Real-Time Speech Recognition in a Multi-talker Reverberated Acoustic Scenario

Rudy Rotili[1], Emanuele Principi[1], Stefano Squartini[1], and Björn Schuller[2,⋆]

[1] A3LAB, Department of Biomedics, Electronics and Telecommunications,
Università Politecnica delle Marche, Via Brecce Bianche 1, 60131 Ancona, Italy
{r.rotili,e.principi,s.squartini}@univpm.it
http://www.a3lab.dibet.univpm.it
[2] Institute for Human-Machine Communication
Technische Universität München
Arcisstr. 21, 80333, Munich, Germany
Schuller@tum.de

1 Introduction

Speech-based Human-Machine interfaces have been gaining an increasing interest among the related scientific community and technology market. One of the key tasks to be faced within these architectures is Automatic Speech Recognition (ASR), for which a certain degree of understanding has been already reached in the literature. Several efforts have been oriented on purpose to take the presence of acoustic distortions into account and reduce the mismatch between training and testing conditions leading to poor recognition performances [1]. Multi-party meetings surely represent an interesting real-life acoustic scenario for this kind of application, where multiple speakers are simultaneously active in a reverberated enclosure: The presence of overlapping speech sources and of the reverberation effect due to convolution with room impulse responses (IRs) strongly degrades

[⋆] At the time of writing, Björn Schuller was a visiting lecturer at Università Politecnica delle Marche in Ancona, Italy and acknowledges funding by the European Union's ERASMUS lecturer mobility programme.

the ASR accuracy and a strong Digital Signal Processing (DSP) intervention is required in order to make the ASR system to work properly. Moreover, another important issue is represented by the real-time constraints: The recognition of the speech content related to each speaker is often required while the audio stream is processed, making the complete task even more challenging.

Several solutions have been proposed in the literature on Multiple-Input Multiple-Output (MIMO) to jointly address the blind separation and dereverberation problems: For example a two stage approach leading to sequential source separation and speech dereverberation based on blind channel identification (BCI) has been proposed in [2]. A real-time implementation of this approach has also been presented by some of the authors in [3]. The present contribution wants to employ such an algorithmic framework as multi-channel DSP front-end for the subsequent ASR system aimed at accomplishing the recognition task for all available speech streaming, always at run-time. It must be remarked that the employed solution also involves a speaker diarization module, able to identify the occurrence of an audio segment characterized by overlapping speakers and therefore correctly pilot the front-end and the ASR operations.

The overall framework has been developed on a freeware software platform, namely NU-Tech [4], suitable for real-time audio processing. The HTK engine [5] has been employed for real-time speech recognition. Experiments performed over an LVCSR corpus (synthetically manipulated to match the addressed acoustic scenarios) confirm the effectiveness and real-time capabilities of the aforementioned architecture implemented on a common PC.

The paper outline is the following. In Section 2 the overall multi-channel DSP framework, aimed at separating and dereverberating the speech sources is described. Section 3 is devoted to analyze the main parametrization and implementation issues relative to such a framework and the selected ASR engine, whereas in Section 4 the experimental setup is discussed and computer simulation results are commented. Conclusions are drawn in Section 5.

2 The Multi-channel DSP Front-End

Assuming M independent speech sources and N microphones with $M < N$; the relationship between them is described by an $M \times N$ MIMO FIR (Finite Impulse Response) system. According to such a model and denoting $(\cdot)^T$ as the transpose operator, the following equation for the n-th microphone signal holds:

$$x_n(k) = \sum_{m=1}^{M} \mathbf{h}_{nm}^T \mathbf{s}_m(k, L_h) + b_n(k), \qquad k = 1, 2, ..., K, \qquad n = 1, 2, ..., N \quad (1)$$

where $\mathbf{h}_{nm} = [h_{nm,0} \; h_{nm,1} \; \ldots \; h_{nm,L_h-1}]^T$ is the L_h-taps IR between the m-th source and the n-th microphone ($m = 1, 2, ..., M$, $n = 1, 2, ..., N$) and $\mathbf{s}_m(k, L_h) = [s_m(k) \; s_m(k-1) \; \ldots \; s_m(k - L_h + 1)]^T$. The signal $b_n(k)$ is a zero-mean gaussian noise with variance $\sigma_b^2, \forall n$. By applying the z-transform, the MIMO system can be expressed as:

$$X_n(z) = \sum_{m=1}^{M} H_{nm}(z)S_m(z) + B_n(z), \qquad n = 1, 2, ..., N. \tag{2}$$

