Using Phonetic Posteriorgram Based Frame Pairing for Segmental Accent Conversion

Guanlong Zhao, *Student Member, IEEE,* and Ricardo Gutierrez-Osuna, *Senior Member, IEEE*

***Abstract*— Accent conversion (AC) aims to transform non-native utterances to sound as if the speaker had a native accent. This can be achieved by mapping source speech spectra from a native speaker into the acoustic space of the target non-native speaker. In prior work, we proposed an AC approach that matches frames between the two speakers based on their acoustic similarity after compensating for differences in vocal tract length. In this paper, we propose a new approach that matches frames between the two speakers based on their phonetic (rather than acoustic) similarity. Namely, we map frames from the two speakers into a phonetic posteriorgram using speaker-independent acoustic models trained on native speech. We thoroughly evaluate the approach on a speech corpus containing multiple native and non-native speakers. The proposed algorithm outperforms the prior approach, improving ratings of acoustic quality (22% increase in mean opinion score) and native accent (69% preference) while retaining the voice quality of the non-native speaker. Further, we show that the approach can be used in the reverse conversion direction, i.e., generating speech with a native speaker's voice quality and a non-native accent. Finally, we show that this approach can be applied to non-parallel training data, achieving the same accent conversion performance.**

***Index Terms*—accent conversion, voice conversion, acoustic model, posteriorgram.**

I. Introduction

LEARNERS who acquire a second language (L2) after a "critical period" [1] usually speak with a non-native accent. Having a non-native accent can often reduce the speaker's intelligibility [2] and may also lead to discriminatory attitudes [3], [4]. Therefore, non-native speakers have much to gain by improving their pronunciation. Several studies [5], [6] have shown that having a suitable native (L1) speaker to imitate – a so-called "golden speaker" with similar voice characteristics as the learner but with a native accent, can be beneficial in pronunciation training. Based on these findings, Felps *et al.* [7] suggested that such a "golden speaker" could be created by resynthesizing the non-native speaker's own voice with a native accent borrowed from a native reference speaker.

Traditional voice-conversion (VC) methods [8]–[11] cannot be used for this purpose since VC cannot decouple the speaker's voice quality from her or his accent, i.e., VC assumes that accent is part of the speaker's identity. In this work, we distinguish two concepts: *voice quality*, which focuses on the physical characteristics of the speaker's voice (e.g., vocal tract and glottal configuration, pitch range), and *speaker identity*, a combination of *voice quality* and other speaker characteristics (e.g., accent, speaking rate, intonation, word choice).

To address the accent-and-voice-quality entanglement issue of traditional VC methods, Aryal and Gutierrez-Osuna [12] proposed a modified VC method where source frames (i.e., from the native reference speaker) and target frames (i.e., from the non-native speaker) were paired based on their acoustic similarity. In a first step, the authors applied vocal-tract length normalization (VTLN) to the source speech, so it matched the target speaker's vocal-tract length. Then, they paired each frame in the source corpus with the closest frame in the target corpus, and vice versa. Though VTLN did improve frame pairing compared to time alignment (i.e., the conventional approach in VC), vocal-tract length is just one of the potentially many differences between speakers, and it is too coarse to account for differences in pronunciation.

To address this issue, we present an approach that matches source and target frames based on their phonetic content. Leveraging advances in acoustic modeling [13], we extract phonetic information from phonetic posteriorgrams (PPGs) [14]. Namely, we compute the posteriorgram for each source and target speech frame through a speaker-independent acoustic model trained on a large corpus of native speech. Then, we use the symmetric Kullback-Leibler (KL) divergence [15] in posteriorgram space to match source and target frames. The result is a set of source-target frames that are paired based on their phonetic similarity, with which we train a Gaussian Mixture Model (GMM) to model the joint distribution of source and target Mel-Cepstral Coefficients (MCEPs). In a final step, we map source MCEPs into target MCEPs using maximum likelihood estimation of spectral parameter trajectories considering the global variance [8] of the target speaker. Our implementation is based on a conventional GMM spectral mapping method to ensure a fair comparison with the prior study [12], but our proposed frame matching method can be combined with any spectral mapping methods (e.g., neural networks, frequency warping) that take frame pairs as input.

Our approach differs from prior works on accent conversion, which modify speech features that carry accent information, such as prosody, formants, spectral envelopes, or articulatory gestures [7], [16]–[18]. Instead, we use a VC technique to capture the voice quality of the (target) non-native speaker while preserving the (source) native speaker's pronunciation characteristics – both segmental and prosodic. Unlike VC methods, however, we avoid the issue of time aligning source and target utterances, which is problematic when the target speaker is non-native. Our approach is related to that of Xie *et al.* [19], who used speaker-adaptive acoustic models to generate

Manuscript received October 3, 2018; revised July 1, 2019.

G. Zhao and R. Gutierrez-Osuna are with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: gzhao@tamu.edu; rgutier@tamu.edu).

posteriorgrams for VC. Their method groups all target speaker training data into phonetic clusters in the posteriorgram space using symmetric KL divergence and K-means clustering. Then, each frame of the source speaker's corpus is mapped to the centroid of the closest target phonetic cluster. The final converted speech is generated from those closest cluster centroids using the maximum probability trajectory generation algorithm. In contrast with their frame clustering approach, we use PPGs to produce frame pairs between source and target speakers, and then we train a GMM using those frame pairs. A second major difference with their approach is that we use speaker-independent acoustic models trained on native speech to ensure that the PPGs only reflect native pronunciations, whereas their approach uses speaker-adaptive training, which would introduce non-native pronunciations into the acoustic models. Initial findings from this work were presented at the International Conference on Acoustics, Speech, and Signal Processing (ICASSP) in 2018 [20]. That earlier conference paper presented preliminary listening test results that verified the effectiveness of the PPG-based frame-matching method. The present manuscript describes our method in detail and significantly expands the perceptual studies and data analyses, including an experimental comparison of the proposed method on parallel and non-parallel data.

The manuscript is organized as follows. Section II reviews prior work on accent conversion in the acoustic and articulatory domains, and discusses the connection between accent conversion and voice conversion. Section III describes the experimental methods used in the study, including the phonetic posteriorgrams, the proposed frame-pairing method for accent conversion, as well as the baseline systems we used for comparison. Section IV describes the experimental setup used for the study, including the speech corpus we used to train the acoustic model, and the native/non-native speech corpus we used for accent conversion. Section V presents results on three different experiments we used to evaluate various aspects of the algorithm. The paper concludes with a discussion of our findings and directions for future work.

II. Literature Review

A. *Algorithms for accent conversion*

Foreign and non-native accents occur when speech deviates from the expected acoustic (e.g., formants) and prosodic (e.g., intonation, duration, and rate) norms of a language [7]. Therefore, prior work has focused on modifying certain speech characteristics to alter the perceived accent. In early work, Yan *et al.* [21] used a voice-morphing software to change the trajectories of formants, pitch, and duration to convert between three different English dialects (British, Australian, and General American English). The authors found that prosodic modifications produced noticeable differences on perceived accent, although not as significant as those produced by modifying formant trajectories. In the approach of Felps *et al.* [7], the spectral envelope of the non-native speech was replaced with that of the native speaker's, which had been normalized to the non-native speaker's vocal tract length with a piecewise linear warping function. Their results showed that the segmental correction was able to significantly reduce the foreign accentedness of the modified utterances. More recently, Jügler *et al.* [22] used PSOLA to correct the prosody of non-native German speech spoken by native French speakers. Prosodic (duration and pitch) corrections were performed at the syllable level, and the results showed a moderate but significant reduction in accentedness of the corrected speech.

A couple of studies also tried to blend native and non-native spectra to control the accent. Huckvale and Yanagisawa [17] blended the spectral envelope of non-native Japanese speech produced by an English Text-To-Speech (TTS) with its native counterpart through voice morphing to reduce the accent. Aryal *et al.* [18] decomposed the cepstrum into spectral slope and spectral detail, and then generated accent conversions by combining the spectral slope of the non-native speaker with a morph of the spectral detail of the native speaker. Though these spectra-blending methods can reduce non-native accents, they also tend to produce syntheses that are perceived as a "third speaker," one who is different from either the source (native) or target (non-native) speaker. To tackle this problem, Aryal and Gutierrez-Osuna [12] adapted VC techniques to perform accent conversion. The authors used vocal-tract-length normalization (VTLN) before pairing acoustic frames between source (native) and target (non-native) speaker, then built a GMM using those frame pairs to perform VC. This method was able to reduce non-native accent significantly, while retaining the non-native speaker's voice quality; however, it required a relatively large set of parallel recordings from the two speakers, and VTLN only accounted for a subset of the speaker characteristics.

