MFCC ENHANCEMENT USING JOINT CORRUPTED AND NOISE FEATURE SPACE FOR HIGHLY NON-STATIONARY NOISE ENVIRONMENTS

Masayuki Suzuki[1], *Takuya Yoshioka*[2], *Shinji Watanabe*[2], *Nobuaki Minematsu*[1], *Keikichi Hirose*[1]

[1]The University of Tokyo, Japan
[2]NTT Communication Science Laboratories, NTT Corporation, Japan
{suzuki,mine,hirose}@gavo.t.u-tokyo.ac.jp
yoshioka.takuya@lab.ntt.co.jp watanabe@merl.com

ABSTRACT

One of the most effective approaches to noise robust speech recognition is to remove the noise effect directly from corrupted MFCC vectors. However, VTS enhancement, which is a typical method for performing MFCC enhancement, provides limited improvement when the noise is highly non-stationary. This is because the VTS enhancement method cannot use a time-varying noise model to keep the computational cost at an acceptable level. This paper proposes a method that can enhance MFCC vectors and their dynamic parameters by using noise estimates that change on a frame-by-frame basis at a practical computational cost. The proposed method employs stereo data-based feature mapping like the well known SPLICE algorithm. The novelty of the proposed method lies in that it uses the joint space spanned by a concatenated vector of corrupted and noise features. It is also proposed to use linear discriminant analysis to effectively reduce the dimensionality of the joint space. The proposed method achieves 19.1% and 8.3% relative error reduction from the SPLICE and noise-mean normalized SPLICE algorithms, respectively.

Index Terms— Noise robust ASR, non-stationary noise, SPLICE

1. INTRODUCTION

Automatic speech recognition performance drops sharply in noisy environments due to a mismatch between the acoustic model of a recognizer and input features corrupted by acoustic environmental noise. A variety of solutions to this problem have been proposed, including feature enhancement and acoustic model adaptation.

One common drawback of many existing noise-robust algorithms is their inefficiency in highly non-stationary noise environments. Many algorithms explicitly or implicitly employ a fixed noise model by assuming stationary or slowly changing noise environments. For example, vector Taylor series (VTS) approximation-based algorithms, which can be used for both feature enhancement [1] and model adaptation [2, 3], usually employ a single Gaussian noise model so that those algorithms can be carried out at a practical computational cost. However, the fixed noise model assumption makes those algorithms ineffective in highly non-stationary noise environments.

One way of exploiting temporally changing noise estimates is to operate on log mel spectra instead of mel frequency ceptral coefficients (MFCCs). For instance, the algorithms proposed in [4, 5] attempt to estimate clean log mel spectra given their noisy versions. However, the speech recognition accuracy obtained with log mel spectrum enhancement is generally significantly lower than that obtained with MFCC enhancement.

With this as a background, this paper proposes a method that can directly enhance corrupted MFCCs and their dynamic parameters by using noise estimates that change on a frame-by-frame basis at a low computational cost. The proposed method is an extension of the well-known SPLICE algorithm [6]. The original SPLICE algorithm divides a corrupted feature space into a number of disjoint regions by using a Gaussian mixture model (GMM) of corrupted features. For each region, a linear mapping function from a corrupted feature to a clean feature is estimated in advance. Given a vector of corrupted MFCCs, SPLICE first finds the region that the observed MFCC vector belongs to (space division step) and then enhances the corrupted MFCC vector by using the mapping function associated with the selected region (feature mapping step). Unlike the original SPLICE algorithm, the proposed method uses a joint vector comprising corrupted and noise MFCCs to perform both space division and feature mapping. In addition, we propose to perform dimensionality reduction of the space where the joint vector is distributed based on linear discriminant analysis (LDA). The proposed method outperformed both noise mean normalized- (NMN-) SPLICE and the VTS feature enhancement algorithm on the Aurora2 task.

The rest of this paper is organized as follows. In Section 2, we review SPLICE. In Section 3, we explain our proposed method. In Section 4, the proposed method is evaluated on the Aurora2 task. In Section 5, our conclusion is presented.

2. REVIEW OF SPLICE

SPLICE is a method for estimating a clean MFCC vector $\boldsymbol{x}$ from its noisy version $\boldsymbol{y}$, where both $\boldsymbol{x}$ and $\boldsymbol{y}$ are composed of static MFCCs and their dynamic parameters. We denote the dimension of $\boldsymbol{x}$ and $\boldsymbol{y}$ as N.

