COMBINING SPECTRAL FEATURE MAPPING AND MULTI-CHANNEL MODEL-BASED SOURCE SEPARATION FOR NOISE-ROBUST AUTOMATIC SPEECH RECOGNITION

Deblin Bagchi, Michael I. Mandel, Zhongqiu Wang, Yanzhang He, Andrew Plummer, Eric Fosler-Lussier*

Department of Computer Science and Engineering
The Ohio State University, Columbus, OH, USA

ABSTRACT

Automatic Speech Recognition systems suffer from severe performance degradation in the presence of myriad complicating factors such as noise, reverberation, multiple speech sources, multiple recording devices, etc. Previous challenges have sparked much innovation when it comes to designing systems capable of handling these complications. In this spirit, the CHiME-3 challenge presents system builders with the task of recognizing speech in a real-world noisy setting wherein speakers talk to an array of 6 microphones in a tablet. In order to address these issues, we explore the effectiveness of first applying a model-based source separation mask to the output of a beamformer that combines the source signals recorded by each microphone, followed by a DNN-based front end spectral mapper that predicts clean filterbank features. The source separation algorithm MESSL (Model-based EM Source Separation and Localization) has been extended from two channels to multiple channels in order to meet the demands of the challenge. We report on interactions between the two systems, cross-cut by the use of a robust beamforming algorithm called BeamformIt. Evaluations of different system settings reveal that combining MESSL and the spectral mapper together on the baseline beamformer algorithm boosts the performance substantially.

Index Terms— Robust Automatic Speech Recognition, Deep Neural Networks, Spectral Feature Mapping, Multi-channel Model-based Source Separation, Beamforming

1. INTRODUCTION

State-of-the-art ASR systems seem to be performing satisfactorily in clean environments. However, modern systems can perform rather poorly in the presence of factors like additive noise and reverberation delays. As a result, most of the research in the field has shifted towards making more robust ASR systems which can perform well even in noisy scenarios. The field has set for itself challenges with increasing difficulty in the last few years: In the CHiME-2 challenge [1], clean binaural speech was corrupted with simulated noise corresponding to different SNR levels inside a family living-room setting. Many groups, including ours, have worked with this dataset to improve performance. The CHiME-3[2] dataset extends the difficulty by providing not only artificially noisy speech, made by combining clean speech with recorded background noise, but also noisy speech recorded in public environments, like a cafe, a bus, a street junction and pedestrian areas. The latter data comes from people talking to a tablet with an array of six microphones, giving rise to important challenges like how to account for change of phase and signal strength across different microphones, noise suppression, dealing with attenuated speech within the signal. Systems are also exposed to different kind of noises that have never been dealt with before.

In this work, we have extended MESSL (Model-Based EM Source Separation and Localization), a source separation algorithm for two-microphone recordings, to multiple microphones, as required by the challenge. We have also attempted to make good use of the CHiME-3 stereo data with the help of a Deep Neural Network (DNN) based spectral mapper. Since MESSL uses both temporal and spatial information, and the front-end spectral mapper uses spectral information for enhancing the features, it seemed interesting to investigate whether combining the two approaches might be complimentary and bring about a gain in performance. Therefore, we compared the interactions of these two with different beamforming algorithms — the baseline enhancement beamformer provided by the CHiME-3 challenge (Minimum-Variance Distortionless Response Beamformer, or MVDR) and BeamformIt [3], a beamformer from ICSI. Our experiments show that MESSL performs well in both cases, whereas the spectral mapper shows improvements only when used in conjunction with the baseline beamformer enhancement algorithm.

The remainder of this paper is organized as follows: we outline related work in Section 2. Subsequently in Section 3 we describe how we integrated our previous model-based source separation technique with both the baseline MVDR and BeamformIt beamformers, requiring an extension of the MESSL algorithm to multiple microphones. We also discuss the addition of a spectral mapper trained to transduce beamformed output into clean filterbank features. Section 4 reports on the evaluation of our systems. We conclude with a brief discussion (Section 5) of the evaluation of our systems and their potential

*Yanzhang He is currently with Google.

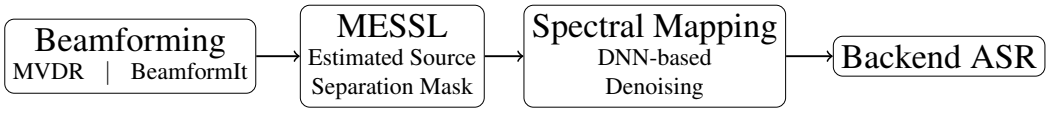

Fig. 1. System diagram for combining beamforming, multi-channel MESSL, and spectral feature mapping.

to serve as a basis for future investigations.