An alternative to using acoustic methods is to operate in the articulatory domain. Along these lines, Felps *et al.* [16] used an articulatory synthesizer based on unit-selection to replace mispronounced non-native diphones with those from the non-native corpus that matched the articulatory configuration of a reference utterance from a native speaker. Later, Aryal and Gutierrez-Osuna used GMMs [23] and DNNs [24] to build an articulatory synthesizer (i.e., a mapping from articulatory gestures into acoustics) for the non-native speaker, then drove the GMM/DNN with articulatory gestures from a native speaker. Methods based on articulatory data generate syntheses that sound more like the non-native (target) speaker than acoustic methods, since they effectively decouple linguistic information (e.g., articulatory gestures from a native [source] speaker) from voice quality (captured by the articulatory-to-acoustic synthesizer of the non-native speaker). However, articulatory methods are expensive and require specialized equipment to collect articulatory data, so they are impractical for pronunciation training.

B. *Connection between accent and voice conversion*

Accent conversion is closely related to the problem of voice conversion [25]. Voice conversion transforms a source speaker's speech into that of a (known) target speaker. The conversion aims to match the voice characteristics of the target speaker, which may include vocal tract configuration, glottal

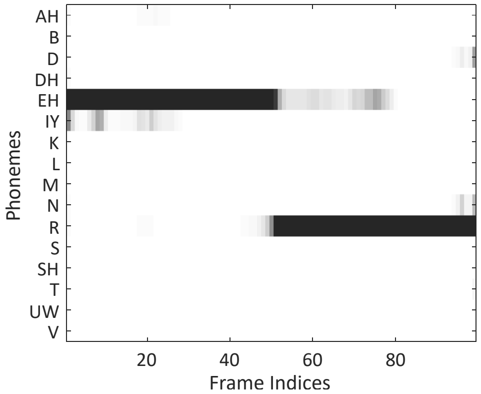


Fig. 1. PPG for the word "air," whose phonetic transcription in ARPABET is "EH R." For visualization purposes, we used a subset of the ARPABET phoneme set and omitted phonemes that had small values.

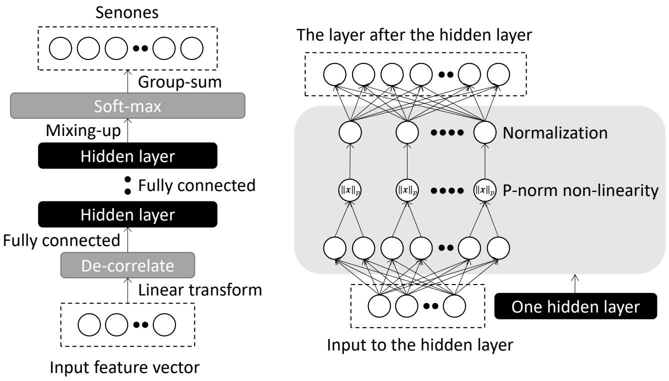


Fig. 2. P-norm deep neural network structure for acoustic modeling.

characteristics, pitch range, pronunciation, and speaking rate. Ideally, the only information retained from the source speech is its linguistic content, i.e., the words that were uttered. Popular methods for voice conversion include joint-density GMMs [8], frequency warping [26], [27], DNNs [28], [29], and sparse coding [11], [30]–[32]. Accent conversion modifies speech at a finer level of granularity, and seeks to combine the linguistic content and pronunciation of the source speaker with the voice quality of the target speaker. Therefore, accent conversion is a more challenging problem than voice conversion in the sense that, first, there is no ground truth for the output voice, and second, accent conversion needs to split the speech into voice quality (converted) and accent (preserved), whereas voice conversion jointly converts both.

III. Methods

A. Phonetic Posteriorgrams

At its core, our proposed method relies on Phonetic Posteriorgrams (PPGs) to measure the similarity of speech frames across speakers. A phonetic posteriorgram is computed by segmenting speech into frames and computing the posterior probability that each frame belongs to a set of pre-defined phonetic units. As an example, Fig. 1 shows the PPG of the spoken word "air." In practice, it is advisable to include context when computing the PPG by concatenating each speech frame with its neighboring right and left frames. Moreover, phoneme labels are too coarse to describe the variety of speech sounds. Therefore, the dimensions in a phonetic posteriorgram are often associated with triphones, as we will see next.

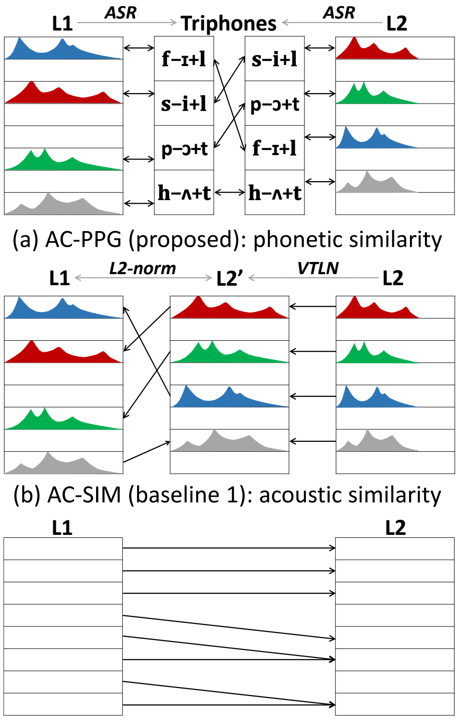
Generally, the phonetic posteriorgram is computed from the acoustic model in an automatic speech recognizer (ASR). The acoustic model in ASR acts as a sequential classifier: given an input acoustic feature vector, the acoustic model assigns how likely it is that the vector belongs to each of a set of states/senones. In recent years, acoustic models based on DNNs have yielded state-of-the-art speech recognition accuracy [13]. The most advanced ASR systems can achieve Word Error Rates that are comparable to or better than expert human transcribers on specific tasks [33].

In this work, we compute phonetic posteriorgrams using a p-norm DNN [34] as the acoustic model. The input layer accepts a feature frame accompanied by its left and right neighbors; then the input is de-correlated by a fixed linear transformation [35]. The de-correlated features are then passed through N hidden layers, each employing the p-norm nonlinearity $\boldsymbol{y} = \|\boldsymbol{x}\|_p = (\sum_i |x_i|^p)^{\frac{1}{p}}$, where $\boldsymbol{y}$ is one output dimension of a hidden layer and $\boldsymbol{x}$ represents a group of hidden neurons of that layer. Therefore, the number of output dimensions of each hidden layer is smaller than the number of hidden neurons. The output of the p-norm layer is then processed by a normalization layer to limit its standard deviation to one [34]. The output of the final hidden layer is fed into a softmax layer that produces more output nodes than the desired number of senones using a technique called "mixing-up" [34]. "Mixing-up" operates as follows. About halfway through training, the dimension of the softmax layer is increased by letting each output senone's probability be a sum over potentially multiple "mixture components." The mixture components are distributed using a power rule, proportional to the senone class priors. The neural network then "group-sums" the output of the softmax layer according to the group assignment defined in the "mixing-up" step, resulting in the final output nodes that correspond to individual senones. Fig. 2 shows the overall structure of the p-norm deep neural network that we use in this work.

During training, inputs to the p-norm DNN consist of stacked MFCC frames $\boldsymbol{X}$, whereas target outputs $\boldsymbol{Y}$ are senone labels obtained from force-alignment using an existing GMM-HMM speech recognizer. The training objective is the sum (across all frames of training data) of the log-probability of $\boldsymbol{Y}$ given $\boldsymbol{X}$: $\sum_i \log p(\boldsymbol{Y}_i|\boldsymbol{X}_i)$. After the DNN is fine-tuned using Stochastic Gradient Descent [36], we compute the posterior probability of observing senone l given the speech frame $\boldsymbol{x}$ by doing a complete forward propagation,

$$p(l|\boldsymbol{x}) = \sum_{g \in G} \frac{\exp x_g^{'}}{\sum_k \exp x_k^{'}}, \qquad (1)$$

where $x_k^{'}$ is the output of the hidden layer that precedes the softmax layer, and G is the set of softmax outputs that are



(a) AC-PPG (proposed): phonetic similarity

(b) AC-SIM (baseline 1): acoustic similarity

(c) AC-DTW (baseline 2): time-alignment

Fig. 3. L1: native, L2: non-native. (a) AC-PPG: proposed AC algorithm that uses phonetic similarity. (b) AC-SIM: Baseline 1 that uses acoustic similarity through VTLN to pair frames [12]. (c) AC-DTW: Baseline 2; native and non-native frames are time-aligned following their ordering in the data.

grouped into senone l during the "mixing-up" procedure. A PPG frame of $\boldsymbol{x}$ is constructed by forming a vector from all possible values of $p(l|\boldsymbol{x})$, see eq. (2).

