Blind Acoustic Parameter Estimation and Spatial Audio Rendering for Augmented Reality

Supporting Repository Documentation

Abstract

This implementation develops a complete end-to-end system for blind acoustic parameter estimation and spatial audio rendering in augmented reality (AR). A Convolutional Recurrent Neural Network (CRNN) is trained on synthetically generated reverberant speech to predict acoustic parameters — RT60, Direct-to-Reverberant Ratio (DRR), and Speech Clarity Index (C50) — from mono audio.

The estimated parameters are then used to control a Feedback Delay Network (FDN)-based rendering module, producing binaural audio consistent with the predicted room acoustics.

Evaluation in Phase I assessed the accuracy of parameter estimation on synthetic and real-world speech, while Phase II focused on perceptual quality and intelligibility of the rendered audio, using both objective and subjective measures.

Repository Structure

```
Blind_Acoustic_Parameter_Estimation_And_Spatial_Audio_Rendering_
For_Augmented_Reality/
|
|-- data/                          # PDF link to real dataset
|-- dry_speech/                    # Clean speech recordings
|-- features/                      # Extracted features
|-- labels/                        # Training targets (RT60, DRR, C50)
|-- models/                        # Model definitions and pre-trained weights
|   `-- crnn_dropout_bigru_v1.pt # Main trained CRNN model
|-- notebooks/                     # Jupyter notebooks for training, inference,
   rendering, evaluation
|   |-- Phase I/                  # Acoustic parameter estimation
|   `-- Phase II/                 # Rendering and evaluation
|-- Evaluation/                    # Scripts and results
|-- Output/                        # Generated outputs, logs, figures
|-- Additional Requirements/      # Supplementary files
```

Usage

The experiments are provided as Jupyter notebooks. A few external Python packages need to be installed before running them. Core scientific libraries (e.g., `numpy`, `scipy`, `matplotlib`, `pandas`, `scikit-learn`, `tqdm`) are typically included in most Jupyter environments, but the following packages should be installed manually via `pip`:

- `torch`, `torchaudio`

- `librosa`, `soundfile`
- `pesq`, `pystoi`
- `pyroomacoustics`

Run notebooks:

- **Phase I:** Blind acoustic parameter estimation (CRNN training/testing).
- **Phase II:** Spatial audio rendering and evaluation.

Update local file paths inside notebooks where necessary.

Notebooks Overview

The repository includes Jupyter notebooks organized into two phases:

Phase I: Acoustic Parameter Estimation

- **Phase_I_Model_Training.ipynb**
  Trains the Convolutional Recurrent Neural Network (CRNN) using synthetic reverberant speech to predict RT60, DRR, and C50.
- **Phase_I_Evaluation.ipynb**
  Evaluates the trained CRNN model on held-out synthetic data using MAE, RMSE, Pearson's r, and R^2.
- **Phase_I_Model_Inference_Real_World_Dataset.ipynb**
  Applies the trained CRNN to real-world recordings, producing estimated acoustic parameters for rendering.

Phase II: Spatial Audio Rendering and Evaluation

- **Phase_II_Pipeline_Exponential_Rendering.ipynb**
  Prototype pipeline for exponential-based rendering methods. - NOT USED
- **Phase_II_Rendering_FDN.ipynb**
  Main notebook for rendering binaural audio with the Feedback Delay Network (FDN), using CRNN-estimated parameters as input.
- **Phase_II_Evaluation_Acoustic_Consistency.ipynb**
  Compares estimated parameters against those embedded in rendered signals, ensuring acoustic consistency.
- **Phase_II_Evaluation_PESQ.ipynb**
  Computes the Perceptual Evaluation of Speech Quality (PESQ) scores on rendered signals.
- **Phase_II_Evaluation_STOI.ipynb**
  Computes the Short-Time Objective Intelligibility (STOI) scores on rendered signals.

- **Phase_II_FDN_RT60_tail_Evaluation_Plots.ipynb**
  Produces diagnostic plots for RT60 decay tails in the FDN-rendered signals, providing insight into reverberation behavior.

Data

- **Synthetic dataset (Phase I):** Included. Generated from LibriSpeech clean speech convolved with simulated Room Impulse Responses (RIRs).
- **Real-world dataset (Phase I):** Not included due to size. A PDF in `data/` contains the download URL.

Models

- Model definitions are included in `models/`.
- Pre-trained CRNN weights:
  - `crnn_dropout_bigru_v1.pt` — checkpoint used for inference and rendering.

Evaluation

Phase I: Acoustic Parameter Estimation

- **Metrics:** MAE, RMSE, Pearson's r, R^2.
- **Scope:** Accuracy of CRNN predictions on synthetic test data and real-world inference.

Phase II: Spatial Audio Rendering

- **Metrics:** PESQ, STOI, and acoustic consistency checks (e.g., RT60 decay analysis).
- **Scope:** Quality and intelligibility of FDN-rendered binaural audio compared to target acoustics.