GENERALIZATION OF TEMPORAL FILTER AND LINEAR TRANSFORMATION FOR ROBUST SPEECH RECOGNITION

Duc Hoang Ha Nguyen[1], Xiong Xiao[2], Eng Siong Chng[1,2], Haizhou Li[1,2,3]

[1]School of Computer Engineering, Nanyang Technological University, Singapore
[2]Temasek Lab@NTU, Nanyang Technological University, Singapore
[3]Institute for Infocomm Research, A*STAR, Singapore.

ng0008ha@e.ntu.edu.sg, xiaoxiong@ntu.edu.sg, aseschng@ntu.edu.sg, hli@i2r.a-star.edu.sg

ABSTRACT

Temporal filtering of feature trajectories and linear transformation of feature vectors are two effective ways to compensate the speech features to achieve robust speech recognition in noisy and reverberant environments. In the previous studies, as the two methods are usually applied in sequence, the interaction between the two methods is not optimized. In this paper, we propose a generalized transform which integrates temporal filter and linear transformation into a single process. The new transform parameters are optimized to minimize an approximated Kullback-Leibler divergence between the distribution of the compensated features and the distribution represented by a clean reference model. The proposed method is evaluated on the Aurora-5 clean condition training task. The experiments show that the generalized transform significantly outperforms the simple cascade of temporal filtering and linear transformation. For example, the word accuracy is improved from 81.55% (cascade) to 83.99% (generalized) and from 72.09% (cascade) to 76.04% (generalized) for office and living room environments, respectively, in speaker based feature adaptation scheme.

Index Terms— Robust speech recognition, temporal filter, linear transformation, reverberant speech recognition, Kullback-Leibler divergence.

1. INTRODUCTION

Robustness to noise and channel effects remains an unsolved problem in automatic speech recognition (ASR). Several factors, such as reverberation, background noise, and transmission channel, introduce distortions to the speech signal and cause a mismatch between the clean trained acoustic model and the distorted test speech signal.

The key to improve the robustness of ASR systems is to reduce the mismatch between the test features and the acoustic model [1]. In the past 3 decades, many techniques have been proposed to deal with robustness issue in ASR, and they can be loosely classified into feature space techniques and model space techniques. Feature space techniques usually compensate [2–4] or normalize [5–9] noisy features to make it closer to the acoustic model. On the other hand, the model space techniques usually either adapts [4, 10, 11] or compensate [12, 13] the acoustic model to represent the noisy test features better. In this paper, we will focus on feature space techniques due to its flexibility to be used in different types of acoustic model.

Among feature space methods, there are two popular types of processings as illustrated in Fig. 1(a). One is the linear transformation of feature vectors, such as the feature space maximum likelihood linear regression (fMLLR, also called constrained MLLR) and its extensions [10, 14–17]. The other is the linear filtering of feature trajectories, such as RASTA [18], ARMA filter [19], data-driven filters [20, 21], temporal structure normalization (TSN) and related methods [22–25], and maximum normalized likelihood linear filter (MNLLF) [26]. From a linear regression point of view, linear transformation uses all dimensions of the current frame to predict the new features that fit the acoustic model under maximum likelihood (ML) criterion. On the other hand, temporal filter uses the context information in neighboring frames to predict the new features. Hence, linear transformation uses inter-dimensional correlation information, while temporal filter uses inter-frame correlation information. In our previous study, we have shown that linear transformation and temporal filter are complementary and applying fMLLR after MNLLF [26] produces better results than the two techniques alone.

Although temporal filter and linear transformation are shown to be complementary, simply applying them in sequence may not produce optimal performance. This is because the cascading of the two techniques does not allow interactions between them, and hence suboptimal performance may be obtained. In this paper, we propose to combine the linear transformation and temporal filter and estimate their parameters using a single objective function. The parameters are estimated to minimize an approximated KL divergence between the distribution of processed features and the distribution of clean training features, represented by a Gaussian mixture model (GMM).

The rest of the paper is organized as follows. The generalization of temporal filter and linear transform is introduced in section 2, followed by experimental study on the Aurora-5 [27] clean condition training task in section 3. Finally, we conclude in section 4.