SPLICE employs piecewise linear transformation to model a mapping from a corrupted feature to a clean feature. With the original version of SPLICE, we obtain an estimate $\hat{\boldsymbol{x}}$ of the clean feature $\boldsymbol{x}$ according to the following formula:

$$\hat{\boldsymbol{x}} = \sum_{k=1}^{K} p(k|\boldsymbol{y}) \boldsymbol{A}_k \boldsymbol{y}', \tag{1}$$

where $\boldsymbol{y}'$ is an augmented feature vector given by $\boldsymbol{y}' = [1\ \boldsymbol{y}^T]^T$. $\boldsymbol{A}_k$ is an $N \times (N+1)$ matrix trained in advance. $p(k|\boldsymbol{y})$ is calculated by using a GMM of corrupted features. K is the size of the GMM. This process performed by SPLICE is illustrated in Fig. 1.

Shinji Watanabe is now with Mitsubishi Electric Research Laboratories (MERL)

978-1-4673-0046-9/12/$26.00 ©2012 IEEE

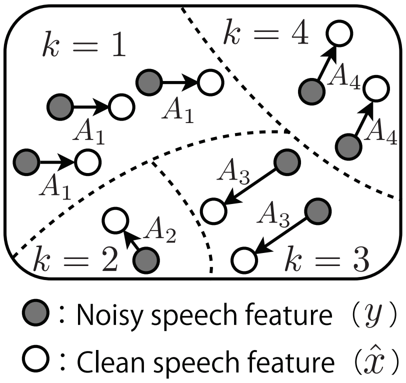
$k=1$
$k=4$
A_1
A_1
A_1
A_4
A_4
A_3
A_3
A_2
$k=2$
$k=3$
● : Noisy speech feature (y)
○ : Clean speech feature ($\hat{x}$)

Fig. 1. Illustration of SPLICE transformation from noisy speech features $\boldsymbol{y}$ to estimated clean speech features $\hat{\boldsymbol{x}}$. k is a region indicator.

Note that calculation of $p(k|\boldsymbol{y})$ corresponds to the space division step of SPLICE while calculation of $\boldsymbol{A}_k\boldsymbol{y}'$ is the feature mapping step. This can be easily understood if we consider forcing the largest posterior probability to be one and the other posterior probabilities to be zero.

2.1. Training of SPLICE parameters

Before using SPLICE, we need to train the parameters of the corrupted feature GMM $p(\boldsymbol{y}) = \sum_{k=1}^{K} p(k)p(\boldsymbol{y}|k)$ and the set of linear transformation matrices $\{\boldsymbol{A}_k\}_{k=1}^{K}$. For this purpose, we use time synchronized feature vector (static and dynamic MFCC) sequences of clean speech $\boldsymbol{X} = [\boldsymbol{x}_1\boldsymbol{x}_2\cdots\boldsymbol{x}_I]$ and noisy speech $\boldsymbol{Y} = [\boldsymbol{y}_1\boldsymbol{y}_2\cdots\boldsymbol{y}_I]$. Here, I is the total number of feature vectors contained in the training data set. This kind of stereo data can be prepared by recording clean speech and noise separately and mixing them artificially on a computer.

First, we train a K-component GMM of corrupted features by using $\boldsymbol{Y}$. We estimate the weight $p(k)$, mean vector $\boldsymbol{\mu}_k^{\boldsymbol{y}}$, and covariance matrix $\boldsymbol{\Sigma}_k^{\boldsymbol{y}}$ for each k value. Here, we assume $\boldsymbol{\Sigma}_k^{\boldsymbol{y}}$ to be diagonal. Using these parameters, we can calculate $p(k|\boldsymbol{y})$ as

$$p(k|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|k)p(k)}{\sum_{k=1}^{K} p(\boldsymbol{y}|k)p(k)}, \tag{2}$$

where $p(\boldsymbol{y}|k) = \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}_k^{\boldsymbol{y}}, \boldsymbol{\Sigma}_k^{\boldsymbol{y}})$.