2. RELATED WORK

Source separation and noise suppression have a long history in automatic speech recognition. While early systems [4] were only able to suppress stationary noise like those from fans and ventilation systems, more recent results are able to suppress or separate non-stationary noise [5], including simultaneous talkers [6]. One popular recent approach is to predict a spectral mask that indicates the degree to which speech or noise dominates individual time-frequency points of a spectrogram [7]. These mask-based separation approaches can be driven by spectral [5] or spatial [8] information, and both have been shown to reduce WERs significantly [5]. Model-based EM Source Separation and Localization (MESSL) [9] predicts these masks based on spatial information. Once a mask is estimated, the simplest way to utilize it in ASR is to simply apply it as a gain to the spectrogram and apply subsequent processing unchanged. Such an approach has been shown to work well if cepstral mean and variance normalization are applied [10]. More sophisticated approaches include treating the regions where the mask is 0 as uncertain, but not silent, and imputing the missing spectral information [11, 12] or accounting for this uncertainty in the recognition process [13, 14].

More recently, deep neural network based approaches are shown to be effective for feature learning and robust to environment variations, particularly suited to using spectral information for improving noise robustness in ASR as well as for speech enhancement systems. DNNs can be used for acoustic modeling that are trained directly with filterbank features in conjunction with simple noise estimates [15]. Time-frequency masks for speech separation can also be estimated with DNNs using spectral information [5], which can further be jointly trained with the acoustic model in a single DNN framework [16, 17]. Alternatively, one can learn a spectral mapper to transform noisy features into clean features and use them directly as inputs to ASR, which can be denoising auto-encoders [18, 19, 20], or deep/recurrent neural networks [21, 22, 23, 24]. The transformed features can also be used to reconstruct the speech waveform [25, 22]. These approaches have been shown to work well where stereo (noisy and clean) data are available.

3. SYSTEM DESIGN

Our system tests out two major components, MESSL enhancement and Spectral Mapping, in the context of two different standard beamforming techniques. Figure 1 gives a block diagram of the overall system. In this section, we step through the different components of the system.

3.1. Beamformers

The CHiME-3 challenge provided a baseline enhancement system (minimum-variance distortionless-response, MVDR), which performs tracking and smoothed estimation of dominant source location over time, and then, with a noise spatial covariance estimate, creates a subband minimum-variance distortionless response (MVDR) beamformer. The MVDR beamformer uses its modeling power to both preserve the signal coming from the target direction and to cancel noise signals coming from other directions, as captured in the spatial covariance matrix.

We also experimented with BeamformIt [3], a well-engineered source tracker and delay-and-sum beamformer. It tracks the dominant source in a mixture over time using cross-correlations between microphone pairs and then performs a Viterbi decoding to eliminate spurious time delay estimates. It uses these estimates to perform delay-and-sum beamforming in the identified target direction.

While the MVDR beamformer should provide better cancellation performance, we found that the source localization of the baseline system was less robust than that of BeamformIt, leading to incorrect look directions, and severe signal degradation. As we discuss in the next section, we used the beamforming output to initialize MESSL. Therefore, a failed localization often led to a failed separation, although MESSL could sometimes recover from such a failure.

3.2. MESSL Enhancement

We augmented the beamforming systems (baseline MVDR and BeamformIt) with a post-filter based on the output of Model-based EM Source Separation and Localization (MESSL) [9]. Because MESSL was developed for binaural recordings, we extended it to the multichannel case by modeling every pair of channels with its own MESSL model, combining them in the E-step of the EM algorithm. This required some care in initialization, but mathematically the extension to the EM algorithm is relatively straightforward.

MESSL performs source separation by clustering time-frequency points based on their interaural phase and level differences (IPD and ILD). It includes a latent variable representing interaural time difference (ITD) that connects the IPD

models across frequency, solving the source permutation problem common to similar sub-band localization approaches. In addition, this ITD latent variable allows MESSL to overcome limitations imposed by spatial aliasing on similar systems, because even though the IPD-to-ITD mapping is ambiguous in the face of IPD phase wrapping at high frequencies, the ITD-to-IPD mapping used by MESSL is unambiguous at all frequencies.