2. GENERALIZATION OF TEMPORAL FILTER AND LINEAR TRANSFORMATION

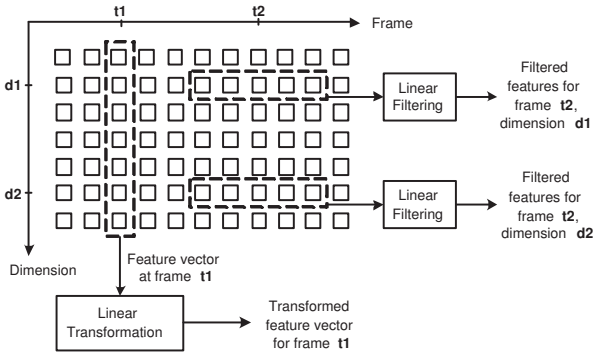
2.1. Generalized Transform

Let's briefly review the processing in temporal filtering and linear transformation. In temporal filtering, if we are using a finite impulse response (FIR) filter such as in TSN [22] and MNLLF [26], we have

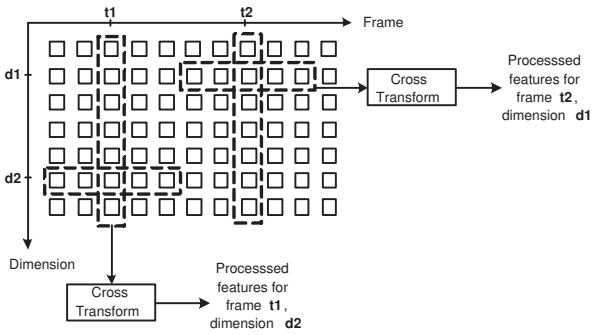
$$y_t^{(d)} = \sum_{\tau=-L}^{L} a_\tau^{(d)} x_{t+\tau}^{(d)} \quad (1)$$

where $x_t^{(d)}$ and $y_t^{(d)}$ represent the d^{th} element of the observed and processed feature vectors at frame t. $a_\tau^{(d)}$ are the filter weights for dimension d with filter length $2L+1$. The temporal filtering can be seen as a linear regression problem in which we use the local feature trajectory centered at $x_t^{(d)}$ to predict the new features $y_t^{(d)}$.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE



(a) Linear transformation and temporal filter



(b) Cross transform

Fig. 1. Cross transform is the combination of linear transformation and temporal filter.

In linear transformation such as fMLLR [10, 14], we have

$$\mathbf{y}_t = \mathbf{B}\mathbf{x}_t + \mathbf{c} \quad (2)$$

$$y_t^{(d)} = \sum_{i=1}^{D} b_i^{(d)} x_t^{(i)} + c^{(d)} \quad (3)$$

where $\mathbf{x}_t = [x_t^{(1)}, ..., x_t^{(D)}]^T$ and $\mathbf{y}_t = [y_t^{(1)}, ..., y_t^{(D)}]^T$ are the observed and processed feature vectors at frame t, respectively. D is the dimensionality of the feature vectors. $b_i^{(d)}$ for $i = 1, ..., D$ is the d^{th} row vector of the transformation matrix $\mathbf{B}$. $c^{(d)}$ is the d^{th} element of the offset vector $\mathbf{c}$. As suggested by its name, fMLLR is also a linear regression problem in which the regressors are the elements of the feature vector at frame t.

As temporal filtering and linear transformation use different information source, namely the inter-frame and inter-dimensional correlation, respectively, they are complementary when used to compensate speech features for robust speech recognition [26]. In [26], a simple cascade of MNLLF filter and fMLLR already shows better results than any of the two methods applied individually. However, we believe that simply applying temporal filtering and linear transformation in sequence do not fully exploit their potential. A better way may be to allow the filter and the transform to be optimized jointly so that interactions between them are accounted for.