Next, we estimate the $N \times (N+1)$ matrix $\boldsymbol{A}_k$ for each k. In this paper, we use a weighted minimum mean square error criterion to estimate the value of $\boldsymbol{A_k}$. Specifically, the optimal linear transformation matrix is given by

$$\boldsymbol{A}_k = \underset{\boldsymbol{A}_k}{\operatorname{argmin}} \sum_{i=1}^{I} p(k|\boldsymbol{y}_i)||\boldsymbol{x}_i - \boldsymbol{A}_k\boldsymbol{y}_i'||^2, \tag{3}$$

where $\boldsymbol{y}_i'$ is an augmented feature vector $[1\ \boldsymbol{y}_i^T]^T$. We can analytically obtain the optimal linear transformation matrix as

$$\boldsymbol{A}_k = \boldsymbol{X}\boldsymbol{P}\boldsymbol{Y}'^T(\boldsymbol{Y}'\boldsymbol{P}\boldsymbol{Y}'^T)^{-1}. \tag{4}$$

$\boldsymbol{Y}'$ is the sequence of augmented feature vectors given by $[1\ \boldsymbol{y}_i^T]^T$. $\boldsymbol{P}$ is a diagonal matrix, which has $[p(k|\boldsymbol{y}_1), p(k|\boldsymbol{y}_2), \cdots p(k|\boldsymbol{y}_I)]$ as its diagonal elements [7].

2.2. Noise mean normalization

The problem of SPLICE is that it provides limited improvement when training and test environments are different or the training environment is non-stationary. This is because SPLICE uses only corrupted features to perform space division in spite of the fact that the feature mapping function should essentially be selected depending on a signal-to-noise ratio. Performing space division by using only the corrupted features mixes up the variabilities of speech and noise, which precludes selection of an appropriate feature mapping function.

To mitigate this degradation, a heuristic solution, called NMN-SPLICE, was proposed [6]. NMN-SPLICE uses the following formula to obtain a clean feature estimate $\hat{\boldsymbol{x}}$:

$$\hat{\boldsymbol{x}} = \sum_{k=1}^{K} p(k|\boldsymbol{y} - \hat{\boldsymbol{n}})\boldsymbol{A}_k(\boldsymbol{y} - \hat{\boldsymbol{n}})', \tag{5}$$

where $\hat{\boldsymbol{n}}$ is an estimate of a noise feature. The difference between the original and NMN-SPLICE algorithms is that NMN-SPLICE uses $\boldsymbol{y} - \hat{\boldsymbol{n}}$ instead of $\boldsymbol{y}$ for space division and feature mapping. The use of $\boldsymbol{y} - \hat{\boldsymbol{n}}$ is known to remove the effect of noise variability to some extent from the variability of corrupted features, thereby leading to a higher speech recognition accuracy.

However, it is a bit unclear why the use of $\boldsymbol{y} - \hat{\boldsymbol{n}}$ can provide the improved recognition accuracy. Perhaps there can be a vector that is calculated based on $\boldsymbol{y}$ and $\hat{\boldsymbol{n}}$ and leads to an even higher recognition accuracy. With this motivation, we seek an answer to the following question: "Is there an alternative vector that leads to a higher recognition accuracy than $\boldsymbol{y} - \hat{\boldsymbol{n}}$?" The proposed method is derived in an effort to answer this question.

3. PROPOSED METHOD

There are two key ideas behind the proposed method: (1) the use of the joint space of corrupted and noise features and (2) LDA-based dimensionality reduction for this joint space. Before describing the algorithm of the proposed method, we explain these two ideas by taking a close look at the original and NMN-SPLICE algorithms.

The first idea is to modify the original SPLICE algorithm so that it uses the joint vector $[\boldsymbol{y}^T\hat{\boldsymbol{n}}^T]^T$, comprising a corrupted feature vector $\boldsymbol{y}$ and a noise feature vector estimate $\hat{\boldsymbol{n}}$, instead of the corrupted feature vector $\boldsymbol{y}$ alone. The use of the joint vector means that we take both $\boldsymbol{y}$ and $\hat{\boldsymbol{n}}$ into account for space division and feature mapping. Therefore, the joint vector approach basically enables us to discriminate the variability of noise features from that of clean features.

However, we cannot gain significant recognition accuracy improvement when we simply replace the corrupted feature vector by the joint feature vector in the SPLICE algorithm due to the high correlation between $\boldsymbol{y}$ and $\hat{\boldsymbol{n}}$ in the low SNR region, where $\boldsymbol{y} \simeq \boldsymbol{n}$. Therefore, it is necessary to reduce the dimension of the joint feature space.