The total log likelihood that MESSL maximizes is

$$\mathcal{L}_p(\Theta;\phi,\alpha) = \sum_{\omega t} \log p(\phi(\omega,t),\alpha(\omega,t)\,|\,\Theta) \tag{1}$$

$$= \sum_{\omega t} \log \sum_{k\tau} \Big[p(z_{k\tau}(\omega,t)\,|\,\Theta) \tag{2}$$

$$\cdot\, p(\phi(\omega,t),\alpha(\omega,t)\,|\,z_{k\tau}(\omega,t),\Theta)\Big]$$

where $\phi(\omega,t)$ represents the IPD observations at each TF point, $\alpha(\omega,t)$ the ILD observations in dB, Θ the model parameters, and $z_{k\tau}(\omega,t)$ the hidden variable representing which source and delay each TF point comes from. The likelihood $p(\phi(\omega,t),\alpha(\omega,t)\,|\,z_{k\tau}(\omega,t),\Theta)$ is Gaussian, making this a Gaussian mixture model. MESSL then performs separation using an EM algorithm. In the E-step, it estimates $p(z_{k\tau}(\omega,t)\,|\,\phi(\omega,t),\alpha(\omega,t),\Theta)$, the posterior probability of each TF point coming from each source model. This provides the time-frequency mask that can be used to separate each source from the mixture. In the M-step, it re-estimates the IPD, ILD, and ITD parameters for each model, Θ, from the masks and the IPD and ILD observations.

In order to separate mixtures observed with more than two microphones, we employ the pair-wise model on every pair of channels. Thus, multichannel MESSL maximizes the following total log likelihood

$$\mathcal{L}(\Theta;\phi,\alpha) = \frac{2}{N} \sum_{i<j=1}^{N} \mathcal{L}_p(\Theta_{ij};\phi_{ij},\alpha_{ij}) \tag{3}$$

Adding together the pairwise log likelihoods is perhaps the simplest way to combine these models and makes the assumption that they are independent of one another. Because for N microphones there are $\binom{N}{2} = N(N-1)/2$ microphone pairs, but only $N-1$ independent sources of pairwise information, the assumption that all $\binom{N}{2}$ observations are independent causes an over-counting of evidence by a factor of $\frac{N}{2}$. The normalization of $\frac{2}{N}$ in (3) aims to correct this over-counting.

We then use an EM algorithm that coordinates the pairwise models through their masks, as shown in Figure 2. The E-step computes global, τ-independent masks

$$\nu_k(\omega,t) \propto \prod_{i<j} \Big(\sum_{\tau} p(\phi_{ij},\alpha_{ij}\,|\,z_{k\tau},\Theta_{ij}) \Big)^{2/N} \tag{4}$$

where the dependence of ϕ_{ij}, α_{ij}, and $z_{k\tau}$ on (ω,t) has been omitted. These then modify the pairwise, τ-dependent masks

$$\nu_{k\tau}^{(ij)}(\omega,t) \propto p(\phi_{ij},\alpha_{ij}\,|\,z_{k\tau},\Theta_{ij}) \cdot \nu_k(\omega,t) \tag{5}$$

which are used to update the parameters of the pairwise models in the M-step.

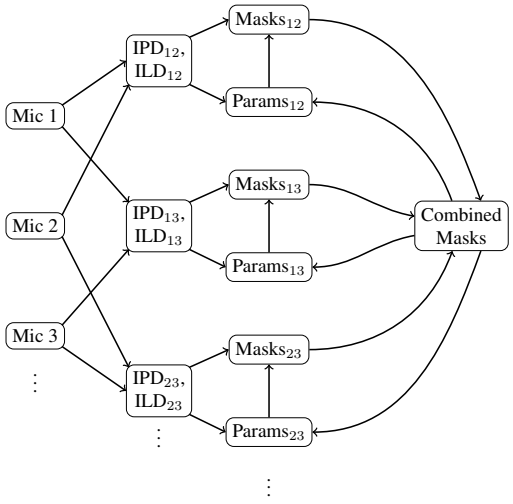

Fig. 2. System diagram showing E and M steps of multichannel MESSL.

This approach makes no use of array geometry information, making it robust to array mis-calibration and even completely novel array configurations, but possibly limiting its performance when the array geometry is known, as in CHiME3. Without knowledge of array geometry, it is difficult to find a correspondence between the parameters of the models of different microphone pairs. The time-frequency mask dominated by each source, however, should be very consistent across all microphones, and thus across microphone pairs.

Initializing the multi-channel model requires initializing the pair-wise models and coordinating the source models across microphone pairs. While there are many possibilities, we explored initialization from ITDs derived from pairwise cross-correlations and initialization from masks derived from level differences between the beamformer output and the rear-facing microphone 2. In pilot experiments, the mask-based initialization performed best, and also had the advantage of aligning the source models across microphone pairs without a separate alignment step. Because the CHiME3 data consists of a single target talker against distant background noise, we model two sources: the target speaker, which is modeled as a point source, i.e., having a single dominant ITD, and everything else, which is modeled as approximately diffuse, having a uniform distribution across ITD and a zero-mean, wide-variance Gaussian for ILD.