The objective is recovering the original clean speech sources by means of a proper source separation and speech dereverberation algorithms considering in addition the presence of overlapping speakers. The framework proposed in [2,3] consists of three main stages: source separation, speech dereverberation and BCI. Firstly source separation is accomplished by transforming the original MIMO system in a certain number of Single-Input Multiple-Output (SIMO) systems and secondly the separated sources (but still reverberated) pass through the dereverberation process yielding the final cleaned-up speech signals. In order to make the two procedures properly working, it is necessary to estimate the MIMO IRs of the audio channels between the speech sources and the microphones by the usage of the BCI stage. As mentioned in the introductory section, this approach suffers of the BCI stage inability of estimating the IRs when two or more sources are concurrently active. To overcome this disadvantage a speaker diarization system can be introduced to steer the BCI stage. Speaker diarization takes as input the microphone mixtures and for each frame, the output $\mathcal{P}_i$ is 1 if the i-th source is the only active, and 0 otherwise. In such a way, the front-end is able to detect when to perform or not to perform the required operation. Using the information carried out by the speaker diarization stage, the BCI will estimate the IRs and the ASR engine will perform recognition if the corresponding source is the only active. In this work the oracle style is assumed, i.e. the speaker diarization system is assumed to operate at 100% of its possibilities. The block diagram of the proposed framework is shown in Fig. 1 where $N = 3$ and $M = 2$ have been considered. The three aforementioned algorithmic stages are now briefly described.

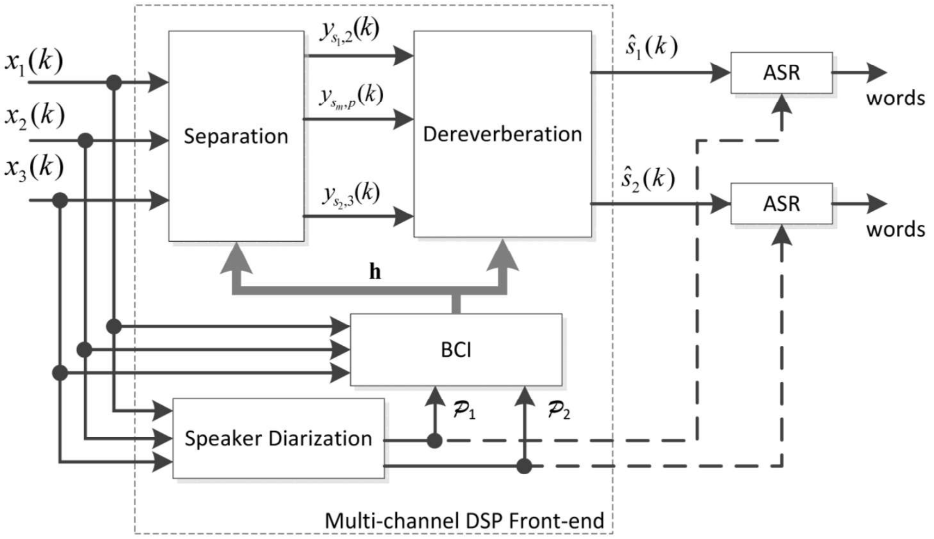

Fig. 1. Block diagram of the proposed framework

Blind Channel Identification Stage. MIMO blind system identification is typically obtained by decomposing the MIMO system in a certain number of SIMO subsystems in order to make the problem tractable and use powerful algorithms to properly estimate involved IRs. Considering a real-time scenario adaptive filter techniques are the most suitable. In particular the so-called unconstrained normalized multi-channel frequency-domain LMS [6] algorithm employed here represents an appropriate choice in terms of estimation quality and computational cost.

Source Separation Stage. Here we briefly review the procedure already described in [2] according to which it is possible to transform an $M \times N$ MIMO system (with $M < N$) in M $1 \times N$ SIMO systems free of interferences, as described by the following relation:

$$Y_{s_m,p}(z) = F_{s_m,p}(z)S_m(z) + B_{s_m,p}(z), \quad m = 1, 2, \ldots, M, \quad p = 1, 2, \ldots, P \quad (3)$$

where $P = C_N^M$ is the number of combinations. It must be noted that the SIMO systems outputs are reverberated, likely more than the microphone signals due to the long IR of equivalent channels $F_{s_m,p}(z)$. Related formula and the detailed description of the algorithm can be found in [2].