B. Frame pairing

Conventional voice conversion methods use time alignment to pair frames from source and target utterances. As such, a VC model trained from time-aligned frame pairs will retain the non-native speaker's accent. Instead, to perform accent conversion, the pairing must be based on the phonetic similarity between source and target frames. In this way, each native speech frame is associated with its most similar non-native counterpart in terms of pronunciation. If we train a spectral conversion model between these frame pairs, the pronunciation from the native speech data will be preserved and the spectral envelope of the native speaker will be modified to match the non-native speaker's voice quality.

1) Frame pairing based on phonetic similarity (AC-PPG): We use PPGs to pair frames between the native and the non-native speaker. Our rationale is straightforward: if an ASR trained on native speech determines that a non-native speech segment $\boldsymbol{y}$ is close to the native speech production of a particular phoneme (or triphone, in our case), then it is reasonable to pair $\boldsymbol{y}$ with a native speech segment $\boldsymbol{x}$ with the same phonetic label; see Fig. 3 (a). Specifically, our approach works as follows. In a first step, we compute PPG frames for speech frames from the two speakers,

$$\mathcal{L}_{\boldsymbol{x}_i} = [p(l_1|\boldsymbol{x}_i), p(l_2|\boldsymbol{x}_i), \ldots, p(l_V|\boldsymbol{x}_i)], \tag{2}$$

where $\boldsymbol{x}_i$ is the acoustic feature vector of the i-th speech frame; $V = \{l_1, l_2, \ldots, l_V\}$ is the predefined senone set; $p(l_j|\boldsymbol{x}_i)$ is the conditional probability that the speech frame belongs to senone l_j given $\boldsymbol{x}_i$; $\sum_j p(l_j|\boldsymbol{x}_i) = 1$.

Given posterior feature vectors $\mathcal{L}_{\boldsymbol{x}_i}$ and $\mathcal{L}_{\boldsymbol{x}_j}$, we calculate their distance using the symmetric KL divergence,

$$D(\mathcal{L}_{\boldsymbol{x}_i}, \mathcal{L}_{\boldsymbol{x}_j}) = (\mathcal{L}_{\boldsymbol{x}_i} - \mathcal{L}_{\boldsymbol{x}_j}) \cdot (\log \mathcal{L}_{\boldsymbol{x}_i} - \log \mathcal{L}_{\boldsymbol{x}_j}). \tag{3}$$

The symmetric KL divergence [15] is commonly used to compute the similarity between distributions, and here, each frame of the PPG functions like a distribution. For each source (i.e., native) frame $\boldsymbol{x}_i$ we find its closest target (i.e., non-native) frame $\boldsymbol{y}_i^*$,

$$\boldsymbol{y}_i^* = \arg\min_{\forall \boldsymbol{y}} D(\mathcal{L}_{\boldsymbol{x}_i}, \mathcal{L}_{\boldsymbol{y}}). \tag{4}$$

Likewise, for each non-native frame $\boldsymbol{y}_i$ we find its closest native frame $\boldsymbol{x}_i^*$,

$$\boldsymbol{x}_i^* = \arg\min_{\forall \boldsymbol{x}} D(\mathcal{L}_{\boldsymbol{x}}, \mathcal{L}_{\boldsymbol{y}_i}). \tag{5}$$

Each frame pairing process only involves two speakers – the given native and non-native speakers. The frame pairing does not constrain the search space. Therefore, it is possible to pair multiple frames from one speaker with the same frame from the other speaker. In this case, we duplicate that frame multiple times. The resulting frame pairs are used to train a Gaussian Mixture Model (GMM).

2) Baseline methods for frame pairing: We compared the proposed PPG-based method against two baseline techniques for frame pairing: the acoustic similarity method of Aryal and Gutierrez-Osuna [12], and dynamic time warping.

Baseline 1 *(AC-SIM).* Following [12], we measured acoustic similarity as the inverse of the L2-norm between native and non-native speaker frames, after normalizing the native speaker to match the vocal tract length of the non-native speaker; see Fig. 3 (b).

In a first step, we learn a VTLN transform to reduce physiological differences in vocal tract between the two speakers. For this purpose, we time-align parallel training utterances of the two speakers, each utterance represented as a sequence of MFCCs. Following Panchapagesan and Alwan [37], we then learn a linear transform between the MFCCs of both speakers using ridge regression:

$$T^* = \arg\min_{T} \|\boldsymbol{x} - T\boldsymbol{y}\|^2 + \lambda\|T\|^2, \tag{6}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors of MFCCs from the native and non-native speakers, respectively, and T^* is the VTLN transform. Next, for each native vector $\boldsymbol{x}_i$ we find its closest non-native vector $\boldsymbol{y}_j^*$ as:

$$\boldsymbol{y}_j^* = \arg\min_{\forall \boldsymbol{y}} \|\boldsymbol{x}_i - T^*\boldsymbol{y}\|^2. \tag{7}$$

We repeat the process for each non-native vector $\boldsymbol{y}_i$ to find its closest match $\boldsymbol{x}_j^*$:

$$\boldsymbol{x}_j^* = \underset{\forall \boldsymbol{x}}{\arg\min} \, \|\boldsymbol{x} - T^* \boldsymbol{y}_i\|^2. \tag{8}$$

The above process results in a lookup table where each native and non-native frame in the database is paired with the closest one from the other speaker.

Baseline 2 *(AC-DTW).* As our second baseline method, we use Dynamic Time Warping (DTW) [38] to time-align native and non-native frames, as illustrated in Fig. 3 (c).

We note that baselines 1 and 2 need parallel data for training, whereas the proposed method can operate on non-parallel data, as we shall see in Section V-C.

C. Spectral conversion

To ensure a fair comparison between the three frame-pairing methods, we use a common spectral conversion technique to map a native source speaker's spectral features to match a non-native target speaker's voice quality. Following Toda *et al.* [8], we use a GMM to model the joint distribution of source and target frame pairs, and then use maximum likelihood parameter generation (MLPG) with global variance (GV) [39] to generate the converted speech for a given source utterance. Specifically, we use 2D-dimensional acoustic features, $X_t = [x_t^\top, \Delta x_t^\top]^\top$ from the source speaker, and $Y_t = [y_t^\top, \Delta y_t^\top]^\top$ from the target speaker, consisting of D-dimensional static and dynamic features, where $(\cdot)^\top$ denotes the transpose. Given the paired source and target features, we train a GMM to model the joint probability density $p(X, Y|\theta)$ where θ denotes model parameters, estimated using Expectation-Maximization (EM):

$$\theta = \text{EM}(\underset{\theta}{\arg\max}\, p(X, Y|\theta)). \tag{9}$$

When converting source static and dynamic feature vectors $X = [X_1^\top, X_2^\top, \ldots, X_T^\top]^\top$ to the target static feature vectors $y = [y_1^\top, y_2^\top, \ldots, y_T^\top]^\top$ – after the GMM is trained, we maximize the function below with respect to y,

$$\hat{y} = \underset{y}{\arg\max} \log \left(p(Y|X, \theta)^\omega p(\nu(y)|\theta_\nu) \right), \;\; Y = Wy, \tag{10}$$

where $p(Y|X, \theta)$ denotes the conditional probability density function (PDF) on the target static and dynamic feature vectors, and $p(\nu(y)|\theta_\nu)$ represents the likelihood of a PDF on the global variance of the target feature vectors, which is represented as a separate GMM (one mixture) and trained using the EM algorithm as well. W is a matrix that appends dynamic features to the static features, and ω adjusts the relative importance between the two distributions and is set as the ratio of number of dimensions between vectors $\nu(y)$ and Y $(= 1/2T)$. We use a GMM instead of a DNN in this study to focus on low-resource accent conversion scenarios – in real pronunciation training applications, we generally have a limited amount of data from the non-native speakers.