We also utilize the recently introduced extension to MESSL of inserting Markov random field mask smoothing between the E and M steps [26] using the sum-product variant

of loopy belief propagation (LBP). We furthermore introduce here the use of the max-product variant of LBP to find the maximum *a posteriori* binary mask in the same Markov random field model after the last EM iteration, estimating a globally consistent binary mask for each source from the soft masks estimated by MESSL.

Preliminary experiments showed that using all pairs of microphones led to higher quality separations than designating a single microphone as reference. For the CHiME3 setup, designating the rear-facing microphone 2 as the reference led to more useful ILD cues, but less useful IPD cues. Similarly, designating one of the front-facing microphones as reference led to more reliable IPD cues, but less reliable ILD cues. A model using all pairs of microphones is able to take advantage of all of these varied relationships.

3.3. Spectral Mapping

In addition to the spatial patterning learned by MESSL, utilizing spectral mapping can also improve performance by learning the mapping of spectral patterns to filterbank outputs. We train a DNN-based spectral mapper for feature denoising. In our previous work [21], we have shown that a DNN-based spectral mapper, which takes noisy spectrogram as input to predict clean filterbank features for ASR, yields good results on the CHiME-2 noisy and reverberant dataset. In order to test the performance of this technique on a more real-world setting, we use it in conjunction with various beamforming and mask estimation systems in CHiME-3.

We extract spectrogram features from noisy speech as the input to the spectral mapper. Specifically, we first divide the input time-domain signals into 25-ms frames with a 10-ms frame shift, and then apply short time Fourier transform (STFT) to compute log spectral magnitudes in each time frame. For a 16 kHz signal, each frame contains 400 samples,and we use 512-point Fourier transform to compute the magnitudes, forming a 257-dimensional log magnitude vector. Temporal dynamics and feature splicing are known to improve performance, so we use deltas and double-deltas with a splice context window of 5. Hence the dimensionality of the input is 257 x 3 x 5 = 3855.

The output target of the spectral mapper is the clean filterbank features of 40 channels for the central frame of the context window. The objective function for optimization is mean square error (MSE). We use the entire CHiME-3 training set consisting of 7138 simulated noisy utterances and 1600 real noisy utterances to train the spectral mapper. For the simulated utterances, we can extract the ground truth from the parallel clean Wall Street Journal (WSJ0) corpus SI84 training set. For the real utterances, we use the close microphone channel as an approximation to the clean ground truth.

The architecture of the network is shown in Figure 3. We use 2 hidden layers and 4096 sigmoid neurons in each layer. The input features are globally normalized to have zero mean and unit variance over all feature vectors in the training set,

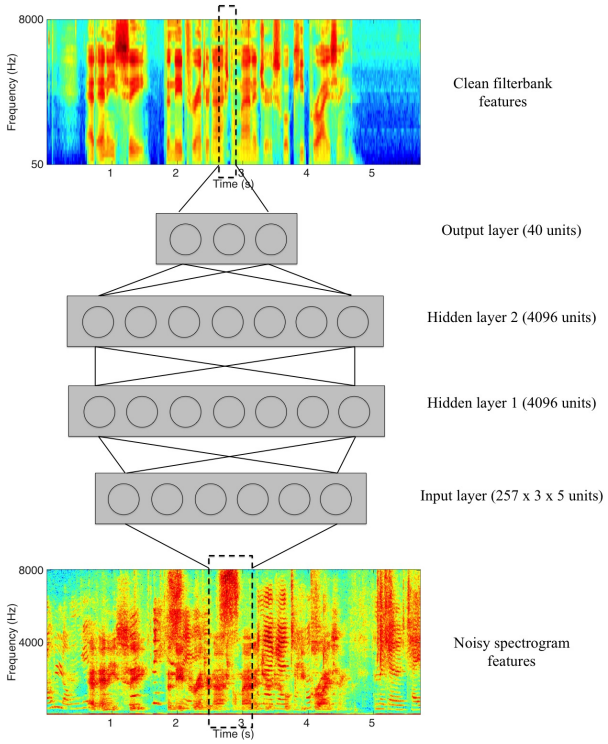

Fig. 3. Architecture of the DNN-based spectral mapper. The inputs are the noisy log spectra of 5-frame context window with deltas and double deltas, and the outputs are the clean filterbank features of the central frame.

and filterbank training labels are normalized into the range of [0,1] matched by the output layer sigmoid units. Training uses back-propagation with mini-batch stochastic gradient descent, and the optimization technique uses adaptive gradient descent along with a momentum term. After training the spectral mapper, we apply it to both training and test sets to generate filterbank features as input to the backend ASR DNN system.

