reverberation tail [12].

The sources' positions will be set in accordance with the visual information. ARI will be positioned in the centre of the 3D space of the video sequence.

[1] https://github.com/ehabets/RIR-Generator

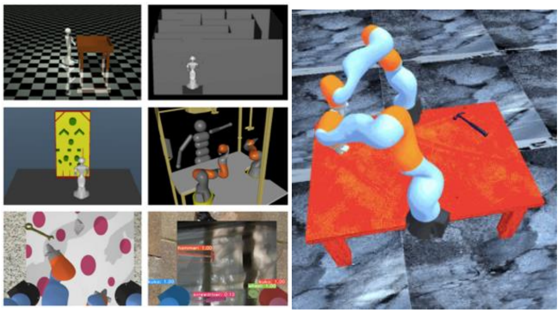
[2] https://www.ece.ucdavis.edu/cipic/spatial-sound/hrtf-data/

The simulator was initially tested with a newly published multi-microphone behind-the-ear HRTFs [11]. We are now designing a recording campaign to establish a database of ARI's DRTF. This will only be measured in multiple azimuth angles and a fixed elevation angle. The measurements should be carried out in an anechoic chamber. This is unavailable at BIU, so we will use BIU's var-echoic chamber at the lowest applicable reverberation level (about 100 ms).

The efficacy of the simulator will be verified in the task of generating large datasets of the speech signals (in multiple reverberation levels) that can be used to train the algorithms developed in the course of the SPRING project. It will also be compared with a DNN-based binaural rendering algorithm (note that ARI comprises four microphones). The algorithms will be evaluated with real data collected with ARI in the various labs and the Broca hospital.

SIMULATORS FOR 3D MAPPING, LOCALIZATION AND OBJECT DETECTION TRAINING

AI Habitat model of Broca Hospital



MyGym Simulation environment

We implemented a simulation of robot movement in a photorealistic environment in the AI Habitat platform [13], which includes Habitat Sim and Habitat Lab.

AI Habitat Sim is a high-performance 3D simulator with configurable agents, multiple sensors, and generic 3D data manipulation. It offers extremely fast rendering, achieving 10000 fps on a single GPU.

AI Habitat Lab is a modular high-level Python library for developing AI in such tasks as robot navigation, instruction following, and question answering. After training and evaluating the agent in the simulation, we can transfer the acquired knowledge and skills to the real robot [15].

3D models for the AI Habitat environment can be generated using 3D reconstruction and scanning. We use 3D Matterport [matterport.com/] technology, which provides good quality models at a reasonable cost.

To train state of the art object detection and segmentation pipeline YOLACT [github.com/dbolya/yolact] using data augmentation, we have implemented data generation in myGym [14] framework. It is a modular framework for developing and benchmarking RL algorithms. myGym contains a module allowing to work with tasks that incorporate visual recognition. It provides an option to generate synthetic datasets for integrated computer vision models such as YOLACT.

REFERENCES

REFERENCES

[1] R. Mart́ın-Mart́ın, M. Patel, H. Rezatofighi, A. Shenoi, J. Gwak, E. Frankel, A. Sadeghian, and S. Savarese. Jrdb: A dataset and benchmark of egocentric robot visual perception of humans in built environments. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.

[2] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett. Darpa TIMIT acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. NASA STI/Recon technical report n, 93:27403, 1993.

[3] A. Richard, D. Markovic, I. D. Gebru, S. Krenn, G. A. Butler, F. Torre, and Y. Sheikh. Neural synthesis of binaural speech from mono audio. In International Conference on Learning Representations, 2020.

[4] J. Allen and D. Berkley, "Image Method for Efficiently Simulating Small Room Acoustics," Journal of the Acoustical Society of America, vol. 65, no. 4, pp. 943-950, 1979.

[5] P. Peterson, "Simulating the response of multiple microphones to a single acoustic source in a reverberant room," Journal of the Acoustical Society of America, vol. 80, no. 5, pp. 1527-1529, Nov. 1986.

[6] R. Badeau, "Unified Stochastic Reverberation Modeling," in European Signal Processing Conf. (EUSIPCO), 2018.

[7] H. Kuttruff, "Room Acoustics," 4th ed. London: Spon Press, 2000.

[8] Jean-Dominique Polack, "Playing billiards in the concert hall: The mathematical foundations of geometrical room acoustics, Applied Acoustics, vol. 38, Issues 2–4, 1993, pp. 235-244,

[9] M.R. Schroeder, "New method of measuring reverberation time," The Journal of the Acoustical Society of America vol. 37, no. 6, 1965, pp. 1187-1188.

[10] V.R. Algazi, R.O. Duda, D.M. Thompson and C. Avendano, "The CIPIC HRTF Database," Proc. 2001 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), pp. 99-102, Mohonk Mountain House, New Paltz, NY, Oct. 21-24, 2001.

[11] H. Kayser, S.D. Ewert, J. Anemüller, T. Rohdenburg, T. Hohmann, and B. Kollmeier, "Database of multichannel in-ear and behind-the-ear head-related and binaural room impulse responses," EURASIP Journal on Advances in Signal Processing, 2009, pp.1-10.

[12] E.A.P. Habets, Sharon Gannot, "Generating sensor signals in isotropic noise fields," J. Acoustical Society of America, Vol. 122 No. 6, pp. 3464-3470 December 2007.

[13] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, D. Batra. Habitat: A Platform for Embodied AI Research. IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[14] M. Vavrecka, N. Sokovnin, M. Mejdrechova, G. Sejnova, M. Otahal. MyGym: Modular toolkit for visuomotor robotic tasks, 2020.

[15] N. Sokovnin, T Pajdla supervisor). Recognizing Unknown Objects for Open-Set 3D Object Detection. MSc thesis. CTU in Prague. 2021.