TABLE I
DEMOGRAPHIC INFORMATION OF THE SPEAKERS

Speaker	Gender	Native Language	English Proficiency
BDL	M	English	Native
CLB	F	English	Native
RRBI	M	Hindi	91
TNI	F	Hindi	99
HKK	M	Korean	114
YKWK	M	Korean	N/A
ABA	M	Arabic	94-101

D. Pitch scaling

Previous studies [7], [17], [21] have shown that prosody modification is an essential part of accent conversion, and the pitch contour contains identity-related information. Since pitch modification is not the focus of this study, we follow the standard procedure [8] and use the pitch trajectory from the source (native) speaker, which captures native intonation patterns, then normalize it to match the pitch range of the target (non-native) speaker using mean and variance normalization in the $\log F_0$ space.

IV. EXPERIMENTAL SETUP

A. DNN acoustic model for extracting PPG

To train the DNN acoustic model, we used Kaldi's Librispeech recipe[1]. The model is a p-norm DNN ($p = 2$), as introduced in the method section, with five hidden layers. We extracted 13-dim MFCC vectors with a 7-frame context, passed the concatenated 91-dim (13×7) MFCCs through a Linear Discriminant Analysis (LDA) to generate a 40-dim input feature vector, then concatenated nine frames of such 40-dim LDA features as the final input to the DNN. The 360-dim (40×9) input features were de-correlated using a fixed linear transform. All hidden layers had 5,000 hidden neurons and output 500 activations because each p-norm non-linearity was computed over ten hidden neurons. Every hidden layer was fully-connected with the previous layer. Right after the last hidden layer was a softmax layer of 14,000 nodes; those nodes were then "group-summed" to produce the final output across senones (5,816 dimensions, which were obtained from state-tying on a phonetic decision tree built from the transcripts of the training data; see [40] for more details on how the decision tree was constructed). The DNN acoustic model was trained on Librispeech's [41] training set, a speech recognition corpus that contains 960 hours of native English speech, the majority being American English. In the following experiments, the Librispeech corpus was used solely for building the acoustic model.

B. Speech corpus for accent conversion

For the native speech synthesis corpus, we used two speakers from the CMU ARCTIC dataset [42]: BDL and CLB. Those recordings have a sampling rate of 16 KHz. For the non-native (L2) English speech synthesis corpus, we used five non-native speakers from the L2-ARCTIC corpus[2] [43]:

[1] https://github.com/kaldi-asr/kaldi/tree/master/egs/librispeech

[2] https://psi.engr.tamu.edu/l2-arctic-corpus/

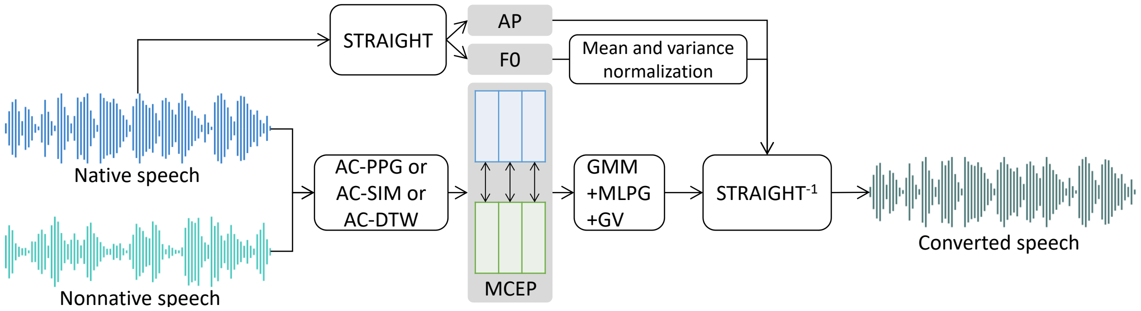


Fig. 4. Accent conversion workflow; frame pairing can be AC-PPG, AC-SIM (baseline 1), or AC-DTW (baseline 2).

two native Hindi speakers, two native Korean speakers; and one native Arabic speaker. Each non-native speaker produced the full ARCTIC dataset (∼1100 utterances; around one hour of speech). The speech was recorded in a quiet room at 44.1 KHz. For the following experiments, we down-sampled all the non-native speech data to 16 KHz using sox[3]. The speaker demographic information is summarized in Table I. For the non-native speakers, their English proficiency level was measured in their TOEFL iBT scores[4] [46].

C. System configuration

In what follows, we will refer to the proposed frame-pairing algorithm, baseline 1 (acoustic similarity), and baseline 2 (dynamic time warping) as **AC-PPG**, **AC-SIM**, and **AC-DTW**, respectively.

We used the TANDEM-STRAIGHT vocoder[5] [49] to decompose speech into aperiodicity (AP), F_0, and a 513-dim spectral envelope. Then, we computed 25-dim MFCCs[6] from the spectral envelopes to learn the VTLN transform and pair frames using acoustic similarity (AC-SIM); see section III-B2. AC-DTW also used those MFCCs (excluding MFCC$_0$) to time-align a source speaker to a target speaker. AC-PPG used the 5816-dim PPGs extracted by the acoustic model to perform frame pairing.

We also computed 25-dim MCEPs from the spectral envelopes as the acoustic feature (excluding MCEP$_0$ since it is energy) to train the spectral conversion models (GMMs) and convert speech from the native speaker to the non-native speaker. MCEPs from the two speakers were frame paired using the three methods (AC-PPG, AC-SIM, AC-DTW) before being fed to the GMMs. Following Aryal and Gutierrez-Osuna [12], all GMMs had 128 mixture components with diagonal covariance matrices. Input features to the GMM include delta features, and therefore the joint feature vectors had 96 dimensions. Once we converted the native speaker's MCEPs to the non-native speaker's space, we reconstructed the spectrogram from the converted MCEPs (MCEP$_0$ being copied from the native speaker), and combined it with the native speaker's AP and normalized F_0 to synthesize speech using the TANDEM-STRAIGHT vocoder. The conversion pipeline is illustrated in Fig. 4.

All experiments were conducted on a desktop running Windows 10 with an Intel Core i7-7700K CPU@4.2GHz, 16GB of memory, and an NVidia GTX 1070 GPU. Most of the algorithms were implemented and run on Matlab v9.3, except for the acoustic model and PPGs, which were computed using Kaldi on Ubuntu 16.04.

V. Results

We conducted three sets of perceptual listening studies to evaluate different properties of the proposed frame-pairing algorithm. In the first experiment, we compared the approach against the two baseline systems by its ability to reduce perceived accents while matching the voice quality of the non-native speakers. In the second experiment, we evaluated whether the approach could also be used for the reverse purpose, i.e., to impart a non-native accent to a native speaker's voice. In the third and final experiment, we evaluated the approach to perform accent conversion using non-parallel speech corpora.

We recruited anonymous human participants from Amazon's Mechanical Turk platform[7] for our listening tests. Following Buchholz and Latorre [50], all listening tests included calibration trials designed to be easy to judge, and we used the participants' responses on those calibration trials to detect cheating behaviors. We excluded data from participants whose responses were below chance level on those calibration questions. All participants' calibration responses were excluded from the final analyses. In addition, and following [16], all human subjects passed a screening test that consisted of identifying various American English accents. We compensated participants for their time at an hourly rate of eight USD. In all experiments, the reference native and non-native English speech were resynthesized from their MCEPs using TANDEM-STRAIGHT to keep their acoustic quality comparable with the converted speech, which went through the same vocoder compression. When selecting testing samples,

[3] http://sox.sourceforge.net/Main/HomePage

[4] Speaker ABA only reported his IELTS [44] score (7.0). We converted it to a TOEFL iBT score following [45].

[5] We used the NDF F_0 extractor [47] instead of the default F_0 extractor that comes with TANDEM-STRAIGHT, because based on our experience and a prior study [48], the NDF F_0 extractor is more robust than the TANDEM-STRAIGHT default.

[6] We only used those MFCCs to generate the frame pairing lookup tables in **AC-SIM** and **AC-DTW** and discarded in other tasks.