4. EVALUATION

4.1. Description of Acoustic Model

We have used the CHiME-3 recipe of the KALDI toolkit [27] to build the ASR system. We have first trained a GMM-HMM acoustic model. We have applied deltas and double-deltas on 13-dimensional MFCC features and used 7 frames of splicing context window. These features are then decorrelated and compressed into 40 dimensions using Linear discriminant analysis (LDA). Further decorrelation was done using Maximum Likelihood Linear Transform (MLLT). Feature-space maximum likelihood linear regression (fMLLR) was applied on the resulting features, which is estimated by speaker adaptive training(SAT), reducing speaker variance. The GMM-HMM system was a triphone system with around 2000 senone (tied triphone state) targets.

Beamformer	MESSL	SpecMap	Dev WER		Test WER	
			Sim	Real	Sim	Real
None	No	No	13.4	14.6	17.7	28.1
None	No	Yes	16.0	18.5	22.4	35.5
MVDR	No	No	**8.5**	19.4	**11.6**	39.4
MVDR	No	Yes	9.2	17.2	11.9	34.7
MVDR	Yes	No	10.4	13.7	13.2	31.2
MVDR	Yes	Yes	11.4	14.0	16.3	30.3
BeamformIt	No	No	11.2	9.4	21.1	18.1
Beamformit	No	Yes	12.9	11.1	22.2	21.9
BeamformIt	Yes	No	11.5	**9.0**	21.0	**16.3**

Table 1. Experimental results on the CHiME3 development and test sets, broken down across simulated and real recordings. Bold indicates the best performing system for each evaluation subset.

The DNN-HMM hybrid system was trained using the alignment generated by the GMM-HMM system. The DNN has 7 hidden layers, with 2048 sigmoid neurons in each layer and a softmax output layer. Splicing context size for the filter-bank features was fixed at 11, with the minibatch-size being 256. Following [28], after DNN training, we realign the data with the trained DNN and retrain the DNN using the new alignment. We repeat this process for three times until the performance become saturated. After that, we train the DNN with sMBR sequence training to achieve better performance. We regenerate the lattices after the first iteration and train for 4 more iterations. Note that we used the improved sequence training proposed in [29] in the latest Kaldi version.

4.2. Results

Table 1 shows the results of experiments that combine variations in beamforming (none,[1] MVDR, Beamformit), use of MESSL, and use of Spectral Mapping; we report word error rates broken down on the development and test sets between real recordings and simulations. Results on real and simulated data are quite negatively correlated with one another, i.e., techniques that help one tend to hurt the other. In particular, this can be seen in the performance of the different beamformers without any other augmentation. The baseline MVDR beamformer reduces WER on simulated test data by 6.1 percentage points, but increases WER on the real test recordings by 11.3 percentage points over a single channel. BeamformIt, on the other hand, increases WER on the simulated data by 3.5 percentage points, and reduces WER on real recordings by 10.2 percentage points. When additional processing is applied on top of the beamformed results, the trend continues for the most part. The real recordings test the abilities of these systems to function in the real world, as they would be in a product, thus we focus on those results as our ultimate measure of success. Performance on the synthetic mixtures, compared to the real recordings, seems to suggest that the simulations are not very reflective of reality.

Because the shortcomings of the baseline MVDR beamformer and the advantages of BeamformIt only became apparent late in the evaluation process once the test set was released, we did not have time to run a full combination of systems on top of BeamformIt. Results with the baseline beamformer, which are consistent between development and test sets, show that the addition of both MESSL and spectral mapping decrease WER, and that these performance improvements are somewhat complementary, leading to the best performance on real test data of 30.3% WER for the combined system: 9.1 percentage points lower than the WER of the baseline beamformer alone, but still 2.2 percentage points higher than using a single noisy microphone. On top of BeamformIt, MESSL was able to improve performance on the real test data by an absolute 1.8 percentage points, although there is no noticeable improvement in the simulated test utterances. It is interesting that the front-end spectral mapper makes matters worse when used on top of Beamformit, increasing WER by 3.8 % in the real test set and 1.1% in the simulated test set. We also ran the spectral mapper on the single channel data, and found that this also increases error, indicating that there may be significant mismatch between our ground truth and the source signal.