With the above discussion in mind, let us look into NMN-SPLICE, which uses the difference, $\boldsymbol{y} - \hat{\boldsymbol{n}}$, between the corrupted and noise features. Since $\boldsymbol{y} - \hat{\boldsymbol{n}}$ is obtained by multiplying $[\boldsymbol{y}^T\hat{\boldsymbol{n}}^T]^T$ by $[\boldsymbol{I}, -\boldsymbol{I}]$ from the left, we find that the space where $\boldsymbol{y} - \hat{\boldsymbol{n}}$ is distributed is the complementary orthogonal subspace of the subspace satisfying $\boldsymbol{y} = \hat{\boldsymbol{n}}$. Therefore, the question we raised at the end of Section 2 is now rephrased as follows: "Is there an alternative subspace that provides a higher recognition accuracy than the subspace of $\boldsymbol{y} - \hat{\boldsymbol{n}}$?"

At this point, the second idea, or LDA-based dimensionality reduction, comes into play. We attempt to find an appropriate dimensionality reduction matrix $\boldsymbol{L}$, which is applied to the joint vector $[\boldsymbol{y}^T\hat{\boldsymbol{n}}^T]^T$, based on a set of training data $\{[\boldsymbol{y}_i^T\hat{\boldsymbol{n}}_i^T]^T\}_{i=1,\cdots,I}$. For this purpose, we need to define the cost function for optimizing matrix $\boldsymbol{L}$. In this paper, we propose to employ LDA, using an index of a clean feature GMM as a class label. Specifically, we first obtain a GMM of clean features $p(\boldsymbol{x}) = p(m)p(\boldsymbol{x}|m)$ and calculate $p(m|\boldsymbol{x}_i)$ for each clean feature vector $\boldsymbol{x}_i$. In this way, we obtain a probabilistic class label $p(m|\boldsymbol{x}_i)$ for each joint feature vector $[\boldsymbol{y}_i^T\hat{\boldsymbol{n}}_i^T]^T$ contained in the training data set. Then, given the training data set, we calculate the dimensionality reduction matrix $\boldsymbol{L}$ based on LDA. After obtaining the dimensionality reduction matrix $\boldsymbol{L}$, we use $\boldsymbol{v} = \boldsymbol{L}[\boldsymbol{y}^T\hat{\boldsymbol{n}}^T]^T$ in place of $\boldsymbol{y}$ in the SPLICE algorithm.

The important idea presented above is the use of a clean feature GMM index as a class label. This idea is quite natural when we recall that the aim of feature enhancement is to obtain estimates of clean features. We see that the proposed method outperformed NMN SPLICE in Section 4, experimentally. Therefore, we can positively answer the question raised in Section 2.

In summary, the proposed method uses the following formula to obtain a clean feature estimate $\hat{\boldsymbol{x}}$:

$$\hat{\boldsymbol{x}} = \sum_{k=1}^{K} p(k|\boldsymbol{v})\boldsymbol{A}_k[1\ \boldsymbol{y}^T\hat{\boldsymbol{n}}^T]^T. \tag{6}$$

Note that, in (6), we use the joint vector $[\boldsymbol{y}^T\hat{\boldsymbol{n}}^T]^T$ instead of $\boldsymbol{v}$ for the feature mapping step because it led to slightly better speech recognition performance in our preliminary experiments. In Section 3.1, we elaborate on the space division step, i.e., calculation of $p(k|\boldsymbol{v})$. Then in Section 3.2, we provide a detailed description of the feature mapping step, i.e., calculation of $\boldsymbol{A}_k[1\ \boldsymbol{y}^T\hat{\boldsymbol{n}}^T]^T$.

As regards noise estimation, our current implementation uses the method proposed in [8] to estimate noise features. This method estimates log mel frequency spectra of noise and was shown to provide good noise estimates even in highly non-stationary noise environments. We use static and dynamic MFCCs derived from the estimated noise log mel frequency spectra.

3.1. Space division step

We propose to use LDA to perform dimensionality reduction of the space of $[\boldsymbol{y}^T\hat{\boldsymbol{n}}^T]^T$ by using $p(m|\boldsymbol{x})$ as a probabilistic class label. We note that the within-class covariance matrix for multi-class LDA is defined as the sum of covariance matrices weighted by $p(m|\boldsymbol{x})$ because $p(m|\boldsymbol{x})$ is not a binary value.