[7] https://www.mturk.com/

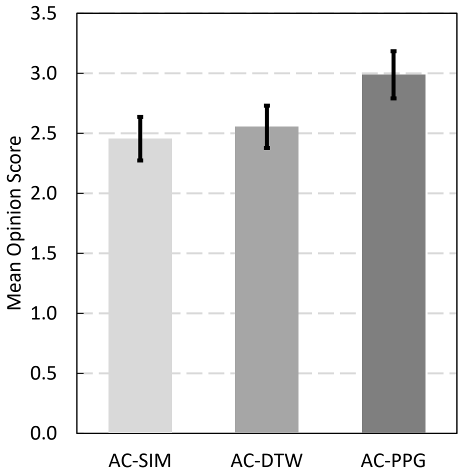


Fig. 5. Mean Opinion Scores for the proposed method (AC-PPG) and the two baseline methods (AC-SIM, AC-DTW); the error bars show 95% confidence intervals.

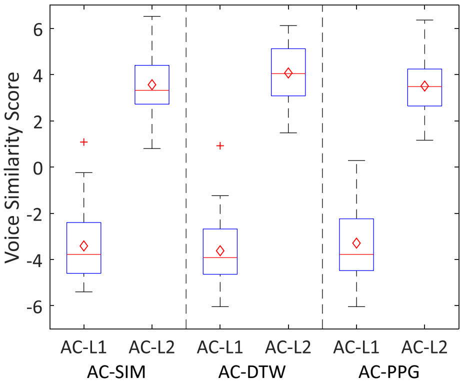


Fig. 6. Voice quality results; AC-L1: VSS between AC and native (L1) speaker; AC-L2: VSS between AC and non-native (L2) speaker; the middle bars in the boxes show the median values and diamond markers (◇) show the mean values, the plus signs (+) indicate outliers, those notations apply to all boxplots in this paper.

we always randomly draw from the available pools, i.e., we did not cherry-pick the audio clips. All test trials were randomly presented. For any listening tests that required pairwise comparisons, the presentation order within an utterance pair was counterbalanced. Unless otherwise noted, we used paired-sample t-tests for the analyses.

A. *Experiment 1: Comparing AC-PPG against baselines*

In this experiment, we considered five native to non-native speaker pairings for accent conversion: BDL to RRBI, BDL to HKK, BDL to YKWK, BDL to ABA, and CLB to TNI. For each speaker pair, we used 100 parallel utterances for training and 50 utterances for testing; there was no overlap between the two sets. We performed accent conversion on all 50 test utterances using models trained on each of the three frame-pairing algorithms, i.e., AC-PPG, AC-SIM, and AC-DTW.

Acoustic quality. We used a standard five-point (1-Bad, 2-Poor, 3-Fair, 4-Good, 5-Excellent) Mean Opinion Score (MOS) to rate the acoustic quality of the synthesized speech. Thirty listeners rated 150 test samples: 50 per system, 10 utterances per conversion direction. Results are shown in Fig. 5. The proposed method (AC-PPG) received a MOS rating of 2.99, which was significantly higher than either baseline: AC-SIM (2.45 MOS, 22% relative improvement; $t(29) = 15.61$, $p \ll 0.001$; one-tail) and AC-DTW (2.55 MOS, 17% relative improvement; $t(29) = 12.04$, $p \ll 0.001$; one-tail). These results suggest that the proposed algorithm can boost the acoustic quality of the converted speech using exactly the same training data without even having to modify the GMM training and spectral conversion methods.

Voice quality. Following our prior work [32], we used a voice similarity score (VSS) ranging from -7 (definitely different speakers) to +7 (definitely same speaker) to assess the speaker's voice quality. Twenty-six participants rated 150 utterance pairs: 50 pairs per system (25 AC-L1 and 25 AC-L2 pairs, each pair contained one AC and one L1 [native]/L2 [non-native] utterance), and ten pairs per conversion direction. Following Felps *et al.* [7], we played utterances in reverse to prevent the accent from interfering with the perception of voice quality. In each trial, listeners first answered whether both utterances were produced by the same speaker (+1) or different speakers (-1), and then rated their confidence level on a 7-point scale (1-Not at all confident, 7-Extremely confident). The VSS was then compiled by multiplying the response from the first question with the confidence rating. Results are summarized in Fig. 6. Overall, the three systems have similar VSS, and AC-L1 pairs received an average VSS between -3.29 to -3.62, indicating that listeners were "*confident*" that the AC utterances had a different voice quality from those of the native speaker. Likewise, AC-L2 pairs received an average VSS between 3.50 to 4.07, indicating that listeners were "*confident*" that the same speaker produced the AC and L2 utterances. When analyzing the AC-L1 pairs, we found no significant differences in VSS between AC-PPG and either baseline (AC-PPG:AC-SIM $t(25) = 1.13$, $p = 0.27$; AC-PPG:AC-DTW, $t(25) = 1.95$, $p = 0.06$; two-tail). These results suggest that the three methods are equivalent in terms of producing speech that is different from the native speaker. When analyzing AC-L2 pairs, we found no significant difference between AC-PPG and AC-SIM ($t(25) = 0.42$, $p = 0.68$, two-tail), suggesting that the new accent conversion algorithm did not sacrifice the speaker's voice quality. However, AC-DTW achieved a higher VSS (4.07) than AC-PPG (3.50); one-tail t-test ($t(25) = 3.59$, $p \ll 0.05$). One possible explanation for this result is that listeners still picked up subtle cues of non-native accent in the AC-DTW speech samples, and used it to rate voice quality. Because AC-DTW only performs voice conversion, it retains some of the non-native speaker's accent. This residual non-native accent may have led listeners to rate samples from AC-DTW as more similar to the non-native speech, even though the recordings were played backwards. This explanation is consistent with prior studies [51], [52] showing that, even when speech is played backwards, native English speakers can still detect non-native English accents.

Non-native accentedness. We used a preference test to

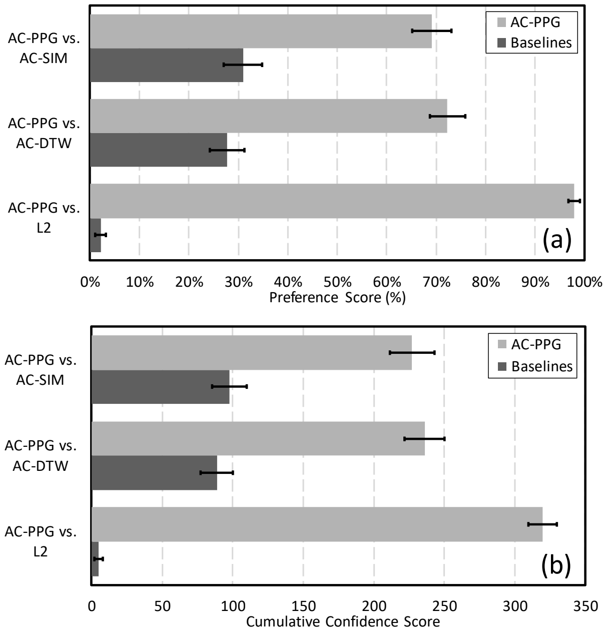


Fig. 7. (a) Accent preference score with 95% confidence interval. (b) Cumulative confidence score for accentedness with 95% confidence interval.

determine if AC-PPG does indeed make the converted speech sound more native-like. Thirty native English speakers rated 150 utterance pairs: 50 pairs for each comparison: AC-PPG vs. AC-SIM, AC-PPG vs. AC-DTW, and AC-PPG vs. L2 (i.e., original utterances from the non-native speaker), ten pairs of utterances per conversion direction, each utterance pair was from the same sentence. Listeners were asked to choose the most native-like (least foreign) utterance from each pair, and then rate their confidence level using a seven-point scale (1-Not confident at all, 7-Very confident). Aryal and Gutierrez [12] had previously established that AC-SIM outperforms AC-DTW and L2 in this task; therefore, we omitted those comparisons in this study.

