

# Speaker-Invariant Features for Automatic Speech Recognition

S. Umesh, D. R. Sanand and G. Praveen

Department of Electrical Engineering

Indian Institute of Technology

Kanpur 208 016, INDIA

{sumesh, drsanand, gpraveen}@iitk.ac.in

## Abstract

In this paper, we consider the generation of features for automatic speech recognition (ASR) that are robust to speaker-variations. One of the major causes for the degradation in the performance of ASR systems is due to inter-speaker variations. These variations are commonly modeled by a pure scaling relation between spectra of speakers enunciating the same sound. Therefore, current state-of-the-art ASR systems overcome this problem of speaker-variability by doing a brute-force search for the optimal scaling parameter. This procedure known as vocal-tract length normalization (VTLN) is computationally intensive. We have recently used Scale-Transform (a variation of Mellin transform) to generate features which are robust to speaker variations without the need to search for the scaling parameter. However, these features have poorer performance due to loss of phase information. In this paper, we propose to use the magnitude of Scale-Transform and a pre-computed “phase”-vector for each phoneme to generate speaker-invariant features. We compare the performance of the proposed features with conventional VTLN on a phoneme recognition task.

## 1 Introduction

Inter-speaker variation is a major cause for the degradation in the performance of Automatic Speech Recognition (ASR). This is especially important in speaker-independent (SI) ASR systems which are typically built to handle speech from any arbitrary unknown speaker, e.g. in applications such as directory assistance. As a rule of thumb speaker-dependent (SD) ASR systems have half the error rates when compared to SI systems for the same task. This difference in performance suggests that the performance of SI systems can be significantly improved if we can account for inter-speaker variations. Note that in practice, we cannot *directly* build SD systems for large vocabulary speech recognition tasks, since the speech data required from that speaker will be too enormous even for a co-operative user. Instead, we build SI systems and then “adapt” the system for a particular speaker to make it speaker-dependent. Hence, irrespective of the application,

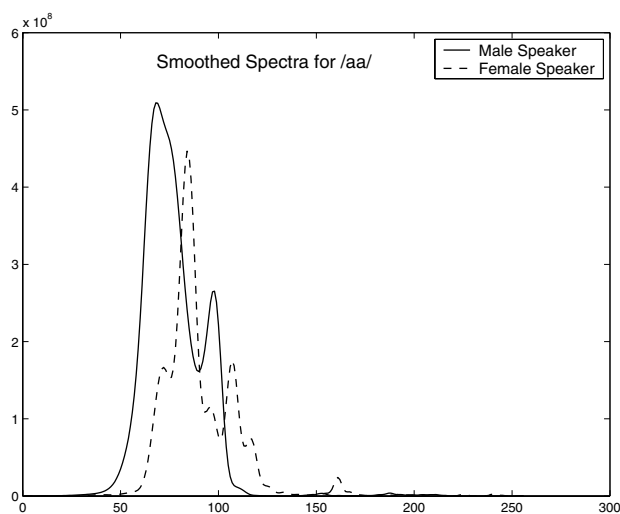


Figure 1: Spectra of 2 speakers for one “frame” of /aa/

there is a lot of interest in improving the performance of SI systems by accounting for inter-speaker variability.

In general, two speakers enunciating the same sound have very different pressure waveforms and the resulting spectra are very different. Fig. 1 shows the smoothed spectra (after smoothing out pitch) for two speakers enunciating the vowel /aa/. As seen from the figure the two spectra are very different even though it is the same vowel spoken by two speakers. Since ASR systems use features extracted from the spectra, the features are themselves different for the same sound enunciated by two speakers. Therefore for an automated system to recognize these different features as belonging to the same sound is difficult leading to a degradation in performance. Note that there will be a certain amount of variability for the same sound spoken by the same person, i.e. intra-speaker variability, which is handled by the statistical model. However, in general inter-speaker variability is substantially larger than intra-speaker variability, leading to “coarse” statistical models and hence increased confusion between the sound classes.