As a training data set for the proposed method, we need time synchronized feature vector (static and dynamic MFCCs) sequences of clean speech $\boldsymbol{X} = [\boldsymbol{x}_1\boldsymbol{x}_2\cdots\boldsymbol{x}_I]$, noisy speech $\boldsymbol{Y} = [\boldsymbol{y}_1\boldsymbol{y}_2\cdots\boldsymbol{y}_I]$, and estimated noise $\hat{\boldsymbol{N}} = [\hat{\boldsymbol{n}}_1\hat{\boldsymbol{n}}_2\cdots\hat{\boldsymbol{n}}_I]$.

First, we train an M-component GMM of clean features using $\boldsymbol{X}$. We estimate the weight $p(m)$, mean vector $\boldsymbol{\mu}_m^{\boldsymbol{x}}$, and covariance matrix $\boldsymbol{\Sigma}_m^{\boldsymbol{x}}$ of the m-th mixture component. Using these parameters, we can calculate $p(m|\boldsymbol{x})$ as

$$p(m|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|m)p(m)}{\sum_{m=1}^{M} p(\boldsymbol{x}|m)p(m)}, \tag{7}$$

where $p(\boldsymbol{x}|m) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_m^{\boldsymbol{x}}, \boldsymbol{\Sigma}_m^{\boldsymbol{x}})$.

Next, we estimate LDA-based dimensionality reduction matrix $\boldsymbol{L}$ by using a set of the joint vectors $\{[\boldsymbol{y}_i^T\hat{\boldsymbol{n}}_i^T]\}_{i=1,\ldots I}^T$ and its corresponding labels $\{p(m|\boldsymbol{x}_i)\}_{i=1,\ldots,I}$ calculated by the clean feature GMM.

Then, we train a K-component GMM of feature vectors, $\boldsymbol{v} = \boldsymbol{L}[\boldsymbol{y}^T\hat{\boldsymbol{n}}^T]^T$, obtained after performing the LDA-based dimensionality reduction. We estimate the weight $p(k)$, mean vector $\boldsymbol{\mu}_k^{\boldsymbol{v}}$, and covariance matrix $\boldsymbol{\Sigma}_k^{\boldsymbol{v}}$ of the k-th mixture component. Using these parameters, we can calculate $p(k|\boldsymbol{v})$ as

$$p(k|\boldsymbol{v}) = \frac{p(\boldsymbol{v}|k)p(k)}{\sum_{k=1}^{K} p(\boldsymbol{v}|k)p(k)}, \tag{8}$$

where $p(\boldsymbol{v}|k) = \mathcal{N}(\boldsymbol{v}; \boldsymbol{\mu}_k^{\boldsymbol{v}}, \boldsymbol{\Sigma}_k^{\boldsymbol{v}})$.

It should be noted that when $\boldsymbol{L} = [\boldsymbol{I}, -\boldsymbol{I}]$, $\boldsymbol{v}$ becomes $\boldsymbol{y} - \hat{\boldsymbol{n}}$ and therefore the proposed space division step reduces to that of NMN-SPLICE.

3.2. Feature mapping step

The feature mapping step of the proposed method is obtained by substituting $[1\ \boldsymbol{y}^T\hat{\boldsymbol{n}}^T]^T$ for $\boldsymbol{y}' = [1\ \boldsymbol{y}^T]^T$ in SPLICE. Accordingly, the size of matrix $\boldsymbol{A}_k$ becomes $N \times (2N+1)$.

We estimate $\boldsymbol{A}_k$ by

$$\boldsymbol{A}_k = \underset{\boldsymbol{A}_k}{\operatorname{argmin}} \sum_{i=1}^{I} p(k|\boldsymbol{v}_i)||\boldsymbol{x}_i - \boldsymbol{A}_k[1\ \boldsymbol{y}_i^T\hat{\boldsymbol{n}}_i^T]^T||^2. \tag{9}$$

We can analytically obtain the optimal linear transformation matrix as

$$\boldsymbol{A}_k = \boldsymbol{X}\boldsymbol{P}[\boldsymbol{Y}'^T\boldsymbol{N}^T]([\boldsymbol{Y}'^T\boldsymbol{N}^T]^T\boldsymbol{P}[\boldsymbol{Y}'^T\boldsymbol{N}^T])^{-1}. \tag{10}$$

$\boldsymbol{Y'}$ is the sequence of augmented feature vectors given by $[1\ \boldsymbol{y}_i^T]^T$. $\boldsymbol{P}$ is a diagonal matrix, which has $[p(k|\boldsymbol{v}_1), p(k|\boldsymbol{v}_2), \cdots\ p(k|\boldsymbol{v}_I]$ as its diagonal elements.