Speech Dereverberation Stage. Given the SIMO system corresponding to source s_m, let us consider the polynomials $G_{s_m,p}(z), p = 1, 2, \ldots, P$ as the dereverberation filters to be applied to the SIMO outputs to provide the final estimation of the clean speech source s_m, according to the following:

$$\hat{S}_m(z) = \sum_{p=1}^{P} G_{s_m,p}(z) Y'_{s_m,p}(z). \quad (4)$$

Optimal filtering is employed in [2], whereas adaptive solutions, like the one presented in [7], can be efficiently adopted to satisfy the real-time constraints. This has been done already in [3] and the same approach is followed here too.

3 Real-Time System Implementation

3.1 The Multi-channel DSP Front-End

This section is devoted to show how the multi-channel algorithmic framework depicted in Fig. 1 has been implemented for real-time ability within the NU-Tech framework [4]. NU-Tech allows the developer to focus on the algorithm implementation without worrying about the interface with the sound card. The ASIO protocol is supported to guarantee low latency times. NU-Tech architecture is plug-in based: An algorithm can be implemented in C++ language to create a NUTS (NU-Tech Satellite) that can be plugged in the graphical user interface. Inputs and outputs can be defined and connected to the sound card

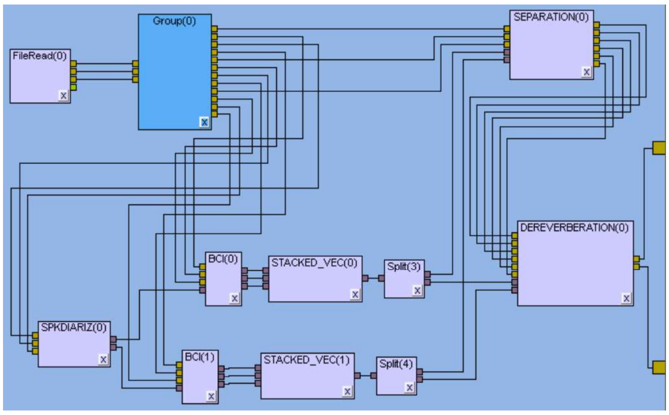

Fig. 2. NU-Tech setup

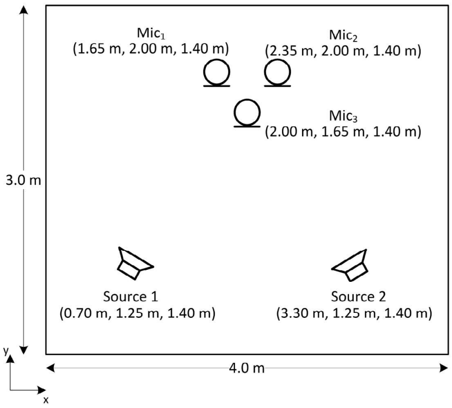

Fig. 3. Room setup

inputs/outputs or other NUTSs. The implementation of the algorithmic architecture reflects the block scheme of Fig. 1: Three NUTSs correspond to the actually developed stages (i.e. `BCI`, `DEREVERBERATION`, `SEPARATION`) and one is devoted to the speaker diarization oracle (`DIARIZATION`). In order to achieve a more optimized and efficient execution, all the NUTSs have been implemented by using the Intel® IPP library. Fig. 2 show the NU-Tech interface, with the previously described plug-ins and their interconnections. The real-time capabilities of the complete framework have been verified processing a 40 s long audio file. The averaged real-time factor obtained in ten algorithm runs is 0.14. Experiments have been conducted on a Intel Core i7 870 with 4 GB of RAM.

3.2 Automatic Speech Recognition Engine

Automatic speech recognition has been performed by means of the Hidden Markov Model Toolkit (HTK) [5] using HDecode, which has been specifically designed for large vocabulary speech recognition tasks. Features have been extracted through the HCopy tool, and are composed of 13 Mel-frequency Cepstral Coefficients, deltas and double deltas, resulting in a 39 dimensional feature vector. Cepstral mean normalization is included in the feature extraction pipeline. Recognition has been performed based on the acoustic models available in [8]. The models differ with respect to the amount of training data, the use of word-internal or cross-word triphones, the number of tied states, the number of Gaussians per state, and the initialization strategy. The main focus of this work is to achieve real-time execution of the complete framework, thus an acoustic model able to obtain adequate accuracies and real-time ability was required. The computational cost strongly depends on the number of Gaussians per state, and in [8]