There is a lot of research that is being done in trying to understand the mathematical relationship between spectra of two speakers enunciating the same sound. This has impor-

tant implications in other areas apart from speech recognition, such as in vowel perception, hearing-aid design and speech pathology. One of the major causes for inter-speaker variations is attributed to the physiological differences in the vocal-tracts of the speakers. Based on this idea, in ASR a commonly used model to describe the relation between spectra of two speakers enunciating the same sound is given by

$$S_A(f) = S_B(\alpha f) \quad (1)$$

where  $S_A(f)$  and  $S_B(f)$  are the spectra of speakers  $A$  and  $B$  respectively; and  $\alpha$  denotes the uniform (constant) frequency-warp factor. We refer to the above model as *linear-scaling model*, since the two spectra are just scaled versions of each other. The motivation for linear scaling comes from the fact that to a first-order approximation, the vocal tract shape can be assumed to be a tube of uniform cross-section and for this simplifying approximation, the resonant frequencies (which characterize different sounds) are inversely proportional to vocal-tract length ([Wakita, 1977]). Therefore, differences in vocal-tract lengths manifest in scaling of resonant frequencies with the scaling being inversely proportional to the vocal-tract length.

In practical ASR systems, we have access only to the acoustic data from the speaker and therefore do not have any idea of the speaker's vocal-tract length. Therefore, we do a brute-force search for the optimal scaling-factor,  $\alpha$  by trying out different values of  $\alpha$ . The optimality criterion is based on maximizing the likelihood with respect to the SI Hidden Markov Model (HMM). Since in ASR we use features derived from the spectra, for each value of  $\alpha$  we scale the frequency-spectra appropriately and then compute the features. Therefore for each value of  $\alpha$ , we recompute the feature of the  $i^{th}$  utterance,  $\mathbf{X}_i^\alpha$ , for that scale-factor. We then find the optimal scale-factor by

$$\hat{\alpha}_i = \arg \max_{\alpha} \Pr(\mathbf{X}_i^\alpha | \lambda, W_i) \quad (2)$$

where  $\alpha$  represents scale factor.  $\lambda$  is the HMM model and  $W_i$  is the transcription of the utterance. Due to physiological constraints of the human vocal-tract apparatus,  $\alpha$  lies in the range of 0.80 to 1.20. Therefore, in state-of-the-art ASR systems, we compute the features for different values of  $\alpha$  from 0.80 to 1.20 in steps of 0.02 and find the optimal  $\alpha$  using Eq. 2. This is a computationally *expensive* process since for every frame the features have to be computed 21 times corresponding to different  $\alpha$ , before finding the optimal feature for that frame after a maximum-likelihood (ML) search. Hence, a lot of research is being done to have alternate normalization schemes that are computationally efficient.

Recently, an alternate method for generating features that are invariant to speaker-variations due to the linear scaling (see Eq. 1) was proposed [Umesh *et al.*, 1999]. In this method, the features are insensitive to the scale-factor  $\alpha$  and hence there is no need to search for the optimal  $\alpha$ . We describe the method in the next section.

## 2 Review of Scale-Invariant Features using Scale-Transform

In the linear-scaling model of Eq. 1, the speaker-dependent warp-factor,  $\alpha$  is a multiplicative constant and is independent

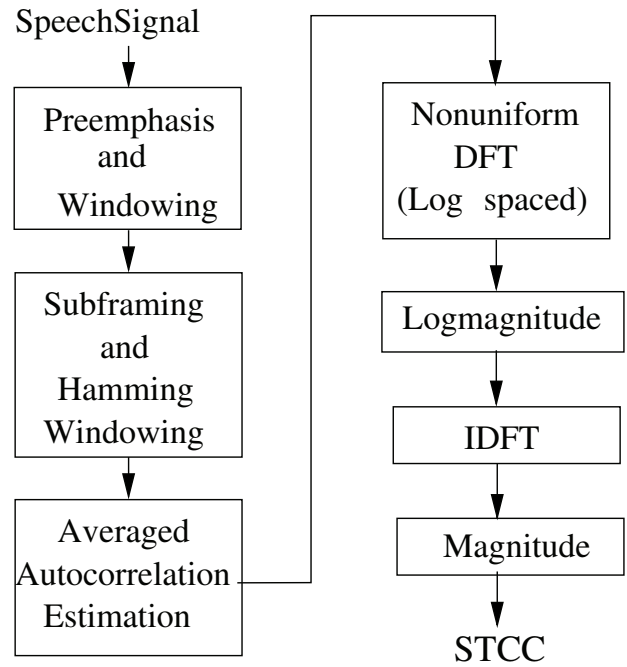


Figure 2: Block diagram for the computation of STCC

of frequency. For such a model, log-warping the frequency-axis of the spectra of speakers separates the linear scaling factor as a translation factor in log-warped spectral domain, i.e.,

$$\begin{aligned} s_a(\lambda) &= S_A(f = e^\lambda) = S_B(\alpha e^\lambda) \\ &= S_B(e^{\lambda + \ln \alpha}) = s_b(\lambda + \ln \alpha). \end{aligned} \quad (3)$$

Hence, in the log-warped domain,  $\lambda$ , the speaker-dependent scale-factor separates out as a translation factor  $\ln \alpha$  for the linear scaling model of Eq. 1.