In a first analysis, we sought to determine if a particular system was preferred as "less-accented" and compared the *preference ratings* from the participants. Results are summarized in Fig. 7 (a). On average, listeners were very confident (mean: 98%, STD: 3%) that the AC-PPG conversions were more native-like than the original non-native utterances. More importantly, listeners were positive that AC-PPG outperformed both AC-SIM (mean: 69%, STD: 11%) and AC-DTW (mean: 72%, STD: 10%). All the above preference scores are statistically significant ($p \ll 0.001$; one-tail) compared with chance levels (50%). Since preference tests sometimes are too coarse and will mask out nuances in raters' attitudes, we further used the *confidence ratings* to compute a more detailed measurement – the cumulative confidence score (CCS) [53]. The CCS for each system in each comparison pair was computed as follows. We treat each response as if it were assigning a number of points to a system; for example, if a listener preferred the AC-PPG system and was "somewhat confident" (rated as three), then the AC-PPG system would receive three points. We then computed the average CCS that listeners allocated to each system. Therefore, the highest score a system can get is 350 points (7×50), within a comparison pair. Results are summarized in Fig. 7 (b). As shown, all comparison pairs have the same trend as in the preference test, with AC-PPG performing significantly better than both baselines. All differences in CCS were statistically significant ($p \ll 0.001$, one-tail).

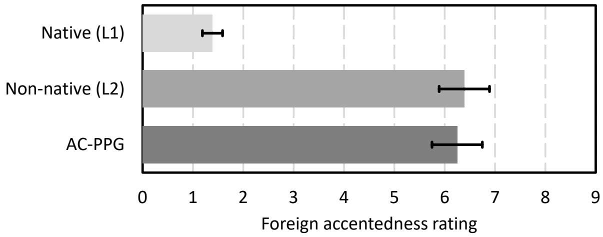


Fig. 8. Foreign accentedness ratings for L1 (native English), L2 (non-native English), and AC speech; the error bars show 95% confidence intervals.

B. Experiment 2: Native to non-native conversion

In a second experiment, we evaluated whether AC-PPG can perform the accent conversion task in the opposite direction – creating a voice that has the native speaker's voice quality, but speaking with a non-native accent. Prior work [54] has tackled this problem from a Text-To-Speech perspective, so we wanted to determine if it could also be achieved through accent conversion. Accordingly, for this experiment we performed accent conversion in five directions that were from non-native to native English speakers, i.e., RRBI to BDL, HKK to BDL, YKWK to BDL, ABA to BDL, and TNI to CLB. The training and testing data for all speakers was identical as that used for *Experiment 1*.

In an initial listening test, we recruited 20 subjects to rate the non-native English accent of the converted speech using a nine-point Likert-scale rating test [2], where 1 corresponded to "no accent" and 9 to "very strong accent." For each conversion direction, we randomly picked five utterances, and we made sure that the final 25 (5×5) utterances for evaluation were from different elicitation sentences. To provide a reference, we also included the same set of sentences that were uttered by the native and non-native speaker in the test. Therefore, all listeners rated 75 (25×3) sentences. Given that our native speakers (BDL and CLB) spoke American English, before the test, we instructed listeners to consider that "*A 'foreign accent' is defined as an accent that is different from the General American English accent*." We also provided two samples of American accent English that were produced by native speakers not used in this study. All listeners were geographically located in the United States and all but one listeners self-reported to be native English speakers. The only listener whose native language is not English is a native Italian speaker who also speaks English and French, and since this participant passed our American accent pretest, we did not exclude this participant's responses. Results are summarized in Fig. 8. On average, listeners rated the native speech to be

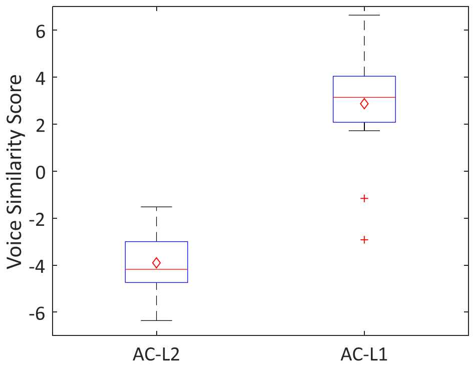


Fig. 9. Voice similarity score for AC-L1 and AC-L2 comparisons.

1.4 points (closer to "no accent") and the non-native speech to be 6.4 points (closer to "very strong accent"). The accent-converted speech had an average rating of 6.2 points (closer to "very strong accent"), which was similar to the ratings of the non-native speech. No significant difference was found between accentedness ratings of non-native and AC speech ($t(19) = 0.82$, $p = 0.42$, two-tail). Therefore, this experiment indicates that our accent conversion approach was able to impart the non-native accent of the non-native speaker to utterances from a native speaker.

In a second listening test, we focused on evaluating whether the converted speech retained the voice quality of the original (native) speaker. Accordingly, we used the same VSS test as in *Experiment 1* to produce voice similarity scores between AC sentences and the original native/non-native sentences. Twenty listeners rated 50 utterance pairs, among which 25 were AC-L1, and the rest were AC-L2 pairs. As before, we randomized all presentation order and played the recordings in reverse. Results are summarized in Fig. 9. Listeners were "confident" that AC utterances had the voice quality of the native speakers (mean AC-L1 VSS score 2.87), and was different from the non-native speaker (mean AC-L2 VSS score -3.90) despite that they share the same accent. Considering the results from both listening tests in this experiment, we can conclude that AC-PPG is able to impart a non-native accent to native voices.

C. Experiment 3: AC-PPG using non-parallel training data

Our method does not impose timing constraints when pairing native and non-native speech frames: an acoustic frame from the native speaker is paired with a frame in the non-native speaker's training set by minimizing the symmetric KL divergence between their respective PPGs. Thus, in principle, our method removes the constraint that native and non-native speakers must produce the same set of utterances. This property is particularly useful for real-world applications because it allows more flexibility when recording training sentences. Therefore, in a third and final experiment we evaluated the AC performance by comparing two variants of our method:

- **AC-PPG-P**: the same system used in *Experiment 1,* i.e., using parallel sentences as the training data;
- **AC-PPG-NP**: a system that used non-parallel sentences. For this purpose, we randomly selected 100 native training utterances that were different from those in the non-native training or non-native test sentences. As a result, the native and non-native speakers never uttered any common sentence. All other configurations for this system were the same as AC-PPG-P.

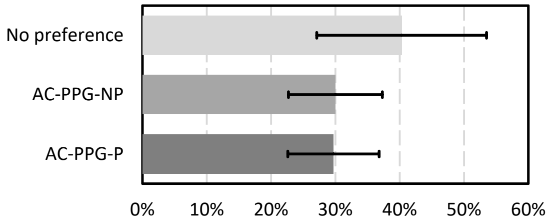


Fig. 10. Preference scores for comparing the acoustic quality of AC-PPG-P and AC-PPG-NP; the error bars display the 95% confidence intervals.

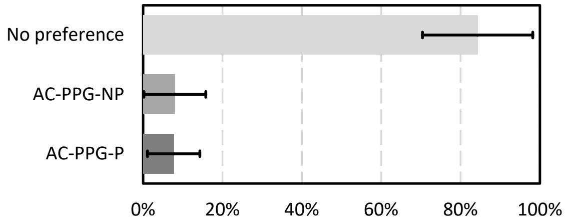


Fig. 11. Preference scores for comparing foreign accentedness of AC-PPG-P and AC-PPG-NP; the error bars display the 95% confidence intervals.

The AC directions and test sentences were the same as those used in *Experiment 1*. For each system, we generated accent converted sentences from all 50 testing samples for evaluation.

In a first listening study, we used a preference test to determine which system yielded better acoustic quality. Twenty participants rated 50 utterance pairs – one from AC-PPG-P and the other from AC-PPG-NP, both utterances having the same linguistic content. We randomly selected 10 utterance pairs from each AC direction. For each pair, participants were asked to pick the utterance that has the best acoustic quality. The test allowed them to choose "no preference" as their response. Results are summarized in Fig. 10. The majority of the votes (40.3%) reflected no difference between the acoustic quality of the two systems ("no preference"), and both systems received a similar percentage of votes (29.7% for AC-PPG-P; 30.0% for AC-PPG-NP). We found no significant difference in terms of acoustic quality between using parallel or non-parallel data ($t(19) = 0.11$, $p = 0.91$, two-tail).

In a second listening test, we investigated whether using non-parallel data would affect the non-native ratings of the converted speech. The experimental protocol was the same as the one we used in the acoustic quality experiment, except that in this case, for each AC-PPG-P and AC-PPG-NP utterance pair, we asked participants to select the one that had the "least foreign accent." Twenty participants rated 50 utterance pairs, 10 pairs for each AC direction. Results are summarized in Fig. 11. The vast majority of the votes (84.3%) indicated that there was no difference between the two systems. Furthermore, a t-test on the preference scores for AC-PPG-P (mean 7.7%) and

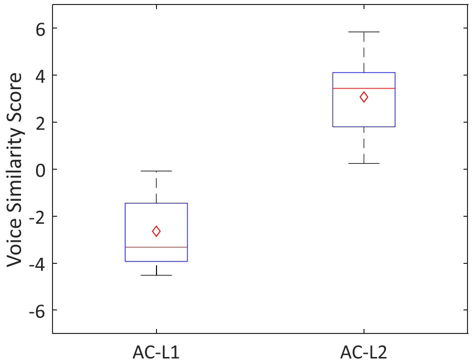


Fig. 12. Voice similarity scores for AC-PPG-NP.