To simultaneously apply temporal filter and linear transformation on speech features, we use a generalization of both temporal filter and linear transform as follows:

$$\mathbf{y}_t = \sum_{\tau=-L}^{L} \mathbf{B}_\tau \mathbf{x}_{t+\tau} + \mathbf{c} = \mathbf{W}\tilde{\mathbf{x}} \quad (4)$$

where $\mathbf{B}_\tau$, $\tau = -L, ..., L$ are the transformation matrixes. $\mathbf{W} = [\mathbf{B}_{-L}, ..., \mathbf{B}_L, \mathbf{c}]$ and $\tilde{\mathbf{x}}_t = [\mathbf{x}_{t-L}^T, ..., \mathbf{x}_{t+L}^T, 1]^T$ are the concatenated transformation matrices and inputs, respectively. The transform in (4) maps a block of noisy features to clean features to make use of the temporal (contextual) information. Similar transform has been used in [28], where simultaneous recordings of clean and noisy speech were available during training and the task was to map the noisy features to clean features using a class-based least square regression. Hence, the task in [28] is different from the task in this paper, which is to dynamically adapt the test features without parallel data.

A major problem of the transform in (4) is that there are too many parameters in $\mathbf{W}$ and a lot of test data is required for its reliable estimation. For example, if we set $L = 15$, i.e. use a context of 31 frames, then there are $31D^2 + D$ parameters, which is not feasible to be reliably estimated from a small amount of test data, e.g. one test utterance. Therefore, in this study, we make $\mathbf{W}$ sparse by setting most of its elements to zero. Specifically, to predict the feature at frame t and dimension d, $y_t^{(d)}$, we only use the local feature trajectory and feature vector that contains $x_t^{(d)}$ as shown in Fig. 1(b). The simplified transform is simply the combination of the linear transform in (2) and temporal filter in (1). In this way, long-term distortion can be handled better. As the transform takes place covering a cross in the speech features, we will call it the cross transform.

The cross transform is a special case of the full transform $\mathbf{W}$. Specifically, we restrict $\mathbf{B}_\tau$ to be diagonal matrix if $\tau \neq 0$ and allow $\mathbf{B}_0$ to be full matrix. The number of free parameters in $\mathbf{W}$ is $2LD + D^2 + D$, and the parameters could be robustly estimated from one test utterance if we use regularization and statistics smoothing as described in the following sections.

2.2. Design Criterion

Similar to MNLLF [26], the parameters of the cross transform can be estimated by minimizing an approximated KL divergence between the distribution of processed features, p_y, and the distribution of clean training features, p_Λ. The cost function to be minimized is obtained in a similar way in [26] as follows

$$f(\mathbf{W}) = \frac{\lambda}{T}\sum_{t=1}^{T} \log(p_y(\mathbf{W}\tilde{\mathbf{x}}_t)) - \frac{1}{T}\sum_{t=1}^{T} \log(p_\Lambda(\mathbf{W}\tilde{\mathbf{x}}_t|\Lambda_m)) + \frac{\beta}{2T}||\mathbf{W} - \mathbf{W}_0||_F^2 \quad (5)$$

where T is the number of test frames, the operator $||.||_F^2$ denotes the Frobenius matrix norm. $\mathbf{W}_0$ is the initial weights, in which $\mathbf{c}$ and $\mathbf{B}_\tau$ contain all zero's for $\tau \neq 0$ and $\mathbf{B}_0$ is the identity matrix. With this design, the initial $\mathbf{y}_t = \mathbf{W}_0\tilde{\mathbf{x}}_t = \mathbf{x}_t$. Tunable parameters β and λ are used to control the contributions of the Frobenius norm and data distribution p_y in the cost function, respectively. By minimizing (5), we are trying to increase the likelihood of the processed features on p_Λ which is trained from the features used to train the acoustic model of the speech recognition system. At the same time, the likelihood of the processed features on p_y is kept down to prevent the processed features from having very small variances. Note that the transformation in feature space will change p_y.

In practice, the distribution p_Λ is represented by a GMM, whose parameters $\Lambda = \{c_m, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m | m = 1, ..., M\}$ are estimated together with the acoustic model from the clean training data. As we only have limited test data, a single Gaussian with full covariance is used to represent p_y. The mean and covariance of p_y are estimated as $\boldsymbol{\mu}_\mathbf{y} = \mathbf{W}\boldsymbol{\mu}_{\tilde{\mathbf{x}}}$ and $\boldsymbol{\Sigma}_\mathbf{y} = \mathbf{W}\boldsymbol{\Sigma}_{\tilde{\mathbf{x}}}\mathbf{W}^T$, where $\boldsymbol{\mu}_{\tilde{\mathbf{x}}} = \frac{1}{T}\sum_{t=1}^{T} \tilde{\mathbf{x}}_t$ and

Table 1. Procedures of estimating the cross transform parameters.