Let the Fourier-transform of the log-warped spectra  $s_a(\lambda)$  be denoted as  $D_a(c)$  and that of  $s_b(\lambda)$  be denoted as  $D_b(c)$ , i.e.

$$s_a(\lambda) \xleftrightarrow{DFT} D_a(c) \quad \text{and} \quad s_b(\lambda) \xleftrightarrow{DFT} D_b(c) \quad (4)$$

Using the property of the Fourier-transform that a translation in one domain appears only as linear-phase in the other domain we have,

$$D_a(c) = D_b(c) e^{-j c \ln \alpha} \quad (5)$$

Therefore, the speaker-dependent term  $\alpha$  appears only in the phase term. Hence the magnitudes of  $D_a(c)$  and  $D_b(c)$  are identical and independent of the speaker-variations i.e.

$$|D_a(c)| = |D_b(c)|. \quad (6)$$

We can therefore use the magnitude as speaker-invariant features for ASR.

We can show that the above procedure is equivalent to taking the Scale-Transform [Cohen, 1993] (a special case of Mellin transform) of  $S_A(f)$  and  $S_B(f)$  and then using the magnitude of Scale-Transform as features in ASR. In this paper, we refer to these features as Scale-Transform Cepstral

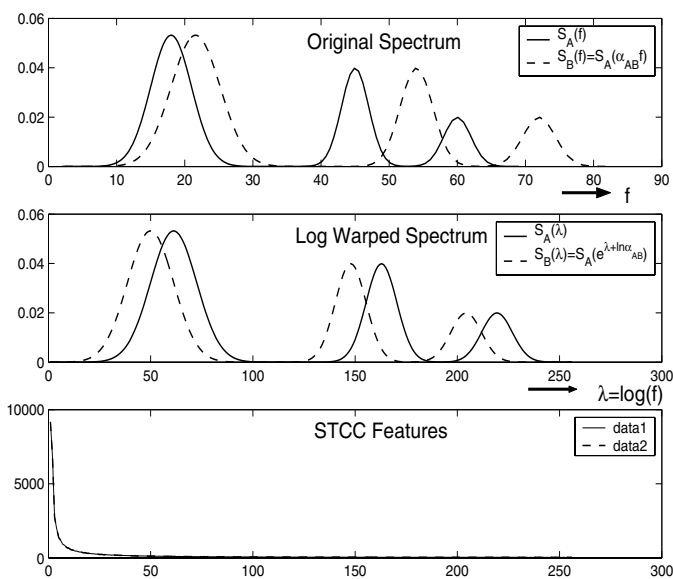


Figure 3: Figure shows that linear-scaling of original spectra, results in their being shifted versions in  $\lambda = \log(f)$  domain. Therefore, the magnitude of the subsequent Fourier-transform (STCC) are identical.

Coefficients (STCC). The steps to compute STCC are shown in Fig. 2.

Fig. 3 illustrates the idea of using Scale-Transform to obtain speaker-invariant features for the linear-scaling model of Eq.1 using a synthetic example. Fig. 3(a) shows synthetic spectra of two speakers enunciating the same sound that are *exactly* scaled versions of each other corresponding to the linear-scaling model of Eq.1. In Fig. 3(b) we show the same spectra in the frequency-warped (i.e. log-warped) domain. As seen from the figure, the frequency-warped spectra are *exactly* shifted versions of each other as expected from Eq. 3. Finally, we take the Fourier-transform of these frequency-warped spectra and then take the magnitude of the Fourier-transform to get the STCC features. These are shown in Fig. 3(c). As seen from the figure, the STCC features are identical, and are therefore invariant to speaker variations that are present in the original spectra as seen in Fig. 3(a).