Table 1. Characteristics of the selected acoustic model

Training data	WSJ0 & WSJ1	# of tied states (approx.)	8000
Initialization strategy	TIMIT bootstrap	# of Gaussians per state	16
Triphone model	cross-word	# of silence Gaussians	32

it has been shown that real-time execution can be obtained using 16 Gaussians per state. The main parameters of the selected acoustic model are summarized in Table 1. The language model consists of the 5k words bi-gram model included in the Wall Street Journal (WSJ) corpus. Recognizer parameters are the same as in [8]: using such values, the word accuracy obtained on the November ’92 test set is 94.30% with a real-time factor of 0.33 on the same hardware platform mentioned above. It is worth pointing out that the ASR engine and the multi-channel DSP front-end can jointly operate in real-time.

4 Computer Simulations

The acoustic scenario under study is made of an array of three microphones and two speech sources located in a small office. The room arrangement is depicted in Fig. 3. The data set used for the speech recognition experiments has been constructed from the WSJ November ’92 speech recognition evaluation set. It consists of 330 sentences (about 40 minutes of speech), uttered by eight different speakers, both male and female. The data set is recorded at 16 kHz and does not contain any additive noise or reverberation.

A suitable database representing the described scenario has been artificially created using the following procedure: The 330 clean sentences are firstly reduced to 320 in order to have the same number of sentences for each speaker. These are then convolved with IRs generated using the RIR Generator tool [9]. No background noise has been added. Two different reverberation conditions have been taken into account: the low and the and high reverberant ones, corresponding to $T_{60} = 120$ ms and $T_{60} = 240$ ms respectively (with IRs 1024 taps long).

For each channel, the final overlapped and reverberated sentences have been obtained by coupling the sentences of two speakers. Following the WSJ November ’92 notation, speaker 440 has been paired with 441, 442 with 443, etc. This choice makes possible to cover all the combinations of male and female speakers, resulting in 40 sentences per couple of speakers. The mean value of overlap has been fixed to 15% of the speech frames for the overall dataset. For each sentence the amount of overlap is obtained as a random value drown from the uniform distribution on the interval [12, 18]. This assumption allows the artificial database to reflect the frequency of overlapped speech in real-life scenarios such as two-party telephone conversation or meeting [10].

4.1 Experimental Results

The focus is on the recognition capabilities of the ASR engine fed by speech signals coming from the multichannel DSP front-end and therefore the quality index employed to evaluate the effectiveness of the approach is the word recognition accuracy. Other indexes, suitable to assess the performances of the sole multi-channel DSP front-end, have been already considered in [3]. Fig. 4 reports the word accuracy values attained in low and high reverberant conditions. The word accuracy obtained assuming ideal source separation and dereverberation is

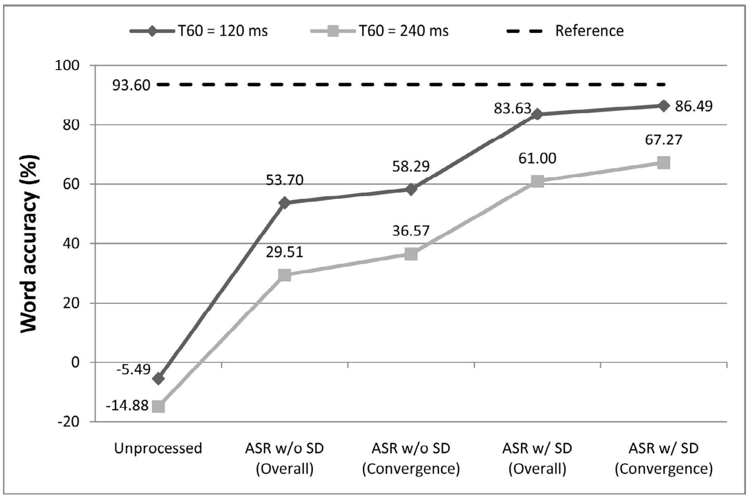

Fig. 4. Word accuracy (in percentage)

93.60%. This situation will be denoted as "Reference" in the remainder of the section. Three different setups have been addressed: In the first (*Unprocessed*), the recognition is performed on the reverberant speech mixture acquired from Mic$_3$ (see Fig. 3); in the second (*ASR w/o SD*), the ASR engine does not exploit the speaker diarization output; in the last one (*ASR w/ SD*) the ASR engine exploits the speaker diarization output.

