N-BEST PARALLEL MAXIMUM LIKELIHOOD BEAMFORMERS FOR ROBUST SPEECH RECOGNITION

L. Brayda[1]*, C. Wellekens*

Institut Eurecom, 2229 Route des Cretes,
06904 Sophia Antipolis, France
{brayda,wellekens}@eurecom.fr

M. Omologo

ITC-irst, Via Sommarive 18,
38050 Povo (TN), Italy
omologo@itc.it

ABSTRACT

This work aims at improving speech recognition in noisy environments using a microphone array. The proposed approach is based on a preliminary generation of N-best hypotheses. The use of an adaptive maximum likelihood beamformer (the Limabeam algorithm), applied in parallel to each hypothesis, leads to an updated set of transcriptions, among which the maximally likely to clean speech models is selected. Results show that this method improves recognition accuracy over both Delay and Sum Beamforming and Unsupervised Limabeam especially at low SNRs. Results also show that it can recover the recognition errors made in the first recognition step.

1. INTRODUCTION

The use of microphone arrays in Automatic Speech Recognition (ASR) has shown significant improvements in the last years [1, 2]. The problem of coupling array processing with distant talking ASR has been recently addressed [3] due to the observation that improving speech intelligibility does not necessary increase recognition performance to the same extent. Advancements were obtained optimizing a Filter-and-Sum Beamformer through a Maximum Likelihood (ML) criterion: this setup showed improvements over both other adaptive filtering techniques [4] and the traditional Delay-and-Sum (D&S) beamformer [1]. Seltzer [5, 6, 3] proved that, by adapting a set of FIR filters on a set of clean speech models in an unsupervised manner, the input signal could be properly beamformed and the likelihood of test data increased. However, a careful analysis of this unsupervised maximum likelihood beamformer showed (see Section 5) that the improvements obtained over D&S were marginal with respect to the best results obtained with supervised methods. This suggested that when performing some kind of ASR feedback, one should rather consider the N-best hypotheses instead of the best one (in a maximum likelihood sense). In this work we propose and discuss an extension of the Limabeam algorithm. The proposed system applies N-best parallel beamformers to multi-channel signals; it independently optimizes and recognizes each hypothesized utterance; then it re-scores the likelihoods to provide a new maximally likely transcription. The paper is organized as follows: Section 2 justifies the use of this approach with a microphone array. Section 3 outlines the Limabeam algorithm, Section 4 describes our N-best approach to the Unsupervised Limabeam, and Section 5 gives the experimental results. Finally, Section 6 presents discussion and future activity.

[1]This work was conducted while L. Brayda was at ITC-irst.

2. N-BEST RECOGNITION WITH MICROPHONE ARRAYS

N-best recognition is known to be a useful approach when using progressive knowledge sources in multi-pass search strategies [7]. When performing ASR in noisy environments, it is well known that the correct transcription could be found in a very late position in an N-best list. Microphone array processing can be adopted to increase the SNR [8], to compensate for additive noise [4] and convolutional distortion or to maximize the likelihoods of the test set utterances with respect to given clean speech models [3]. Actually an array spans one dimension more (i.e. the spatial dimension) than the usual time-frequency domain spanned by its single-microphone counterpart. Thus, one can expect that the use of a microphone array can raise the position of the correct transcription in the N-best list. Let us assume to have a system that outputs the N-best hypotheses for known sentences and thus is able to select the correct one. Figure 1 depicts the performance of the system when one or eight microphones (in this case D&S is applied) are used in our experimental conditions (see Section 5). It is evident that the use of the array increases the amount of correct sentences among the N-best, i.e. the chances of picking up the correct transcription. This happens because all the candidates are more intelligible, less noisy, or better matching the models, depending on the array processing method adopted. At this point one would like the correct transcription be the first choice rather than the n-th. One way to do that is to re-process each hypothesis until it better matches the models to which it will be compared. We chose the Limabeam algorithm (see Section 3) to optimize and recognize in parallel each hypothesis (see Section 4), then re-score them and produce a final transcription.