Since the linear scaling model of Eq. 1 is only a crude approximation, recently [Umesh *et al.*, 2002] have proposed the use of mel-scale for frequency-warping and not the log-scale. With this mel-scale frequency-warping they claim better results [Sinha and Umesh, 2002]. Therefore, for the purposes of this paper, we will use mel-scale warping to compute the STCC features.

We now compare the performance of conventional MFCC features with STCC features. TIMIT database was used for the experiments and we considered the classification of 42 monophones. The train and test set consist of 155015 and 56424 monophone occurrences for adult male and female speakers respectively. The HMM models consisted of 3 emitting and 2 non-emitting states, with left-to-right and with-

	STCC	MFCC
Acc%	56.98	62.14

Table 1: Accuracy of the phoneme recognizer using STCC and MFCC

out skips over states. Two mixture Gaussians with full covariance matrices per state were used. The features vectors are of 39 dimensions comprising normalized log-energy,  $c_1 \dots c_{12}$  (excluding  $c_0$ ) and their first and second order derivatives. Conventional MFCC features are computed as described in [Lee and Rose, 1998] while the STCC features are computed as described in [Umesh *et al.*, 1999] and a block diagram is shown in Fig. 2.

The classification performance between STCC and the MFCC features are shown in Table 1. As seen from the Table, STCC features have a lower classification performance when compared to MFCC features. Since STCC provides speaker-invariant features, it should have provided improved normalization performance when compared to MFCC. The reason for the degradation is the complete loss of phase-information. We discuss about the loss of phase information in Section 3 and propose new features to overcome this loss of information by introducing an average phase vector which is discussed in detail in Section 4.

### 3 Drawback of STCC – loss of phase information

Theoretically, STCC provides speaker-normalization by exploiting the fact that the magnitude of STCC are identical as seen in Eq. 6. However, note that  $D_a(c)$  and  $D_b(c)$  are *complex* quantities with associated magnitude and phase. Let

$$D_a(c) = |D_a(c)|e^{j\phi(c)} \quad \text{and} \quad (7)$$

$$D_b(c) = |D_a(c)|e^{j\phi(c)}e^{-j\ln \alpha_{AB}c}. \quad (8)$$

Therefore  $|D_a(c)| = |D_b(c)|$ . However, we have *completely lost* the phase information  $e^{j\phi(c)}$  which is also important for discrimination of phonemes. This loss in phase information degrades the performance of the STCC and even though it is insensitive to speaker-variations, the final performance is inferior to the MFCC. In the next section, we *propose* a method to overcome this problem and help improve the performance of the STCC.

### 4 Proposed Improvement over STCC features

As seen in the previous section, the loss in phase information  $e^{j\phi(c)}$  leads to degradation in performance when compared to MFCC features even though STCC features are invariant to speakers as seen in Eq. 6. As seen in Eq. 8,  $D_a(c)$ ,  $D_b(c)$  contain both the magnitude and phase information and the speaker-specific factor,  $\alpha$ , appears only in the phase term. Hence retaining the complete phase will result in *no* normalization since the  $\ln \alpha$  will also be present. On the other hand, taking *only* the magnitude will result in the loss of  $e^{j\phi(c)}$  information which provides additional information for discrimination between vowels.

In this section, we propose a method to incorporate the phase information  $e^{j\phi(c)}$  but not the speaker-specific  $\ln \alpha$  in the STCC features. The basic idea of this approach is to estimate an “average” phase vector for each phoneme using training data from all speakers.

As previously discussed, for the linear-scaling model, the speaker differences manifest themselves as speaker-specific shifts in the log-warped domain as seen in Fig. 3. The STCC exploits this fact by considering the magnitude of the subsequent Fourier-transform but in the process loses the phase information completely. Our approach is to estimate the average phase for each phoneme from the training data and use this same phase-vector for every occurrence of that phoneme irrespective of the speaker. For example, for the phoneme /ae/, our *average-phase* STCC features with acronym AP-STCC will be of the type  $|D_a(c)|e^{j\phi_{avg-ae}(c)}$ . We will first illustrate this idea through a synthetic example that is discussed below.