Note that our proposed feature mapping step is a generalization of that of NMN-SPLICE. To see this, let us denote the submatrices of $\boldsymbol{A}_k$ corresponding to $\boldsymbol{y}$ and $\hat{\boldsymbol{n}}$ by $\boldsymbol{A}_k^{\boldsymbol{y}}$ and $\boldsymbol{A}_k^{\hat{\boldsymbol{n}}}$, respectively. Then, we can see that forcing $\boldsymbol{A}_k^{\boldsymbol{y}}$ to be equal to $-\boldsymbol{A}_k^{\hat{\boldsymbol{n}}}$ makes the proposed feature mapping step equivalent to the NMN-SPLICE's feature mapping step. In the same way, we can easily find that using $[1\ \boldsymbol{v}^T]^T$ for feature mapping is a spacial case of the proposed feature mapping method. Therefore, when we have enough training data, the use of joint feature vectors for feature mapping is expected to yield the highest recognition accuracy. (This fact was confirmed in our preliminary experiments.)

4. EXPERIMENTS

The proposed feature enhancement method was evaluated on the Aurora2 task, that is widely used for evaluation of noise-robustness techniques [9]. In this experiment, we used a clean acoustic model trained according to the complex back-end recipe of Aurora2. As a feature vector for speech recognition, we used a 39-dimensional vector consisting of 13 MFCCs (including C0) and their velocity and acceleration parameters. This means that the dimension of the joint feature vector $[\boldsymbol{y}^T, \hat{\boldsymbol{n}}^T]^T$ is 78. As described in the previous section, the proposed method compresses the joint vector space by using LDA. We set the subspace dimension at 39 to compare the proposed method with NMN-SPLICE in a fair manner. The sizes, K and M, of the GMMs needed for enhancement were set to 1024. In order to train the parameters of the proposed method, we used the multi-style training data set.

We limited our test sets to Aurora2 test sets A and B, where the same convolutive distortion is present as in the multi-style training

Table 1. Summary of word accuracies for SPLICE.

Set A	N1	N2	N3	N4	Avg.
SNR20	99.20	99.43	99.37	99.07	99.27
SNR15	98.43	98.73	98.66	98.15	98.49
SNR10	96.56	97.16	96.39	95.77	96.47
SNR5	90.39	87.61	87.98	87.57	88.39
SNR0	71.91	57.41	59.29	66.40	63.75
Avg.	91.30	88.07	88.34	89.39	**89.27**
Set B	N1	N2	N3	N4	Avg.
SNR20	99.11	98.97	99.16	99.29	99.13
SNR15	98.56	98.16	99.02	98.43	98.54
SNR10	96.47	94.50	97.17	95.53	95.92
SNR5	88.18	82.62	89.38	83.96	86.04
SNR0	63.65	52.30	63.64	51.65	57.81
Avg.	89.19	85.31	89.67	85.77	**87.49**

Table 2. Summary of word accuracies for NMN-SPLICE.

Set A	N1	N2	N3	N4	Avg.
SNR20	99.14	99.33	99.31	99.11	99.22
SNR15	98.25	98.55	98.93	98.24	98.49
SNR10	96.13	97.10	96.81	96.30	96.59
SNR5	90.39	90.05	89.86	89.17	89.87
SNR0	69.82	63.18	58.19	68.99	65.05
Avg.	90.75	89.64	88.62	90.36	**89.84**
Set B	N1	N2	N3	N4	Avg.
SNR20	99.08	99.03	99.16	99.38	99.16
SNR15	98.71	98.37	98.81	98.52	98.60
SNR10	96.81	95.13	97.38	96.61	96.48
SNR5	90.36	89.21	90.55	88.77	89.72
SNR0	68.38	62.67	67.61	60.44	64.78
Avg.	90.67	88.88	90.70	88.74	**89.75**

Table 3. Summary of word accuracies for our proposed method.