Step 1: Set $n = 1$ and $\mathbf{W}_{n-1} = \mathbf{W}_0$
Step 2: Compute statistical in (8), (9) and (10)
Step 3: Estimate $\mathbf{W}_n$ to minimize $Q(\mathbf{W}_{n-1}, \mathbf{W}_n)$ using L-BFGS algorithm [29] with gradient defined in (11).
Step 4: If convergence is met or maximum number of iterations is reached, exit. Otherwise, set $n = n + 1$ and go to Step 2.

$\mathbf{\Sigma}_{\tilde{\mathbf{x}}} = \frac{1}{T}\sum_{t=1}^{T}(\tilde{\mathbf{x}}_t - \boldsymbol{\mu}_{\tilde{\mathbf{x}}})(\tilde{\mathbf{x}}_t - \boldsymbol{\mu}_{\tilde{\mathbf{x}}})^T$ are the sample mean and covariance of $\tilde{\mathbf{x}}$. With the definitions of p_y and p_Λ, (5) can be rewritten as

$$f(\mathbf{W}) = const - \frac{\lambda}{2}\log\det(\mathbf{W}\mathbf{\Sigma}_{\tilde{\mathbf{x}}}\mathbf{W}^T) + \frac{\beta}{2T}||\mathbf{W} - \mathbf{W}_0||_2 - \frac{1}{T}\sum_{t=1}^{T}\log\sum_{m=1}^{M} c_m\mathcal{N}(\mathbf{W}\tilde{\mathbf{x}}_t; \boldsymbol{\mu}_m, \mathbf{\Sigma}_m) \quad (6)$$

where $\det(\cdot)$ denotes the determinant of a matrix. Note that in this work, the full covariance matrix, $\mathbf{\Sigma}_y$, is used to model the processed feature distribution instead of the diagonal one in [26] in order to model the relationship between different features.

2.3. Parameter Estimation

To find the optimal parameters of $\mathbf{W}$, an EM algorithm is used. An auxiliary function to be minimized is derived from (6) as follows

$$\begin{aligned} Q(\mathbf{W}_{n-1}, \mathbf{W}_n) &= -\frac{\lambda}{2}\log\det(\mathbf{W}\mathbf{\Sigma}_{\tilde{\mathbf{x}}}\mathbf{W}^T) + \frac{\beta}{2T}||\mathbf{W} - \mathbf{W}_0||_2 \\ &+ \frac{1}{2T}\sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_{t,m}^{(n-1)}(\mathbf{W}_n\tilde{\mathbf{x}}_t - \boldsymbol{\mu}_m)^T\mathbf{\Sigma}_m^{-1}(\mathbf{W}_n\tilde{\mathbf{x}}_t - \boldsymbol{\mu}_m) \\ &= -\frac{\lambda}{2}\log\det(\mathbf{W}_n\mathbf{\Sigma}_{\tilde{\mathbf{x}}}\mathbf{W}_n^T) + \frac{\beta}{2T}||\mathbf{W}_n - \mathbf{W}_0||_2 \\ &+ \frac{1}{2}\sum_{d=1}^{D}\mathbf{e}_d^T\mathbf{W}_n\mathbf{G}^{(d,n-1)}\mathbf{W}_n^T\mathbf{e}_d - \sum_{d=1}^{D}\mathbf{e}_d^T\mathbf{W}_n\mathbf{p}^{(d,n-1)} \end{aligned} \quad (7)$$

where $\mathbf{e}_d$ is a $D \times 1$ vector whose elements are all zero except that the d^{th} element is 1. The statistics are defined as

$$\gamma_{t,m}^{(n-1)} = \frac{c_m\mathcal{N}(\mathbf{W}_{n-1}\tilde{\mathbf{x}}_t; \boldsymbol{\mu}_m, \mathbf{\Sigma}_m)}{\sum_{i=1}^{M} c_i\mathcal{N}(\mathbf{W}_{n-1}\tilde{\mathbf{x}}_t; \boldsymbol{\mu}_i, \mathbf{\Sigma}_i)} \quad (8)$$