Consider that a particular phoneme is enunciated by three different speakers and the corresponding spectra after frequency-warping are shifted versions of one another as shown in Fig. 4(a). The three spectra in the Fourier-domain will differ only in a linear-phase term and this can be seen mathematically as:

$$\begin{aligned} s_a(\lambda) &\stackrel{DFT}{\rightleftharpoons} D_a(c) = |D_a(c)|e^{j\phi(c)} & (9) \\ s_a(\lambda - \tau_1) &\stackrel{DFT}{\rightleftharpoons} D_a(c)e^{-j\tau_1c} = |D_a(c)|e^{j\phi(c)}e^{-j\tau_1c} \\ s_a(\lambda + \tau_2) &\stackrel{DFT}{\rightleftharpoons} D_a(c)e^{+j\tau_2c} = |D_a(c)|e^{j\phi(c)}e^{+j\tau_2c} \end{aligned}$$

If we now compute the average phase of all the three phases, i.e.

$$\begin{aligned} \phi_{avg}(c) &= \frac{\phi(c) + (\phi(c) - \tau_1c) + (\phi(c) + \tau_2c)}{3} & (10) \\ &= \phi(c) + \frac{(\tau_2 - \tau_1)c}{3}, \end{aligned}$$

then the average-phase  $\phi_{avg}(c)$  is same as the phase of the phoneme,  $\phi(c)$ , with an additional linear-phase term that corresponds to the average of all the three shifts. Therefore, the average phase  $\phi_{avg}(c)$  preserves the phase of the phoneme with an additional term corresponding to the average shift of the spectra of the training speakers. If we now use this average phase for all speakers along with the magnitude of the STCC feature, we get speaker-normalized features. This can be easily seen by recalling the fact that  $|D_a(c)|$  is constant for all speakers; and when it is multiplied by the *same* phase term  $e^{j\phi_{avg}(c)}$  for all speakers, then the resulting features are same for all speakers. This is shown in Fig. 4.

In practice, since the processing is done in the discrete-domain, the phases are  $2\pi$  wrapped, and hence we first unwrap the phases before averaging. The other important practical limitation at this point is the fact that we need to know the phoneme *a priori* so that we can add the appropriate average-phase of that phone to the magnitude vector  $|D_a(c)|$ . We are now working on practical algorithms to find the appropriate phase for a given frame without having the knowledge of which phoneme the frame came from. However, as a first

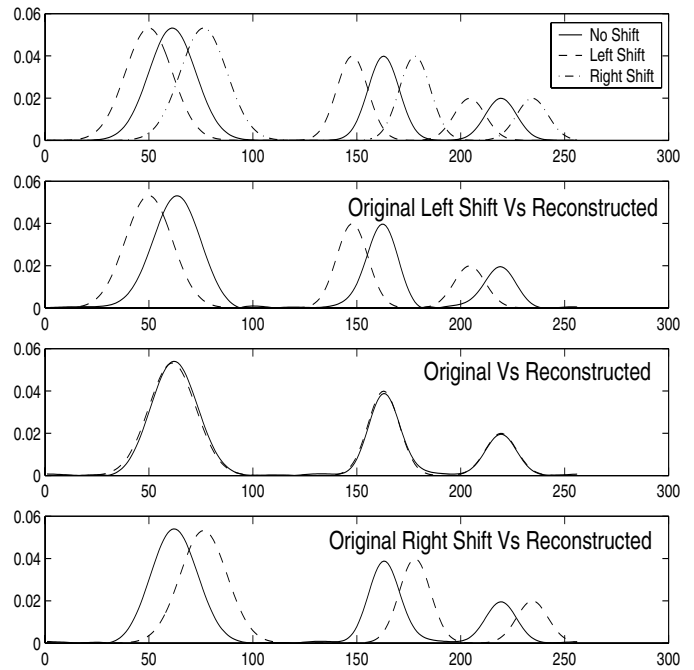


Figure 4: Figure shows our proposed method of normalization using average phase. The top figure shows the shifted spectra from different speakers in the frequency-warped domain. Using the average phase and the magnitude of the DFT we then reconstruct the spectra (shown in solid line).

step, we will assume knowledge of appropriate phase vector and test the efficacy of this method of normalization and compare it with the conventional method of normalization.

## 5 Performance of the Proposed Normalization Method

In this section, we will compare the performance of the proposed normalization scheme with conventional normalization scheme.

Here we considered the classification of 8 “most confusable” vowels in contrast to the classification of 42 monophones used for MFCC and STCC feature vectors discussed in section.2. The main reason was to study the effect of our proposed phase-estimation procedure in more detail. The vowels were extracted from TIMIT database, where the train and test set consisted of 22686 and 8265 utterances respectively. The HMM models consisted of 3 emitting and 2 non-emitting states, with left-to-right and without skips over states. We used single mixture Gaussian with diagonal variance for each state. The feature vectors are of 26 dimensions comprising normalized log-energy,  $c_1 \dots c_{12}$  (excluding  $c_0$ ) and their first order derivatives.