Set A	N1	N2	N3	N4	Avg.
SNR20	99.32	99.27	99.37	98.95	99.23
SNR15	98.62	98.79	98.87	98.33	98.65
SNR10	96.87	97.40	97.61	96.39	97.07
SNR5	91.56	90.45	90.34	89.79	90.54
SNR0	72.89	66.05	64.27	69.85	68.27
Avg.	91.85	90.39	90.09	90.66	**90.75**
Set B	N1	N2	N3	N4	Avg.
SNR20	99.05	99.09	99.11	99.32	99.14
SNR15	99.02	98.28	99.11	98.70	98.78
SNR10	96.93	96.10	97.91	96.67	96.90
SNR5	91.10	88.88	91.71	89.48	90.29
SNR0	70.89	63.48	72.50	63.44	67.58
Avg.	91.40	89.17	92.07	89.52	**90.54**

set. The motivation behind this is that the proposed method does not take the convolutive distortion into account at present. The multi-style training set consists of the following four different noise environments: Subway (N1), Babble (N2), Airport (N3), and Exhibition (N4). Test set A consists of the same noise environments as the multi-style training set while test set B contains Restaurant (N1), Street (N2), Airport (N3), and Train-station (N4), which are unseen in the training set.

Tables 1,2, and 3 show the word accuracies obtained with SPLICE, NMN-SPLICE, and our proposed method, respectively. NMN-SPLICE outperformed SPLICE in almost all noise environments. In particular, the improvement provided by NMN-SPLICE was prominent in set B. This result coincides with the conclusion of [6] and clearly shows that noise mean normalization effectively reduces the SPLICE's sensitivity to variability of noise environments. The proposed method outperformed NMN-SPLICE in all noise environments. The relative improvement over NMN-SPLICE was 8.3% on average. It is note worthy that the proposed method achieved a very high accuracy even for the babble noise environments (test set A, N2), where the noise characteristics change rapidly. This result indicates that proposed method can handle the non-stationarity of noise effectively.

Apart from the above experiment, we compared the proposed method to the VTS-based MFCC enhancement algorithm. As the noise model for VTS, we used a fixed single Gaussian model. The proposed method outperformed VTS, which achieved word accuracies of 88.93% and 88.84% for test sets A and B, respectively.

5. CONCLUSION

In this paper, we proposed a new feature enhancement technique that operates directly on MFCC vectors and exploits temporally changing noise estimates. The proposed method modifies the original SPLICE so that it uses temporally changing noise estimates for both space division and feature mapping of SPLICE. The proposed method was evaluated on the Aurora2 task, and it was shown that it could achieve highly effective and computationally feasible MFCC enhancement in non-stationary noise environments.

6. REFERENCES

[1] V. Stouten, "Robust Automatic Speech Recognition in Time-varying Environments," *PhD thesis*, 2006.

[2] A. Acero, Li Deng, T. Kristjansson, and J. Zhang, "HMM Adaptation Using Vector Taylor Series for Noisy Speech Recognition," *Proc. ICSLP*, pp. 869–872, 2000.

[3] Y. Zhao and B.H. Juang, "On noise estimation for robust speech recognition using vector Taylor series," *Proc. ICASSP*, pp. 4290–4293, 2010.

[4] Li Deng, J. Droppo, and A. Acero, "Enhancement of log mel power spectra of speech using a phase-sensitive model of the acoustic environment and sequential estimation of the corrupting noise," *Speech and Audio Processing, IEEE Transactions on*, vol. 12, no. 2, pp. 133 – 143, march 2004.

[5] M. Fujimoto and S. Nakamura, "Sequential Non-Stationary Noise Tracking Using Particle Filtering with Switching Dynamical System," *Proc. ICASSP*, pp. 769–772, 2006.

[6] J. Droppo, Li Deng., and A. Acero, "Evaluation of SPLICE on the Aurora 2 and 3 Tasks," *Proc. ICSLP*, pp. 29–32, 2002.

[7] Y. Qiao and N. Minematsu, "Mixture of probabilistic linear regressions: a unified view of GMM-based mapping techniques," *Proc. ICASSP*, pp. 3913–3916, 2009.

[8] T. Yoshioka and T. Nakatani, "Speech enhancement based on log spectral envelope model and harmonicity-derived spectral mask, and its coupling with feature compensation," *Proc. ICASSP*, pp. 5064–5067, 2011.

[9] H.G. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions," *Proc. ISCA ITRW ASR*, 2000.