Table 2 shows the WERs for BeamformIt alone and BeamformIt+MESSL, the best system, broken down by noise condition and across the various datasets. In terms of conditions, the bus seemed to be consistently the most difficult in the real data. The bus recordings were also most difficult for MESSL, which gave much larger improvements on the other conditions. This might be because MESSL assumes sources are spatially stationary throughout an entire utterance, while on the bus there could have been more movement of the talker relative to the microphones. Utilizing an online EM [30] implementation of MESSL would relax this assumption and might improve performance on the bus recordings.

5. DISCUSSION

While the overall trend of MESSL masks and spectral mapping improving performance separately and together was more or less expected, the trends between beamformers, between real and synthetic data, and between development and test sets were not. In particular, as shown in Tables 1 and 2, the test data turned out to be much harder than the development data, especially for the baseline beamformer. It is not entirely clear why this would be the case, but it could have to do with the nature of the subjects who were recruited for each dataset. Another possibility is that the baseline beamformer was optimized for the development set, and then had difficulty generalizing to the test set. This issue was only discovered when the test set was released, and it took some time to identify the problem and a solution for it, leaving less time to run extensive experimentation on top of the BeamformIt output.

[1] Here, *none* means that we take a single channel as reference and ignore all other channels.

System	Recordings	Set	BUS	CAF	PED	STR	Average
BeamformIt	Simulated	Dev	9.8	13.8	9.2	12.2	11.2
BeamformIt + MESSL	Simulated	Dev	13.4	12.2	10.0	10.5	11.5
BeamformIt	Simulated	Test	14.9	22.2	23.4	24.1	21.1
BeamformIt + MESSL	Simulated	Test	23.1	17.3	18.8	24.9	21.0
BeamformIt	Real	Dev	11.5	9.0	6.8	10.2	9.4
BeamformIt + MESSL	Real	Dev	12.5	7.8	6.1	9.7	9.0
BeamformIt	Real	Test	24.8	17.8	16.3	13.3	18.1
BeamformIt + MESSL	Real	Test	24.7	14.0	13.7	12.9	16.3

Table 2. Comparison of WERs for BeamformIt by itself (baseline) and followed by MESSL (best system) on real and simulated recordings in both dev and test sets broken down by environment: bus (BUS), café (CAF), pedestrian (PED), and street (STR).

In addition, the output of BeamformIt differed in at least two ways from that of the baseline beamformer, which made running additional systems on top of it more complicated. Specifically, it applies a time-varying (and utterance-specific) gain to its output, and appears to introduce an utterance-specific delay of several milliseconds between its output and the mic signals, including the reference close-talking mic. Both of these differences make it difficult to use the spectral mapper. For the delay, the spectral mapper is essentially expected to predict an arbitrary delay of a frame in either direction. Utilizing a context of multiple neighboring frames around an input frame helped to deal with this problem, but could not solve it because of the seemingly random nature of the delays. For the gain, the spectral mapper is expected to predict the original gain of the reference signal from a gain-normalized input file. We did not have time to find a solution to this problem, hence the worse results in Table 1 for BeamformIt+SpecMap.

These differences in BeamformIt output also created a difficulty for MESSL's initialization. In particular, because MESSL was initialized using a mask calculated from the ILD between the beamformer output and the rear-facing mic 2, gain variations in the beamformer output led to shifts of this ILD derived from it. To overcome this issue, we switched from generating the mask from an absolute ILD threshold to a relative threshold: the 70th percentile of ILD values.

Another unforseen difficulty that only arose on the output of BeamformIt was MESSL's introduction of musical noise and spectral coloring. In applying the binary mask derived from MESSL's estimates to the signal, we originally enforced a maximum suppression of 40 dB. This turned out to be much too harsh for subsequent ASR processing, and reducing this maximum suppression to 9 dB, at which artifacts just started to be audible, significantly improved ASR performance. This number was not tuned, and we plan further experiments to find the optimal value. This result implies that more suppression might be possible if we were to use a more sophisticated combination of the mask with the ASR process, such as the analysis-by-synthesis approach proposed in [31].

The original design of our system utilized an additional set of robust ASR features [17] on top of the output of the beamformer+MESSL+spectral mapping. Adding robust features to the system, including PNCC [32], MRCG [33], AMS [34], RASTA-PLP [35] and MFCC features, has been shown to improve spectral mapping performance in CHiME-2 [17, 36, 37]. Due to time constraints, we were only able to augment one system with these features, with surprisingly equivocal results. We plan to investigate why these supplemental features failed to produce additional improvements on the CHiME-3 dataset.