The conventional normalization scheme is based on Eq. 2, which involves ML estimation of the warp-factor  $\alpha$ . This method is also popularly known as Vocal-tract length normalization (VTLN). As discussed in the introduction, we do a brute-force search for the optimal  $\alpha$  by computing the MFCC feature for each  $\alpha$  by appropriately scaling the mel filter-bank

Condition	No Norm. Baseline	Norm. Recog. Trans.	Norm. True Trans.
VTLN-MFCC	60.99	61.17	69.69
AP-STCC	60.42	36.58	99.36

Table 2: % of Accuracy for VTLN-MFCC and AP-STCC speaker normalization methods for classification of 8 vowels in TIMIT test set. First column is without normalization. In the second column we have used recognition output of first column as transcription for the normalization, while the third column shows the normalization performance when the *true* transcription is known.

and choosing the feature that maximizes the likelihood with respect to the statistical HMM model and the given transcription  $W_i$ . Note that although the method needs transcription, there is graceful degradation when there are errors in the transcription.

In our proposed method, we are still working on methods to find the optimal phase-vector to multiply a given frame of speech. As seen from Table 2 if we are given exact transcription, then we can multiply by the correct average-phase vectors and the performance is exceptional. On the other hand, even with exact transcription the VTLN method is far inferior to the proposed method of using AP-STCC features. But, if we use the recognition output of the baseline recognizer (i.e. without normalization) as the transcription then the VTLN normalization method degrades gracefully while the AP-STCC completely falls apart – it is worse than even baseline. Note that we do not need to know the transcription for AP-STCC, but a method of finding “optimal” phase-average vector for each utterance. We are working on various distance measures to obtain an “optimal” solution. However, it is important to note that at least under ideal conditions (when true transcription is known) our method shows that it is possible to remove almost all the speaker-variability which conventional VTLN-MFCC is never able to achieve.

Another approach to measure the normalization performance is to measure the separability of the vowel models using the various normalization schemes. One good measure of separability is the F-ratio between models considered pair-wise. Tables 3,4,5,6 shows the F-ratio for the unnormalized features, the VTLN features and proposed AP-STCC features. These tables show F-ratio of models that are built using training data for which true transcription is always known. The F-ratio shows the separability between models and the higher the number the better. Note that since MFCC and STCC are computed slightly differently the features are slightly different and hence there are small differences in performance in the un-normalized case. From the Table, it can be seen that the separability between the vowels models are excellent for the AP-STCC features.

## 6 Conclusion & Discussion

In this paper, we have proposed a method of obtaining speaker-invariant features for automatic speech recognition. Our method is motivated by the fact that conventional methods reduce inter-speaker variability by doing a

brute-force search for optimal features and are therefore computationally very expensive. Recently a method has been proposed that uses a special transform called Scale-Transform to obtain speaker-invariant features. Since the Scale-Transform uses only the magnitude of the features to obtain speaker-invariance, there is a loss of discriminability between phonemes due to the loss of phase information. In our proposed method, we use the average-phase of the corresponding phoneme in place of the lost phase information of STCC. We are currently working on various distance measures to find the appropriate average-phase for normalization. Using vowel classification experiments and F-ratio measures we show that if we have optimal average-phase information then the proposed method provides excellent normalization performance when compared to the conventional VTLN method.

## 7 Acknowledgments

This work was supported in part by funding from the Dept. of Science & Technology, Government of India under project No. SR/S3/EECE/0008/2006-SERC-Engg.