AC-PPG-NP (mean 8.0%) revealed no significant differences ($t(19) = 0.17$, $p = 0.86$, two-tail).

Finally, we asked 21 listeners to rate AC-PPG-NP sentences in terms of voice quality. Each listener rated 50 converted utterances, where we randomly selected 10 utterances from all 5 conversion directions. The VSS scores are summarized in Fig. 12. The average AC-L1 VSS is -2.64 (std: 1.52), and 3.07 (std: 1.48) for AC-L2. Using a two-tail independent samples t-test assuming unequal variances[8], we found no significant difference between the average VSS for the AC-PPG system in Fig. 6 and those in Fig. 12. For AC-L1, the test gave $t(43) = 1.44$, $p = 0.16$. For AC-L2, the test yielded $t(40) = 1.06$, $p = 0.29$. Thus, this experiment verified that using non-parallel data still allows our frame-pairing technique to preserve the non-native speaker's voice quality in the converted speech.

VI. Discussion

In prior work [12], Aryal and Gutierrez-Osuna had shown that paring speech frames based on acoustic similarity (i.e., the AC-SIM baseline in our study), and then using the resulting frame pairs to train a voice conversion model could be used to create a voice that captured a native speaker's pronunciation and a non-native speaker's voice quality. Their method was able to achieve significantly better accentedness rating compared with pairing frames using DTW, though the results were based on a single pair of speakers. During our internal evaluations (results not shown) with multiple pairs of speakers and several set of non-native accents, we found that the speech generated by AC-SIM still contained noticeable mispronunciations. Since AC-SIM normalizes the vocal tract length difference between native and non-native speakers, we hypothesized that there remains a lot of other unattended speaker-dependent (SD) information in the VTLN-transformed acoustic feature space, which makes the resulting frame pairing not ideal. PPGs, on the other hand, are produced by speaker-independent (SI) acoustic models built for ASR. As a result, the most dominant information in PPGs is linguistic information. These analyses reinforced our intuition to use AC-PPG to eliminate the effects of SD cues in the frame pairing process.

The listening tests in *Experiment 1* show that the proposed frame pairing method can significantly reduce the non-native accent ratings compared with two baselines. In terms of *voice similarity* between the non-native speaker and the converted speech, AC-PPG performs as well as AC-SIM. Although the speech generated by AC-DTW was rated more similar to the non-native speaker than AC-PPG, we suspected that it is hard to decouple the influence of *accent* and *voice quality* on the perceived *speaker identity* (refer to Section I for difference between *voice quality* and *speaker identity*). Listeners may have used the remaining foreign accent in the AC-DTW utterances to select the *speaker identity* of the utterances instead of their *voice quality*. Therefore, an interesting future direction would be to design a new perceptual experiment protocol that can better decouple *voice quality* and *accent* in spoken sentences, compared with the current solution of playing audio in reverse.

Another interesting observation from *Experiment 1* is that despite using the same spectral conversion model as the two baseline systems, AC-PPG can significantly boost the acoustic quality of the synthesis. When comparing the speech syntheses from AC-PPG with the others, we did notice that there were fewer noises and artifacts. One possible explanation for this is that AC-PPG pairs frames with similar phonetic context. Therefore, frame pairs have similar spectral structures, making the statistical regression model for spectra estimation less likely to introduce odd shapes in the predicted spectral envelopes. Consequently, better spectral predictions lead to better synthesis quality. Future work could investigate if this property of AC-PPG generalizes to other statistical conversion models that take frame pairs as training input (e.g., deep neural networks [10], [29], direct waveform modification [9]).

Experiments 2 and *3* investigated other interesting aspects of the proposed frame-pairing method. *Experiment 2* verified that AC-PPG could also work in the opposite conversion direction – creating an artificial voice that has a native speaker's voice quality while speaking in a foreign accent. This artificial voice can be useful for generating materials for perceptual studies. For example, it can map speech from speakers that have different accents to the same voice quality, therefore removing the impact of voice quality when comparing differences in accents. *Experiment 3* verified that we could use non-parallel dataset to achieve the same accent conversion performance (measured in acoustic quality, accentedness, and voice quality) using AC-PPG. One possible reason why we could use non-parallel training data is that AC-PPG looks at a fine-grained context (95 ms in the current implementation)[9], and this context size is comparable with the duration of a vowel [55] or consonant [56] segment in American English. Therefore, as long as the two sets of training data from native and non-native

[8] The two groups we are comparing have 26 (AC-PPG in Fig. 6) and 21 (AC-PPG-NP) subjects respectively, therefore, it is not reasonable to assume that they have the same variance.

[9] Each frame of PPG feature looks at a larger context than the analysis window (25 ms), because the input to the acoustic model consists of nine frames of adjacent LDA feature, and each frame was computed from seven consecutive MFCC feature vectors (25 ms). Therefore, the total context for a frame of PPG feature is 9+7-1=15 consecutive analysis windows, which converts to 95 ms under a 5 ms window shift.

speakers have a balanced phonetic distribution, the approach is indifferent to the actual word-level prompts. The non-parallel data constraint is much more relaxed than the widely used parallel constraint, making the proposed method applicable to real-world scenarios, where parallel data are scarce or tedious to obtain.

AC-PPG can run efficiently with careful optimization and GPU-based parallelization. In our experiments, it generally took no more than two minutes to compute the pairing between 100 training utterances ($\sim$5 minutes of speech) from the native and non-native speakers. Further reductions in computation time may be achieved via dimensionality reduction and clustering.

At present, our ratings of acoustic quality are on the low end of what state-of-the-art voice conversion systems can achieve [57]. This is largely due to the choice of voice conversion system used, i.e., a conventional GMM-based spectral conversion system as a case study, which was needed to ensure a fair comparison with our previous work [12]. Fortunately, our frame-pairing approach can be combined with other spectral conversion methods to produce higher quality speech synthesis. For example, instead of converting speech frame-by-frame, we could perform the conversion over a larger context (e.g., sequence to sequence conversion [58].) Using a larger conversion context is likely to increase the acoustic quality [59], [60]. More importantly, mispronunciations often occur at the segment level, which is beyond the scope of frame-level conversion, and contextual information has to be taken into consideration to accurately correct segmental pronunciation errors in accent conversion.

Another line of ongoing work in our group is to relax the non-parallel data constraint further to allow the use of cross-lingual training data. In preliminary experiments (not shown here), we successfully performed accent conversion using utterances recorded in the target speaker's native language to capture their voice quality[10].

VII. Conclusion

We have proposed a new frame-pairing method based on the phonetic similarity between acoustic frames. To measure phonetic similarity, we map source and target frames into a phonetic posteriorgram space using speaker-independent acoustic models trained on a native English corpus. Through a series of perceptual studies, we have shown that merely changing the frame pairing method can lead to significant improvement in acoustic quality and "nativeness," while keeping the voice quality of the non-native speaker. Our results also show that the approach works well across multiple non-native speakers with different native tongues. Additionally, the proposed algorithm does not need parallel data for training, which is ideal for real-world applications. Our approach only requires 5-10 minutes of speech data from the non-native speaker, making it practical for pronunciation training in realistic settings [63]. The implementation of the proposed system can be found at `https://github.com/guanlongzhao/ppg-gmm`.

VIII. Acknowledgment

This work was supported by NSF awards 1619212 and 1623750. We would like to thank Adam Hair, Shaojin Ding, and Dennis Rodrigo da Cunha Silva for reviewing early versions of the manuscript. We are grateful for the feedback from the reviewers, which helped improve the quality of the manuscript.

[10] In these preliminary experiments, we used native Brazilian Portuguese speakers from the SID dataset [61] as the target speakers. Since Portuguese share some phonological similarities with English [62], we used the acoustic model used in this study directly to produce the PPGs from native Portuguese speech. For future work and more general cases (e.g., languages from the Sino-Tibetan family), we have to include senones from the target speaker's native tongues in the acoustic modeling process.