6. CONCLUSION

This paper has described the combination of three system components for increasing the noise robustness of an ASR front end: beamforming, a mask-based post-filter, and spectral mapping. For beamforming, we found that BeamformIt led to much lower word error rates than the baseline beamformer and a single noisy microphone. For post-filtering, we introduced a multi-channel variant of MESSL, which generates a spectral mask by spatially clustering time-frequency points, and found that it reduced WERs on top of all beamformers. For spectral mapping, we found that a DNN-based front end spectral mapper that predicts clean filterbank features from noisy spectra reduced WERs on top of the baseline beamformer, in a way that was complementary to MESSL. This improvement did not hold on top of BeamformIt, most likely because of a data mismatch problem, an unknown gain or delay present in the output of BeamformIt.

7. ACKNOWLEDGMENTS

The authors would like to thank Shinji Watanabe for discussions about beamforming and processing the input mixtures using BeamformIt. This material is based upon work supported by the NSF under Grant No. IIS-1409031.

8. REFERENCES

[1] Emmanuel Vincent, Jon Barker, Shinji Watanabe, Jonathan Le Roux, Francesco Nesta, and Marco Matassoni, "The second 'CHiME' speech separation and recognition challenge: Datasets, tasks and baselines," in *Proc. ICASSP*, 2013, pp. 126–130.

[2] Jon Barker, Ricard Marxer, Emmanuel Vincent, and Shinji Watanabe, "The third 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *Proc. ASRU*, 2015, to appear.

[3] Xavier Anguera, Chuck Wooters, and Javier Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE Tr. ASLP*, vol. 15, no. 7, pp. 2011–2022, Sept. 2007.

[4] B.H. Juang, "Speech recognition in adverse environments," *Comp. Speech & Lang.*, vol. 5, no. 3, pp. 275–294, July 1991.

[5] Arun Narayanan and DeLiang Wang, "Investigation of speech separation as a front-end for noise robust speech recognition," *IEEE/ACM Tr. ASLP*, vol. 22, no. 4, pp. 826–835, Apr. 2014.

[6] John Hershey, Steven Rennie, Peder Olsen, and Trausti Kristjansson, "Super-human multi-talker speech recognition: A graphical modeling approach," *Comp. Speech & Lang.*, vol. 24, no. 1, pp. 45–66, 2010.

[7] DeLiang Wang, "On Ideal Binary Mask As the Computational Goal of Auditory Scene Analysis," in *Speech Separation by Humans and Machines*, Pierre Divenyi, Ed., pp. 181–197. Springer US, Boston, 2005.

[8] Nicoleta Roman, DeLiang Wang, and Guy Brown, "A classification-based cocktail party processor," in *NIPS*, 2003, pp. 1425–1432.

[9] Michael I. Mandel, Ron J. Weiss, and Daniel P. W. Ellis, "Model-based expectation maximization source separation and localization," *IEEE Tr. ASLP*, vol. 18, no. 2, pp. 382–394, Feb. 2010.

[10] William Hartmann, Arun Narayanan, Eric Fosler-Lussier, and DeLiang Wang, "A Direct Masking Approach to Robust ASR," *IEEE Tr. ASLP*, vol. 21, no. 10, pp. 1993–2005, Oct. 2013.

[11] Bhiksha Raj and Rita Singh, "Reconstructing noise-corrupted spectrographic components for robust speech recognition," in *Robust speech recognition of uncertain or missing data: Theory and applications*, Dorothea Kolossa and Reinhold Haeb-Umbach, Eds., chapter 6, pp. 127–156. Springer, 2011.

[12] Jort Florent Gemmeke and Ulpu Remes, "Missing-data techniques: Feature reconstruction," in *Techniques for Noise Robustness in Automatic Speech Recognition*, Tuomas Virtanen, Bhiksha Raj, and Rita Singh, Eds., chapter 15, pp. 399–432. John Wiley & Sons, Ltd, 2012.

[13] Hank Liao, "Uncertainty decoding," in *Techniques for Noise Robustness in Automatic Speech Recognition*, Tuomas Virtanen, Bhiksha Raj, and Rita Singh, Eds., chapter 17, pp. 463–486. John Wiley & Sons, Ltd, 2012.

[14] Dorothea Kolossa and Reinhold Haeb-Umbach, Eds., *Robust speech recognition of uncertain or missing data: Theory and applications*, Springer, 2011.

[15] Michael L. Seltzer, Dong Yu, and Yongqiang Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. ICASSP*, 2013, pp. 7398–7402.