3. THE LIMABEAM ALGORITHM

The Limabeam algorithm uses an adaptive filter-and-sum beamformer. Such a beamformer can be represented as follows:

$$x[k] = \sum_{m=1}^{M} h_m[k] * s_m[k] \tag{1}$$

where $s_m[k]$ is the discrete time domain signal received at the m-th microphone, $h_m[k]$ is the FIR filter for the m-th channel, $x[k]$ is the output of the beamformer, $*$ denotes convolution and k is the time index. The whole set of FIR coefficients of all the microphones can be represented by a super-vector $\mathbf{h}$. For each frame, recognition features can be derived and expressed in function of $\mathbf{h}$:

$$\mathbf{y}_L(\mathbf{h}) = \log_{10}\left(W\,|\mathrm{FFT}(\mathbf{x}(\mathbf{h}))|^2\right) \tag{2}$$

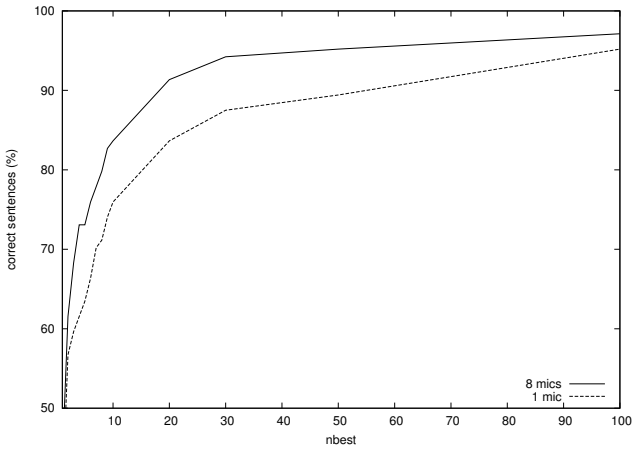

Figure 1: Percentage of correct sentences found by a system that recognizes the N-best hypotheses over transcribed sentences in a noisy environment. With more microphones the correct sentence is "pushed up" in the first alternatives. This result represents an upper-bound for system performance reported in the following.

where $\mathbf{x}(\mathbf{h})$ is the observed vector, $|\mathrm{FFT}(\mathbf{x}(\mathbf{h}))|^2$ is the vector of individual power spectrum components, W is the Mel filter matrix and $\mathbf{y}_L(\mathbf{h})$ is the vector of the Log Filter Bank Energies (LFBE). Cepstral coefficients are derived via a DCT rotation:

$$\mathbf{y}_C(\mathbf{h}) = \mathrm{DCT}(\mathbf{y}_L(\mathbf{h})). \tag{3}$$

Limabeam aims at deriving a set of M FIR filters, which maximize the likelihood of $\mathbf{y}_L(\mathbf{h})$ given an estimated state sequence of a hypothesized transcription. This is expressed by:

$$\hat{\mathbf{h}} = \arg\max_{\mathbf{h}} P(\mathbf{y}_L(\mathbf{h})\,|w) \tag{4}$$

where w is the hypothesized transcription, $P(\mathbf{y}(\mathbf{h})\,|w)$ is the likelihood of the observed features given the transcription considered and $\hat{\mathbf{h}}$ is the FIR parameter super-vector derived. The optimization is done via non-linear Conjugate Gradient. The state sequence can be estimated either using the array beamformed output (Unsupervised Limabeam) or, alternatively, assuming that the correct transcription is available (Oracle Limabeam). More details can be found in [5]. The Unsupervised Limabeam works well in noisy environments, even with a single channel. However, we found that preliminary experiments revealed two facts: first, the Oracle Limabeam performance on a single channel was close to the simple D&S on eight channels; second, there was still a margin of improvement between the Unsupervised and the Oracle Limabeam version applied to the multi-channel signals.

4. COMBINING N-BEST RECOGNITION AND ML BEAMFORMING

The Limabeam algorithm increases the likelihood of the first hypothesized transcription after a first recognition step. We propose to apply N-best parallel such optimizations: this approach is based on the fact that the N-best list, prior to parallel optimization, is ordered by likelihood and not necessarily by Word Error Rate (WER), which should be the optimal criterion. Applying the Limabeam algorithm on each hypothesis, the ranking order of the N-best list changes. We show at experimental level that the new hypothesis chosen (the new maximally likely) in this new list has, on average, a lower WER than the first chosen in the old list. The system is described in the following. For each of the N-best hypotheses [2] we derive a set of FIR filters:

$$\hat{\mathbf{h}}_n = \arg\max_{\mathbf{h}} P(\mathbf{y}_L(\mathbf{h})\,|w_n) \tag{5}$$