$$\mathbf{G}^{(d,n-1)} = \frac{1}{T}\sum_{t=1}^{T}\sum_{m=1}^{M}\frac{\gamma_{t,m}^{(n-1)}}{{\sigma_m^{(d)}}^2}\tilde{\mathbf{x}}_t\tilde{\mathbf{x}}_t^T \quad (9)$$

$$\mathbf{p}^{(d,n-1)} = \frac{1}{T}\sum_{t=1}^{T}\sum_{m=1}^{M}\frac{\gamma_{t,m}^{(n-1)}}{{\sigma_m^{(d)}}^2}\mu_m^{(d)}\tilde{\mathbf{x}}_t \quad (10)$$

and they are collected using the previous estimate $\mathbf{W}_{n-1}$. Note that deriving (7) requires $\mathbf{\Sigma}_m$ being diagonal. The posterior probabilities of the Gaussians $\gamma_{t,m}^{(n-1)}$ is updated after each time we update $\mathbf{W}$.

The gradient of the objective function w.r.t. the d^{th} row of $\mathbf{W}$ is

$$\frac{\partial Q(\mathbf{W}_{n-1}, \mathbf{W}_n)}{\partial \mathbf{w}^{(d)}} = -\lambda\mathbf{e}_d^T(\mathbf{W}_n\mathbf{\Sigma}_{\tilde{\mathbf{x}}}\mathbf{W}_n^T)^{-1}\mathbf{W}_n\mathbf{\Sigma}_{\tilde{\mathbf{x}}}^T + \mathbf{w}^{(d)}\mathbf{G}^{(d,n-1)} - {\mathbf{p}^{(d,n-1)}}^T + \frac{\beta}{T}(\mathbf{w}^{(d)} - \mathbf{e}_d^T\mathbf{W}_0) \quad (11)$$

The EM algorithm is summarized in Table 1. The maximum number of EM iterations is set to 10. In the maximization step of each EM iteration, we use L-BFGS [29] to minimize the auxiliary function.

Table 2. Recognition accuracy (%) achieved by speaker-based processing on artificial reverberant and noisy data. "MNLLF+fMLLR" represents the cascade of the two methods. $\lambda = 0.8$ is used. "Office" is corrupted by reverberation only, while the 4 rows below "Office" are corrupted by both reverberation and additive noise. The same rule also applied to "Living".

SNR	Baseline	MNLLF	fMLLR	MNLLF +fMLLR	Cross transform
Clean	99.38	99.35	**99.57**	**99.57**	99.51
Office	94.26	97.05	98.03	**98.54**	98.41
15dB	83.99	90.49	92.88	95.11	**95.90**
10dB	75.61	83.00	85.20	88.85	**91.08**
5dB	61.65	68.79	69.94	75.05	**79.00**
0dB	41.69	46.81	44.73	50.22	**55.54**
Avg	71.44	77.23	78.16	81.55	**83.99**
Living	83.07	91.40	92.07	95.37	**95.46**
15dB	68.62	81.01	79.13	87.38	**91.11**
10dB	60.37	72.07	69.10	77.95	**83.32**
5dB	48.77	57.25	52.99	61.35	**68.00**
0dB	33.20	37.95	33.21	38.40	**42.33**
Avg	58.81	67.94	65.30	72.09	**76.04**

2.4. Statistics Smoothing

The estimation of $\mathbf{W}$ relies on 3 statistics: $\mathbf{\Sigma}_{\tilde{\mathbf{x}}}$, $\mathbf{G}^{(d)}$ and $\mathbf{p}^{(d)}$. The size of these statistics is quite large (e.g. with $L = 16$ and $D = 39$, $\mathbf{\Sigma}_{\tilde{x}}$ is 1288×1288), reliable estimation is difficult with limited test data. To address this issue, we smooth the statistics collected from current test utterance by interpolating them with the statistics collected from clean training data as follows

$$\hat{\mathbf{\Sigma}}_{\tilde{x}} = \frac{T}{T_0 + T}\mathbf{\Sigma}_{\tilde{x}} + \frac{T_0}{T_0 + T}\mathbf{\Sigma}_{\tilde{x}}^{\text{clean}} \quad (12)$$