The word accuracy has been evaluated in two different condition: *Overall*, where the complete test file is processed by the multi-channel DSP front-end and recognition is performed on the separated and dereverberated streams and *Convergence*, where the recognition is performed starting from the first silence frame after the BCI and dereverberation stages converge. Additional experiments have demonstrated that this is reached after $20 - 25$ s of speech activity. Observing the reported results, it can be immediately stated that feeding the ASR engine with unprocessed audio files leads to very poor performances. The missing source separation and the related wrong matching between the speaker and the corresponding word transcriptions result in a significant amount of insertions which justify the occurrence of negative word accuracy values.

Conversely, when the audio streams are processed, the ASRs are able to recognize most of the spoken words, specially once the front-end algorithms have reached the convergence. The usage of speaker diarization information to drive the ASR activity significantly increases the performance. In the *Convergence* evaluation case study, when $T_{60} = 120$ ms, a word accuracy of 86.49% is obtained, which is about 7% less than the result attainable in the "Reference" conditions. As expected, the reverberation effect has a negative impact on the recognition performances especially in presence of high reverberation, i.e. $T_{60} = 240$ ms. However, it must be observed that the convergence margin is even more significant w.r.t. the low-reverberant scenario, further highlighting the effectiveness of the proposed algorithmic framework as multichannel front-end.

5 Conclusions

In this paper, an ASR system was successfully enhanced by an advanced multichannel DSP front-end to recognize the speech content coming from multiple speakers in reverberated acoustic conditions. The overall architecture is able to blindly identify the impulse responses, to separate the existing multiple overlapping sources, to dereverberate them and to recognize the information contained within the original speeches. A speaker diarization oracle to pilot the BCI stage and the ASR engine has been also included in the overall framework. All the algorithms work in real-time and a PC-based implementation of them has been discussed in this contribution. Performed simulations, based on a existing large vocabulary database , have shown the effectiveness of the developed system, making it appeal in real-life human-machine interaction scenarios. As future works, a real speaker-diarization system will be integrated in the overall framework and its impact in terms of final recognition accuracy will be evaluated: the authors have already developed some interesting real-time solutions on purpose [11].

References

1. Peinado, A., Segura, J.: Speech Recognition Over Digital Channels. John Wiley & Sons, Ltd., Chichester (2006)
2. Huang, Y., Benesty, J., Chen, J.: A blind channel identification-based two-stage approach to separation and dereverberation of speech signals in a reverberant environment. IEEE Trans. Speech Audio Process. 13(5), 882–895 (2005)
3. Rotili, R., De Simone, C., Perelli, A., Cifani, S., Squartini, S.: Joint multichannel blind speech separation and dereverberation: A real-time algorithmic implementation. In: Huang, D.-S., McGinnity, M., Heutte, L., Zhang, X.-P. (eds.) ICIC 2010. CCIS, vol. 93, pp. 85–93. Springer, Heidelberg (2010)
4. Squartini, S., Ciavattini, E., Lattanzi, A., Zallocco, D., Bettarelli, F., Piazza, F.: NU-Tech: implementing DSP algorithms in a plug-in based software platform for real time audio applications. In: Proc. of 118th Convention of the AES (2005)
5. Young, S., Everman, G., Kershaw, D., Moore, G., Odell, J.: The HTK Book. Cambridge University Engineering (2006)
6. Huang, Y., Benesty, J.: A class of frequency-domain adaptive approaches to blind multichannel identification. IEEE Trans. Speech Audio Process. 51(1), 11–24 (2003)
7. Rotili, R., Cifani, S., Principi, E., Squartini, S., Piazza, F.: A robust iterative inverse filtering approach for speech dereverberation in presence of disturbances. In: Proc. of APCCAS 2008, pp. 434–437 (2008)
8. Vertanen, K.: Baseline WSJ acoustic models for HTK and Sphinx: Training recipes and recognition experiments. Cavendish Laboratory, University of Cambridge, Tech. Rep. (2006), http://www.keithv.com/software/htk/us/
9. Habets, E.A.P.: Room impulse response (RIR) generator (May 2008), http://home.tiscali.nl/ehabets/rirgenerator.html
10. Shriberg, E., Stolcke, A., Baron, D.: Observations on Overlap: Findings and Implications for Automatic Processing of Multi-Party Conversation. Word Journal of the International Linguistic Association, 1–4 (2000)
11. Colagiacomo, V., Principi, E., Cifani, S., Squartini, S.: Real-time speaker diarization on TI OMAP3530. In: Proc. of EDERC 2010 (2010)