where w_n is the n-th hypothesized transcription at first recognition step, $P(\mathbf{y}(\mathbf{h})\,|w_n)$ is the likelihood of the observed features given the n-best transcription considered. Note that Equation (5) is equivalent to Unsupervised Limabeam when n is 1. After all the N-best FIR vectors are optimized in parallel, new features are calculated and recognition is performed. The transcription which gives the ML is then chosen:

$$\hat{n} = \arg\max_{n} P(\mathbf{y}_C(\hat{\mathbf{h}}_n)\,|\hat{w}_n) \tag{6}$$

where $\hat{w}_n$ is the transcription generated at second step recognition and $\hat{n}$ is the index of the most likely transcription, which is $\hat{w}_{\hat{n}}$. Note that the optimization is done in the LFBE domain, while recognition is done in the Cepstral domain as in [3]. We re-score likelihoods in the Cepstral domain as well. The system we propose is depicted in Figure 2. The signal coming from a microphone array is processed via conventional D&S, then Feature Extraction (FE) and a first recognition step is performed (REC). The HMM recognizer generates N-best hypotheses. For each hypothesis and in parallel, the Limabeam algorithm is applied: first a Viterbi alignment is performed (switch to 1: ALIGN) and fixed, then FIR coefficients are adaptively optimized via Conjugate Gradient (switch to 2: OPT). After convergence, the N-best features are recognized (switch to 3: REC) and another set of new transcriptions is produced. Finally, the last block compares the new N-best Log-LikeliHoods (LLH-rescoring) choosing the highest and the recognized sentence is produced.

5. EXPERIMENTAL RESULTS

Experiments were conducted using the HTK HMM-based recognizer [9] trained on the clean TI-digits corpus. Word models are represented by 18 state left-to-right HMMs. Output distributions are defined by 1 Gaussian pdf. The training set consists of 8440 utterances, pronounced by 110 speakers (55 men and 55 women). The test set consists of 104 phrases, pronounced by 52 men and 52 women. Both training and test data were up-sampled to 44.1 kHz via a polyphase filter prior to any processing. This sampling frequency was chosen in order to guarantee a high temporal resolution for the FIR filters and to ensure consistency with the real experimental framework being addressed [10]. In this work real noise coming from a computer fan was recorded in an office and added to the clean speech after being properly shifted channel by channel: this resulted in simulating a 8-microphone array with 7 cm inter-microphone distance and a source of fan noise in end-fire position. The FIR filters to be optimized

[2] Note that here "N-best" results from a preliminary reduction to a list that does not include repetitions of the same word sequence, which could be caused by different number and location of silences/background noise units.

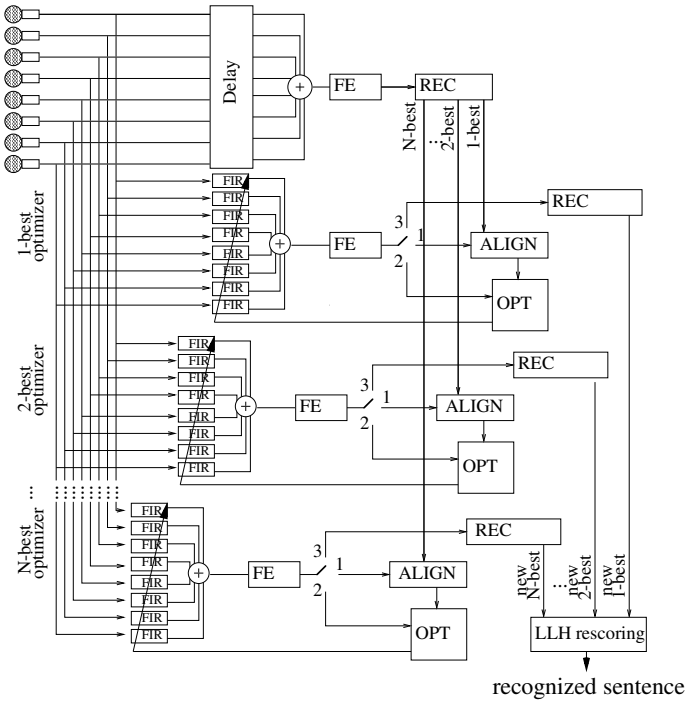

Figure 2: Block diagram of the N-best Unsupervised Limabeam.