$$\hat{\mathbf{G}}^{(d)} = \frac{T}{T_0 + T}\mathbf{G}^{(d)} + \frac{T_0}{T_0 + T}\mathbf{G}^{(d)\text{clean}} \quad (13)$$

$$\hat{\mathbf{p}}^{(d)} = \frac{T}{T_0 + T}\mathbf{p}^{(d)} + \frac{T_0}{T_0 + T}\mathbf{p}^{(d)\text{clean}} \quad (14)$$

where $\mathbf{\Sigma}_{\tilde{x}}^{\text{clean}}$, $\mathbf{G}^{(d)\text{clean}}$ and $\mathbf{p}^{(d)\text{clean}}$ are collected from clean training data. T_0 is used to control the level of smoothing. The smoothed versions of the statistics are used to compute the gradient in (11) instead of the ones computed from test data.

3. EXPERIMENTS

We evaluate the proposed cross transform on the Aurora-5 clean-condition training task [27]. The acoustic model was trained using the standard script [27]. There are totally 11 test cases, including 1 clean case, 5 cases in living room environment, and 5 in office environment. For details of these test cases and model training, please refer to [27]. Speech features are 39D MFCC features, including c0-c12, and their first and second derivatives. Two feature preprocessing methods are applied in cascade to each utterance, i.e. mean and variance normalization (MVN) [6] and TSN [22].

Two types of experiments, i.e. utterance-based processing and speaker-based processing, were conducted. In utterance-based processing, cross transforms are estimated for each test utterance and

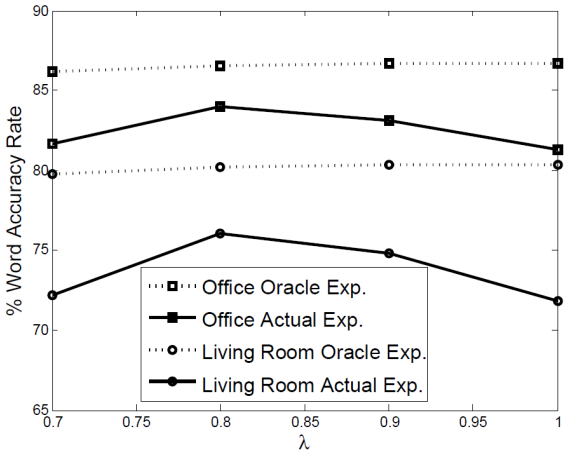


Fig. 2. The average word recognition accuracy rate of the oracle and actual experiments with various values of λ.

statistic smoothing is applied. In speaker-based processing, cross transforms are estimated for each test speaker and statistic smoothing is not used.

For both MNLLF and cross transform, the reference GMM p_λ is obtained by pooling the Gaussians in the clean acoustic model. For fMLLR, a 2-pass decoding scheme and the clean acoustic model are used to estimate the transforms. For both MNLLF and cross transform, we use a context size of 33 frames, i.e. $L = 16$. The variable λ and β are empirically set to 1 if not otherwise stated. In utterance mode, the statistics smoothing parameters T_0 is set to 100. No offset is used in the cross transform.

3.1. Cross Transform by Speaker

Results for speaker-based processing are shown in Table 2. It is observed that cross transform outperforms the cascading of temporal filter MNLLF and linear transformation fMLLR significantly, except for clean case and Office room no additive noise case. The number of free parameters in cross transform is $39 \times 32 + 39 \times 39 = 2769$, which is slightly less than the combination of MNLLF $39 \times 33 = 1287$ and fMLLR $39 \times 39 + 39 = 1560$. The number of utterances for each test speaker is about 77 utterances, which represents a duration of 144 seconds or 14,400 frames of feature vectors. Hence, the amount of data is enough for reliable estimation of all involved filters and transformations. The results proved that by using cross transform to combine both temporal filtering and linear transformation and estimate the parameters jointly, better performance can be obtained than simple cascading.