[16] Arun Narayanan and DeLiang Wang, "Joint noise adaptive training for robust automatic speech recognition," in *Proc. ICASSP*, 2014, pp. 2523–2527.

[17] Zhong-Qiu Wang and DeLiang Wang, "Joint training of speech separation, filterbank and acoustic model for robust automatic speech recognition," in *Proc. Interspeech*, 2015, pp. 2839–2843.

[18] Takaaki Ishii, Hiroki Komiyama, Takahiro Shinozaki, Yasuo Horiuchi, and Shingo Kuroiwa, "Reverberant speech recognition based on denoising autoencoder.," in *Proc. Interspeech*, 2013, pp. 3512–3516.

[19] Xue Feng, Yaodong Zhang, and James Glass, "Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition," in *Proc. ICASSP*, 2014, pp. 1759–1763.

[20] Felix Weninger, Shinji Watanabe, Yuuki Tachioka, and Björn Schuller, "Deep recurrent de-noising auto-encoder and blind de-reverberation for reverberated speech recognition," in *Proc. ICASSP*, 2014, pp. 4623–4627.

[21] Kun Han, Yanzhang He, Deblin Bagchi, Eric Fosler-Lussier, and DeLiang Wang, "Deep neural network based spectral feature mapping for robust speech recognition," in *Proc. Interspeech*, 2015, pp. 2484–2488.

[22] Kun Han, Yuxuan Wang, DeLiang Wang, William S Woods, Ivo Merks, and Tao Zhang, "Learning spectral mapping for speech dereverberation and denoising," *IEEE/ACM Tr. ASLP*, vol. 23, no. 6, pp. 982–992, 2015.

[23] Andrew L. Maas, Quoc V. Le, Tyler M. O'Neil, Oriol Vinyals, Patrick Nguyen, and Andrew Y. Ng, "Recurrent neural networks for noise reduction in robust ASR," in *Proc. Interspeech*, 2012, pp. 22–25.

[24] Jun Du, Qing Wang, Tian Gao, Yong Xu, Lirong Dai, and Chin-Hui Lee, "Robust speech recognition with speech enhanced deep neural networks," in *Proc. Interspeech*, 2014, pp. 616–620.

[25] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Tr. ASLP*, vol. 23, no. 1, pp. 7–19, 2015.

[26] Michael I. Mandel and Nicoleta Roman, "Enforcing consistency in spectral masks using Markov random fields," in *Proc. EUSIPCO*, 2015.

[27] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011, pp. 1–4.

[28] Chao Weng, Dong Yu, Shinji Watanabe, and B.H. Juang, "Recurrent deep neural networks for robust speech recognition," in *Proc. ICASSP*, 2014, pp. 5532–5536.

[29] Vijayaditya Peddinti, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Reverberation robust acoustic modeling using i-vectors with time delay neural networks," in *Proc. Interspeech*, 2015, pp. 2440–2444.

[30] Michael L. Seltzer, Ivan Tashev, and Alex Acero, "Microphone array post-filter using incremental bayes learning to track the spatial distributions of speech and noise," in *Proc. ICASSP*, 2007, vol. 1, pp. 29–32.

[31] Michael I. Mandel and Arun Narayanan, "Analysis-by-synthesis feature estimation for robust automatic speech recognition using spectral masks," in *Proc. ICASSP*, 2014, pp. 2509–2513.

[32] Chanwoo Kim and Richard M. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," in *Proc. ICASSP*, 2012, pp. 4101–4104.

[33] Jitong Chen, Yuxuan Wang, and DeLiang Wang, "A feature study for classification-based speech separation at low signal-to-noise ratios," *IEEE/ACM Tr. ASLP*, vol. 22, no. 12, pp. 1993–2002, 2014.

[34] Birger Kollmeier and René Koch, "Speech enhancement based on physiological and psychoacoustical models of modulation perception and binaural interaction," *J. Acous. Soc. Am.*, vol. 95, no. 3, pp. 1593–1602, 1994.

[35] Hynek Hermansky and Nelson Morgan, "Rasta processing of speech," *IEEE Tr. SAP*, vol. 2, no. 4, pp. 578–589, 1994.

[36] Yuxuan Wang, Kun Han, and DeLiang Wang, "Exploring monaural features for classification-based speech segregation," *IEEE Tr. ASLP*, vol. 21, no. 2, pp. 270–279, 2013.

[37] Arun Narayanan and DeLiang Wang, "Improving robustness of deep neural network acoustic models via speech separation and joint adaptive training," *IEEE/ACM Tr. ASLP*, vol. 23, no. 1, pp. 92–101, 2015.