## References

- [Cohen, 1993] L. Cohen. The Scale Representation. *IEEE Trans. Signal Processing*, 41:3275–3292, Dec. 1993.
- [Lee and Rose, 1998] L. Lee and R. Rose. Frequency Warping Approach to Speaker Normalization. *IEEE Trans. Speech Audio Processing*, 6:49–59, Jan. 1998.
- [Sinha and Umesh, 2002] Rohit Sinha and S. Umesh. Non-Uniform Scaling Based Speaker Normalization. In *Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 1, pages 589–592, 2002.
- [Umesh *et al.*, 1999] S. Umesh, L. Cohen, N. Marinovic, and D. Nelson. Scale Transform in Speech Analysis. *IEEE Trans. Speech Audio Processing*, Jan. 1999.
- [Umesh *et al.*, 2002] S. Umesh, L. Cohen, and D. Nelson. Frequency Warping and the Mel Scale. *IEEE Signal Processing Letters*, 9(3):104–107, March 2002.
- [Wakita, 1977] H. Wakita. Normalization of Vowels by Vocal Tract Length and its Application to Vowel Identification. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-25(2):183–192, Apr. 1977.

	aa	ae	ah	eh	er	ih	ow	uh
aa	0	10.6791	3.3998	10.0033	13.6230	20.2214	8.6107	9.7719
ae	10.6791	0	6.1473	2.7679	19.2485	5.5922	16.4265	9.3211
ah	3.3998	6.1473	0	4.7876	11.2611	10.4775	6.7118	4.3584
eh	10.0033	2.7679	4.7876	0	12.4709	3.8072	13.1436	5.0537
er	13.6230	19.2485	11.2611	12.4709	0	18.3380	18.3751	11.4699
ih	20.2214	5.5922	10.4775	3.8072	18.3380	0	16.0073	4.7921
ow	8.6107	16.4265	6.7118	13.1436	18.3751	16.0073	0	6.9622
uh	9.7719	9.3211	4.3584	5.0537	11.4699	4.7921	6.9622	0

Table 3: F-ratio test for Un-Normalized AP-STCC

	aa	ae	ah	eh	er	ih	ow	uh
aa	0	166.9895	45.6957	145.0410	64.4161	254.3571	73.0718	116.6179
ae	166.9895	0	84.6296	35.5867	162.3246	126.5625	171.1731	138.8447
ah	45.6957	84.6296	0	55.6417	48.9849	141.6923	57.2495	50.8937
eh	145.0410	35.5867	55.6417	0	114.7508	67.1734	118.8632	74.0480
er	64.4161	162.3246	48.9849	114.7508	0	180.8260	79.5003	76.1799
ih	254.3571	126.5625	141.6923	67.1734	180.8260	0	144.3595	61.7577
ow	73.0718	171.1731	57.2495	118.8632	79.5003	144.3595	0	33.4184
uh	116.6179	138.8447	50.8937	74.0480	76.1799	61.7577	33.4184	0

Table 4: F-ratio test for Normalized AP-STCC

	aa	ae	ah	eh	er	ih	ow	uh
aa	0	11.1627	3.1279	10.1707	14.7092	20.9919	10.2234	10.1223
ae	11.1627	0	8.3630	2.9611	19.4991	6.2119	19.2229	10.1541
ah	3.1279	8.3630	0	5.6553	12.6427	12.2763	8.0236	4.3762
eh	10.1707	2.9611	5.6553	0	12.5816	4.1006	14.8935	5.3250
er	14.7092	19.4991	12.6427	12.5816	0	17.6689	21.5590	12.1847
ih	20.9919	6.2119	12.2763	4.1006	17.6689	0	17.9766	5.0690
ow	10.2234	19.2229	8.0236	14.8935	21.5590	17.9766	0	8.6819
uh	10.1223	10.1541	4.3762	5.3250	12.1847	5.0690	8.6819	0

Table 5: F-ratio for Un-Normalized VTLN-MFCC

	aa	ae	ah	eh	er	ih	ow	uh
aa	0	16.7331	4.4782	13.7949	24.1999	25.9021	11.8576	12.6169
ae	16.7331	0	13.3697	4.0665	29.5038	8.8052	27.0284	18.3779
ah	4.4782	13.3697	0	8.4281	20.4874	16.9298	11.2715	7.0433
eh	13.7949	4.0665	8.4281	0	18.2864	5.3348	20.8485	9.34
er	24.1999	29.5038	20.4874	18.2864	0	23.7503	32.8729	18.3630
ih	25.9021	8.8052	16.9298	5.3348	23.7503	0	22.5615	8.2142
ow	11.8576	27.0284	11.2715	20.8485	32.8729	22.5615	0	10.0568
uh	12.6169	18.3779	7.0433	9.3426	18.3630	8.2142	10.0568	0

Table 6: F-ratio for Normalized VTLN-MFCC. Note that the F-ratio is considerably smaller than that of Normalized AP-STCC