We also investigate the effect of the data distribution term in (6). The average recognition accuracies on Office and Living room test cases are plotted in Fig. 2 for different λ. For comparison, we also plotted the performance of a "oracle test", in which the Gaussian posteriors $\gamma_{t,m}$ are obtained by using the underlying clean features of the noisy utterances. From Fig. 2, it is observed that when oracle Gaussian posterior are used, the performance of cross transform does not change much with λ. However, if the Gaussian posteriors are estimated from noisy observations, the performance of cross transform depends on careful selection of λ and is the best when $\lambda = 0.8$ for this particular test. Fig. 2 also shows the limitation of the linear cross transform. For example, even with "oracle" Gaussian posteriors, the average accuracy on Living room is only around 80%. This is properly due to the fact that the linear cross transform is not able to deal with some nonlinear distortions in the cepstral domain.

Table 3. Recognition accuracy achieved by utterance based processing on artificial noisy data. fMLLR uses diagonal transforms. $\lambda = 1.0$ is used.

SNR	Baseline	MNLLF	fMLLR	MNLLF +fMLLR	Cross transform
Clean	**99.38**	99.22	99.37	99.27	99.08
Office	94.26	96.66	95.32	**96.83**	96.34
15dB	83.99	89.82	85.64	90.05	**91.16**
10dB	75.61	82.72	76.76	82.78	**85.36**
5dB	61.65	69.61	61.88	69.30	**73.65**
0dB	41.69	48.53	40.27	47.46	**52.63**
Avg	71.44	77.47	71.97	77.28	**79.83**
Living	83.07	89.93	85.72	**90.58**	89.60
15dB	68.62	79.58	70.62	80.02	**81.01**
10dB	60.37	71.14	61.37	71.26	**73.55**
5dB	48.77	57.72	48.53	57.31	**60.85**
0dB	33.20	39.16	31.83	37.97	**42.15**
Avg	58.81	67.51	59.61	67.43	**69.43**

3.2. Cross-Transform by Utterance

In the previous subsection, we have demonstrated the potential of cross transform when there are sufficient test data to estimate its parameters. In this subsection, we investigate how cross transform works with limited test data.

The performance of cross transform and other methods are shown in Table 3. Interestingly, we observe similar results as those in speaker based processing in Table 2. Specifically, cross transform outperforms MNLLF, fMLLR, and their combination, in all test cases, except for the clean and reverberant test cases without additive noise. In utterance mode, the number of free parameters in cross transform is the same as that in speaker mode, i.e. 2769 free parameters. This is also true for MNLLF that has 1287 parameters. For fMLLR, only diagonal transform with $39 + 39 = 78$ parameters is used as the average utterance length is only 2s and not enough to estimate even block diagonal transforms. From the results, we conclude that cross transform is still preferable especially at low SNR levels. The gain of cross transform over other methods may be due to two reasons: 1) the joint optimization of temporal filter and linear transformation; 2) the smoothing of statistics.

4. CONCLUSIONS

In this paper, we proposed a generalized transform called cross transform that combines temporal filter and linear transformation for robust speech recognition. The cross transform utilizes both inter-frame and inter-dimensional information of speech features for feature processing. The performances of the cross transform in both utterance and speaker-based processing significantly outperforms the cascade of MNLLF filter [26] and fMLLR transform [10]. In future, we will investigate nonlinear forms of cross transform that may be better at handling nonlinear distortions in the cepstral domain.

5. REFERENCES

[1] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, April 2014.

[2] J. Droppo, L. Deng, and A. Acero, "Evaluation of the SPLICE algorithm on the Aurora2 database," in *Proc. InterSpeech '01*, Aalborg, Denmark, Sept. 2001, pp. 217–220.

[3] J. Li, M. L. Seltzer, and Y. Gong, "Improvements to vts feature enhancement," in *Proc. ICASSP '12*, Vancouver, Canada, May 2012, pp. 4677–4680.

[4] A. Sankar and C.-H. Lee, "A maximum-likelihood approach to stochastic matching for robust speech recognition," *IEEE Trans. Speech and Audio Processing*, vol. 4, no. 3, pp. 190–202, 1996.

[5] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 29, no. 2, pp. 254–272, 1981.

[6] O. Viikki and K. Laurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition," *Speech Communication*, vol. 25, pp. 133–147, 1998.

[7] Á. de la Torre, A. M. Peinado, J. C. Segura, J. L. Pérez-Córdoba, M. C. Benítez, and A. J. Rubio, "Histogram equalization of speech representation for robust speech recognition," *IEEE Trans. Speech and Audio Processing*, vol. 13, no. 3, pp. 355–366, 2005.