are 10 taps long. The feature extraction involves 12 Mel Frequency Cepstral Coefficients (MFCC) and the log-Energy together with their first and second derivatives, for a total of 39 coefficients. Features were calculated every 10 ms, using a 25 ms sliding Hamming window. The frequency range spanned by the Mel-Scale filterbank was limited to 100-7500 Hz to avoid frequency regions with no useful signal energy. Cepstral Mean Normalization is applied. While recognition is performed in the cepstral domain, the optimization process is done in the LFBE domain using 24 coefficients for the features and single-Gaussian output distributions [3] for the models, but without CMN. In our implementation of the Limabeam algorithm the Conjugate Gradient algorithm [11] is the same adopted in [3] and no modifications were applied to the original one. This was done to ensure compliance with Seltzer's work.

Preliminary results show the usefulness of a microphone array in a noisy environment. Table 1 reports the performance obtained with D&S, with Unsupervised Limabeam and with Oracle Limabeam (transcription is known). In this case the amount of fan and office noise caused a SNR of -5 dB. The Oracle on one channel performs only slightly better than the simple D&S on eight channels. Furthermore, note that Unsupervised Limabeam gives 16% relative improvement over D&S, while performance could be improved up to a 35%.

-5 dB	D&S	Uns. Lim.	Oracle Lim.
1 ch	63.79%	66.11%(6%)	70.43%(18%)
8 ch	69.10%	74.09%(16%)	80.07%(35%)

Table 1: Usefulness of Limabeam with respect to a single microphone. Relative improvements over D&S are reported in parentheses. Results are in terms of digit accuracy.

We then applied the N-best Unsupervised Limabeam to the same test set. Figure 3 depicts its behavior at -5 dB in terms of digit accuracy against the number of N-best hypothesis considered. The solid line corresponds to D&S performance, while the upper flat line corresponds to the Oracle Limabeam. Except for the 2-best and the 3-best, where a major boosting in accuracy is observed, performance grow roughly linearly. When 19 hypotheses are considered in parallel, the N-best Unsupervised Limabeam is comparable to the Oracle Limabeam, showing a 34.4% relative improvement: in practice we achieved the same performance of a supervised algorithm in an unsupervised manner. The non-monotonic behavior is due to the mismatch between the maximum likelihood and the minimum WER criterion: especially if few hypotheses are considered (e.g 3 or 5 on this task, see Figure 3), after LLH rescoring some new maximally likely transcriptions can have a lower WER. The key is to consider as many hypotheses as possible, because this increases the probability of having a transcription which minimizes the WER as well. Further experiments with higher SNRs did not show significant improvements over D&S: Table 2 shows that the major gain of the N-best Unsupervised Limabeam is at low SNRs.

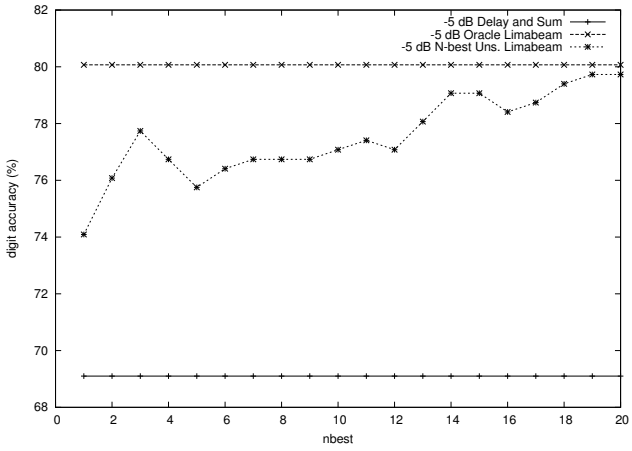

Figure 3: Performance of the N-best Unsupervised Limabeam compared to D&S and to the supervised version of Limabeam at -5 dB: the N-best approach makes the unsupervised optimization comparable to Oracle Limabeam when N becomes high. For N=1 the system is equivalent to the original Unsupervised Limabeam.

	D&S	Uns.Lim.	N-best Lim.	Oracle Lim.
15dB	98.34%	98.34%	98.34% (1)	98.34%
5 dB	95.68%	96.35%	96.68% (9)	96.68%
-5 dB	69.10%	74.09%	79.73% (19)	80.07%

Table 2: Comparison among D&S, Limabeam and N-best Unsupervised Limabeam across different SNRs. At higher SNRs the use of a ML-based beamformer does not seem to provide significant improvement, while at low SNRs our N-best approach performs almost as the Oracle Limabeam. The performance is the one relative to the n-best in parenthesis. Results are in terms of digit accuracy.

