ACOUSTIC CHARACTERIZATION OF ENVIRONMENTS (ACE) CORPUS SOFTWARE INSTRUCTIONS

J. Eaton and P. A. Naylor

Department of Electrical and Electronic Engineering, Imperial College London, UK

1. INTRODUCTION

This document describes how to use the Acoustic Characterization of Environments (ACE) corpus database and software. The ACE corpus [1] was devised to evaluate state-of-the-art algorithms for blind acoustic parameter estimation from speech and to promote the emerging area of research in this field. Participants evaluated their algorithms for Reverberation Time (T_{60}) and Direct-to-Reverberant Ratio (DRR) estimation against the 'ground truth' values provided with the data-sets and presented the results in a paper describing the method used. The proceedings of the challenge is available [2]. In the ACE Challenge, the Development (Dev) used codenames for the rooms, and the Evaluation (Eval) database was obscured using a random permutation of IDs for each dataset. These were decoded on submission of results to the organizers. The ACE corpus is fully open. It comprises real recorded speech, Acoustic Impulse Response (AIR) and noise.

1.1. Contents of the ACE Corpus

The corpus comprises the following components:

1. Corpus components (anechoic speech utterances, AIRs, noise)
2. Dev datasets (6 microphone configurations $\times$ 288 files)
3. Eval datasets (6 microphone configurations $\times$ 4500 files)
4. Matlab software (Create datasets, test algorithms, analyse results) and example files for verification
5. Documentation

2. CORPUS DESCRIPTION

Details of the corpus are included in [1]. Please refer to this paper for more detailed information regarding the corpus, and we request that you cite this in your work.

3. CORPUS SOFTWARE

A suite of functions is provided to prepare a test database, test an algorithm, and analyse the results of the test. The software is contained in a compressed archive. Top level functions described below call functions in the *private/* folder which do not need to be modified to verify that the software works and test an algorithm. Test results files are provided to verify that the software is correctly installed.

3.1. Preparing a database

Dev and Eval databases have been generated as part of the corpus. However, the software used to generate these databases is also provided, along with the speech, noise and AIR to allow researchers to generate their own configurations, or generate the databases to avoid downloading the full package. The following functions are provided:

- *test_gen_ACE_corpus_dataset.m* Script to generate datasets. The script can be modified to generate customized datasets at different sample rates and including different components, however, without configuration it will generate the Dev datasets based on the same talkers, rooms and noises as for the ACE Challenge, and with the modification of the DATASET variable on line 160 it will produce a new version of the Eval dataset.
  - Because the noise used to generate a dataset is randomly selected from the available noise files which are some minutes in length, the datasets generated will not be identical to those used in the ACE Challenge.
  - Generating all the datasets will take a long time. The Dev set took around 4 hours to generate on our server, whilst the Eval set took around 3 days.

- Files for each dataset are written into sub-folders according to the microphone configuration. The ground truth values for each dataset for each microphone configuration are stored in each sub-folder with the name:
  <yyyymmdd>T<hhmmss>_test_gen_corpus_dataset_results.csv
  e.g. 20150723T134416_test_gen_corpus_dataset_results.csv in comma separated values (.csv) format. The .csv format was chosen for all data files in the software because it is platform independent, can be read without special tools, when opened in spreadsheet applications allows analysis, and when opened in the Finder in OS X is formatted in columns. Another benefit of using .csv files is that jobs can be interrupted and restarted from the point of interruption and no data will be lost.
- The naming convention for each .wav file is as follows:
  <Mic. config.>_<Room>_<Room config.>_<Talker>_<Utterance>_<Noise>_<Signal-to-Noise Ratio (SNR) in dB>dB.wav
  e.g. EM32_Meeting_Room_1_1_M1_s1_Ambient_-1dB.wav

3.2. Testing an algorithm

The following functions are provided:

- *test_estimator.m*. This function runs through all the noisy reverberant speech files in a dataset for a given microphone configuration and tests an algorithm using that file. This should be modified as follows:
  - Set the sample rate F_S to the sample rate required by your algorithm. Dev and Eval are stored 16kHz 16-bit depth. The components of the corpus were recorded at 48kHz 32-bit depth. If your sample rate is different to f_s =16 kHz used by the ACE challenge, the noisy reverberant speech will be resampled to the rate specified in the F_S variable.
  - Set the GT_BASIS. With GT_BASIS = 0, the ground truth values loaded during testing of an algorithm will be the mean across all channels for the microphone configuration. With GT_BASIS = n, the ground truth values for channel n will be added to the results file.
  - Set CORPUS_FOLDER_DEV and CORPUS_FOLDER_EVAL to point to where these datasets have been installed. When execution is restarted, new results will be appended to this file without generating a new header.
  - MIC_CONFIGS_DEV and MIC_CONFIGS_EVAL contain the names of the ground truth data files for each microphone configuration (Single, Crucif, Mobile, etc.) for the Dev and Eval datasets. These are provided within the datasets. These names should be modified if the datasets are regenerated.
- *callEstimator.m* calls any included T_{60} and/or DRR estimation function. The noisy reverberant speech and sample rate are available directly as parameters. Random values are included in order to verify that the software is correctly installed. These should be removed and replaced with the values returned from the estimation function. It should be modified as follows:
  - Add your algorithm within each of the comments
  - If your algorithm estimates the SNR and/or Direction-of-Arrival (DoA), fields are provided within the results file format for this to be analysed later.
  - The *results* variable is passed into the *callEstimator.m* function already initialised with zeros and place-holder values. Update the fields in this structure as appropriate to your algorithm. If your algorithm estimates less than the number of subband positions available, do not truncate the subband vector as this may affect subsequent reading of the results file. The author's T_{60} estimation algorithm [3] is included. This may be helpful as a baseline in testing.

  Measurement of real-time factor is performed around the call to *callEstimator.m* within this function and is reported in the results.

3.3. Analysing the results of a test

The following function is provided:

- *analyse_results.m*.
  This function reads in results produced by *test_estimator.m* and produces box plots at each of the SNRs and T_{60}s used in the test, and saves them as .png files into the current directory. This file is pre-configured with results generated using the author's T_{60} [3] estimator in the call estimator function. It should be modified as follows:
  - To analyse results from the *test_estimator.m* function, enter the filename without the .csv extension in the RESULTS_FILE variable.
  - Set the ANALYSIS_TYPE to 1 for percentage error (e.g. when analysing T_{60}) or 0 for absolute error (e.g. when analysing DRR).
  - Set the TASK_TYPE, 0 for T_{60} and 1 for DRR Set the PLOT_TYPE for the *print* function. This determines the format of the saved plot files.

  Running the function generates box plots of the results by SNR and by T_{60} and saves them in the current directory.

4. REFERENCES

[1] J. Eaton, N. D. Gaubitch, A. H. Moore, and P. A. Naylor, "The ACE challenge - corpus description and performance evaluation," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, 2015.

[2] ——, "ACE Challenge Workshop, a satellite event of IEEE-WASPAA," 2015. [Online]. Available: http://arxiv.org/abs/9999.9999v2

[3] J. Eaton, N. D. Gaubitch, and P. A. Naylor, "Noise-robust reverberation time estimation using spectral decay distributions with reduced computational cost," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013, pp. 161–165.