[8] X. Xiao, J. Li, E. S. Chng, and H. Li, "Maximum likelihood adaptation of histogram equalization with constraint for robust speech recognition," in *Proc. ICASSP '11*, Prague, Czech, May 2011, pp. 5480–5483.

[9] X. Xiao, J. Li, H. Li, and E. S. Chng, "Feature normalization using structured full transforms for robust speech recognition," in *Proc. InterSpeech '11*, Florence, Italy, Aug. 2011, pp. 693–696.

[10] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[11] J. L. Gauvain and C. H. Lee, "Maximum *a posterirori* estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Trans. Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, Apr. 1994.

[12] M. J. F. Gales and S. J. Young, "Cepstral parameter compensation for HMM recognition," *Speech Communication*, vol. 12, no. 3, pp. 231–239, Jul. 1993.

[13] J. Li, L. Deng, D. Yu, Y. Gong, and A. Acero, "HMM adaptation using a phase-sensitive acoustic distortion model for environment-robust speech recognition," in *Proc. ICASSP '08*, Las Vegas, Nevada, USA, Apr. 2008, pp. 4069–4072.

[14] V. Digalakis, D. Rtischev, and L. G. Neumeyer, "Speaker adaptation using constrained estimation of Gaussian mixtures," *IEEE Trans. Speech and Audio Processing*, vol. 3, no. 5, pp. 357–366, 1995.

[15] Y. Li, H. Erdogan, Y. Gao, and E. Marcheret, "Incremental on-line feature space MLLR adaptation for telephony speech recognition," in *Proc. ICSLP '02*, Denver, USA, Sept. 2002, pp. 1417–1420.

[16] G. Stemmer, F. Brugnara, and D. Giuliani, "Adaptive training using simple target models," in *Proc. ICASSP '05*, Philadelphia, USA, Mar. 2005, vol. I, pp. 997–1000.

[17] X. Lei, J. Hamaker, and X. He, "Robust feature space adaptation for telephony speech recognition.," in *Proc. INTERSPEECH '06*, 2006.

[18] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Trans. Speech and Audio Processing*, vol. 2, no. 4, pp. 578–589, 1994.

[19] C.-P. Chen and J. A. Bilmes, "MVA processing of speech features," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 257–270, 2007.

[20] J.-W. Hung and L.-S. Lee, "Optimization of temporal filters for constructing robust features in speech recognition," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 14, no. 3, pp. 808–832, 2006.

[21] J.-W. Hung and W.-Y. Tsai, "Constructing modulation frequency domain-based features for robust speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 3, pp. 563–577, March 2008.

[22] X. Xiao, E. S. Chng, and H. Li, "Normalization of the speech modulation spectra for robust speech recognition," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1662–1674, Nov. 2008.

[23] X. Xiao, E. S. Chng, and H. Li, "Temporal structure normalization of speech feature for robust speech recognition," *IEEE Signal Processing letters*, vol. 14, no. 7, pp. 500–503, 2007.

[24] X. Xiao, E. S. Chng, and H. Li, "Joint spectral and temporal normalization of features for robust recognition of noisy and reverberated speech," in *Proc. ICASSP '12*, Kyoto, Japan, Apr. 2012, pp. 4325–4328.

[25] X. Lu, M. Unoki, and S. Nakamura, "Sub-band temporal modulation envelopes and their normalization for automatic speech recognition in reverberant environments," *Computer Speech and Language*, vol. 25, no. 3, pp. 571–584, 2011.

[26] X. Xiao, E. S. Chng, and H. Li, "Temporal filter design by minimum KL divergence criterion for robust speech recognition," in *Proc. ICASSP '13*, Vancouver, Canada, May.

[27] H. G. Hirsch, "Aurora-5 experimental framework for the performance evaluation of speech recognition in case of a hands-free speech input in noisy environments," Tech. Rep., 2007.

[28] L. Neumeyer and M. Weintraub, "Probabilistic optimum filtering for robust speech recognition," in *Proc. ICASSP '94*, Adelaide, Austrailia, Apr. 1994, pp. 417–420.

[29] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Program.*, vol. 45, no. 3, pp. 503–528, Dec. 1989.