6. DISCUSSION AND FUTURE ACTIVITY

We found that an N-best approach to Unsupervised Limabeam, which optimizes in parallel a set of FIR filters on the first N-best hypotheses in an unsupervised manner, is able to reach performance comparable to the Oracle Limabeam. This approach is an extension of the Limabeam algorithm, which has the advantage of modifying the FIR coefficients via a ML criterion.

One could expect that the N-best hypotheses of the first recognition step should be optimized in the same way and, consequently, their ranks should not change significantly. This is generally not true. In fact an *n*-best hypothesis can have, after the first recognition step, a lower likelihood with respect to the 1-best, and a better WER at the same time. Multiple instances of the Limabeam algorithm generate different Viterbi alignments, objective functions to be minimized and FIR supervectors: it is our intuition that some of these filters significantly increase (with respect to the first step) the relative likelihoods of the minimal WER hypotheses. This may be due to the fact that the Viterbi alignment is closer to the correct transcription. Further work is planned to clarify this aspect.

We also point out that in noisy environments it is not strictly necessary to have the correct transcription, among the N-best, to generate a more likely transcription, as the literature states [7].

In conclusion, this work showed that performing N-best recognition is a way to improve the potential of Limabeam algorithm. The effectiveness of the proposed solution results from the synergy between a multi-channel signal processing and a multi-pass search strategy, which allows to recover errors introduced in early recognition steps. The robustness of the N-best Unsupervised Limabeam is still to be proved with a larger real database, recorded in reverberant environments, as meeting rooms, where such application would be desirable, but in which the overall performance is likely to decrease. Preliminary experiments on real-data indicated that the margin between D&S and the Oracle is smaller than in the here presented simulation. This is probably related to the regularities characterizing simulations versus variabilities in real-world experiments. As typical meeting room impulse responses are even more than 600ms long, the FIR filters length would be much higher. The temporal resolution of the filters should be guaranteed by the high sampling frequency (44.1 kHz) adopted, which complies with the microphone array we plan to use in our future studies [10]. To this regard, other issues to address in the next work will refer to the dimensionality of the problem (number of taps, number of microphones etc...) and to a possible reduction of the system complexity.

7. ACKNOWLEDGEMENTS

The authors would like to thank Michael Seltzer for clarifications about the Limabeam algorithm. L. Brayda would also like to acknowledge the French MESR (Ministère de l'Einsegnement Supérieur et de la Recherche) and Istituto Trentino di Cultura for having supported this work.

REFERENCES

[1] M. Brandstein and D. Ward, *Microphone arrays - signal processing techniques and applications*, New York: Springer-Verlag, 2001.

[2] Acero A., *Acoustical and environmental robustness in automatic spech recognition.*, Boston, MA:Kluwer Academic Publishers, 1993.

[3] Seltzer M., Raj B., and Stern R. M., "Likelihood-maximizing beamforming for robust hands-free speech recognition," in *IEEE Trans. on Speech and Audio Procesing*, 2004, vol. 12, no, 5.

[4] Bitzer J., Simmer U., and Kammeyer K.D., "Theoretical noise reduction limits of the generalized sidelobe canceler (gsc) for speech enhancement," in *Proceedings of ICASSP*, Phoenix, AZ, USA, 1999.

[5] Seltzer M., *Microphone array processing for robust speech recognition*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2003.

[6] Seltzer M. and Raj B., "Speech recognizer-based filter optimization for microphone array processing," in *IEEE Signal Processing Letters*, 2003, vol. 10, no, 3.

[7] Huang X., Acero A., and Hon H., *Spoken Language Processing*, Carnegie Mellon University, 2001.

[8] Flanagan J.L., Surendran A.C., and Jan E.E., "Spatially selective sound capture for speech and audio processing," *Speech Communication*, 1993.

[9] Young S. et al, *The HTK Book Version 3.0.*, Cambridge University, 2000.

[10] Brayda L., Bertotti C., Cristoforetti L., Omologo M., and Svaizer P., "Modifications on NIST MarkIII array to improve coherence properties among input signals," in *AES, 118th Audio Engineering Society Convention, Barcelona, Spain*, 2005.

[11] Press W. et al, *Numerical recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1988.