References

[1] E. H. Lenneberg, "The biological foundations of language," *Hospital Practice*, vol. 2, no. 12, pp. 59–67, 1967.

[2] M. J. Munro and T. M. Derwing, "Foreign accent, comprehensibility, and intelligibility in the speech of second language learners," *Language learning*, vol. 45, no. 1, pp. 73–97, 1995.

[3] D. L. Rubin and K. A. Smith, "Effects of accent, ethnicity, and lecture topic on undergraduates' perceptions of nonnative English-speaking teaching assistants," *International Journal of Intercultural Relations*, vol. 14, no. 3, pp. 337–353, 1990.

[4] S. Lev-Ari and B. Keysar, "Why don't we believe non-native speakers? The influence of accent on credibility," *Journal of Experimental Social Psychology*, vol. 46, no. 6, pp. 1093–1096, 2010.

[5] K. Probst, Y. Ke, and M. Eskenazi, "Enhancing foreign language tutors–in search of the golden speaker," *Speech Communication*, vol. 37, no. 3-4, pp. 161–173, 2002.

[6] M. P. Bissiri, H. R. Pfitzinger, and H. G. Tillmann, "Lexical stress training of German compounds for Italian speakers by means of resynthesis and emphasis," in *Proceedings of the 11th Australian International Conference on Speech Science & Technology*, 2006, pp. 24–29.

[7] D. Felps, H. Bortfeld, and R. Gutierrez-Osuna, "Foreign accent conversion in computer assisted pronunciation training," *Speech Communication*, vol. 51, no. 10, pp. 920–932, 2009.

[8] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[9] K. Kobayashi, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "Statistical singing voice conversion with direct waveform modification based on the spectrum differential," in *Proc. Interspeech*, 2014, pp. 2514–2518.

[10] T. Nakashika, T. Takiguchi, and Y. Ariki, "Voice conversion using RNN pre-trained by recurrent temporal restricted boltzmann machines," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 580–587, 2015.

[11] C. Liberatore, S. Aryal, Z. Wang, S. Polsley, and R. Gutierrez-Osuna, "SABR: Sparse, Anchor-Based Representation of the speech signal," in *Proc. Interspeech*, 2015, pp. 608–612.

[12] S. Aryal and R. Gutierrez-Osuna, "Can voice conversion be used to reduce non-native accents?" in *Proc. ICASSP*, May 2014, pp. 7879–7883.

[13] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[14] T. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *2009 IEEE Workshop on Automatic Speech Recognition Understanding*, Nov 2009, pp. 421–426.

[15] I. J. Taneja, "On generalized information measures and their applications," in *Advances in Electronics and Electron Physics*. Elsevier, 1989, vol. 76, pp. 327–413.

[16] D. Felps, C. Geng, and R. Gutierrez-Osuna, "Foreign accent conversion through concatenative synthesis in the articulatory domain," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 8, pp. 2301–2312, 2012.

[17] M. Huckvale and K. Yanagisawa, "Spoken language conversion with accent morphing," in *Sixth ISCA Workshop on Speech Synthesis*, 2007, pp. 64–70.

[18] S. Aryal, D. Felps, and R. Gutierrez-Osuna, “Foreign accent conversion through voice morphing,” in *Proc. Interspeech*, 2013, pp. 3077–3081.

[19] F.-L. Xie, F. K. Soong, and H. Li, “A KL divergence and DNN-based approach to voice conversion without parallel training sentences,” in *Proc. Interspeech*, 2016, pp. 287–291.

[20] G. Zhao, S. Sonsaat, J. Levis, E. Chukharev-Hudilainen, and R. Gutierrez-Osuna, “Accent conversion using phonetic posteriorgrams,” in *Proc. ICASSP*, April 2018, pp. 5314–5318.

[21] Q. Yan, S. Vaseghi, D. Rentzos, and C.-H. Ho, “Analysis and synthesis of formant spaces of British, Australian, and American accents,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 2, pp. 676–689, 2007.

[22] J. Jügler, F. Zimmerer, J. Trouvain, and B. Möbius, “The perceptual effect of L1 prosody transplantation on L2 speech: The case of French accented German,” in *Proc. Interspeech*, 2016, pp. 67–71.

[23] S. Aryal and R. Gutierrez-Osuna, “Reduction of non-native accents through statistical parametric articulatory synthesis,” *Journal of the Acoustical Society of America*, vol. 137, no. 1, pp. 433–446, 2015.

[24] ——, “Data driven articulatory synthesis with deep neural networks,” *Computer Speech & Language*, vol. 36, pp. 260–273, 2016.

[25] S. H. Mohammadi and A. Kain, “An overview of voice conversion systems,” *Speech Communication*, vol. 88, no. 88, pp. 65–82, 2017.

[26] D. Erro, A. Moreno, and A. Bonafonte, “Voice conversion based on weighted frequency warping,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 922–931, 2010.

[27] E. Godoy, O. Rosec, and T. Chonavel, “Voice conversion using dynamic frequency warping with amplitude scaling, for parallel or nonparallel corpora,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1313–1323, 2012.

[28] T. Nakashika, T. Takiguchi, and Y. Ariki, “Voice conversion using speaker-dependent conditional restricted boltzmann machine,” *Eurasip Journal on Audio, Speech, and Music Processing*, vol. 2015, no. 1, p. 8, 2015.

[29] L. Sun, S. Kang, K. Li, and H. M. Meng, “Voice conversion using deep bidirectional long short-term memory based recurrent neural networks,” in *Proc. ICASSP*, 2015, pp. 4869–4873.

[30] R. Takashima, T. Takiguchi, and Y. Ariki, “Exemplar-based voice conversion in noisy environment,” in *2012 IEEE Spoken Language Technology Workshop (SLT)*, 2012, pp. 313–317.

[31] Z. Wu, E. S. Chng, and H. Li, “Exemplar-based voice conversion using joint nonnegative matrix factorization,” *Multimedia Tools and Applications*, vol. 74, no. 22, pp. 9943–9958, 2015.

[32] G. Zhao and R. Gutierrez-Osuna, “Exemplar selection methods in voice conversion,” in *Proc. ICASSP*, 2017, pp. 5525–5529.

[33] W. Xiong, J. Droppo, X. Huang, F. Seide, M. L. Seltzer, A. Stolcke, D. Yu, and G. Zweig, “Toward human parity in conversational speech recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 12, pp. 2410–2423, 2017.

[34] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, “Improving deep neural network acoustic models using generalized maxout networks,” in *Proc. ICASSP*, 2014, pp. 215–219.

[35] S. P. Rath, D. Povey, K. Veselý, and J. Černocký, “Improved feature processing for deep neural networks,” in *Proc. Interspeech*, 2013, pp. 109–113.

[36] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in *Proceedings of COMPSTAT'2010*, Y. Lechevallier and G. Saporta, Eds. Heidelberg: Physica-Verlag HD, 2010, pp. 177–186.

[37] S. Panchapagesan and A. Alwan, “Frequency warping for VTLN and speaker adaptation by linear transformation of standard MFCC,” *Computer Speech & Language*, vol. 23, no. 1, pp. 42–64, 2009.

[38] D. J. Berndt and J. Clifford, “Using dynamic time warping to find patterns in time series,” in *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, ser. AAAIWS'94. AAAI Press, 1994, pp. 359–370.

[39] T. Toda, A. W. Black, and K. Tokuda, “Spectral conversion based on maximum likelihood estimation considering global variance of converted parameter,” in *Proc. ICASSP*, vol. 1, 2005, pp. 9–12.

[40] S. J. Young, J. J. Odell, and P. C. Woodland, “Tree-based state tying for high accuracy acoustic modelling,” in *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics, 1994, pp. 307–312.

[41] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An ASR corpus based on public domain audio books,” in *Proc. ICASSP*, 2015, pp. 5206–5210.

[42] J. Kominek and A. W. Black, “The CMU Arctic speech databases,” in *Fifth ISCA ITRW on Speech Synthesis*, 2004, pp. 223–224.

[43] G. Zhao, S. Sonsaat, A. Silpachai, I. Lucic, E. Chukharev-Khudilaynen, J. Levis, and R. Gutierrez-Osuna, “L2-ARCTIC: A non-native English speech corpus,” in *Proc. Interspeech*, 2018, pp. 2783–2787.

[44] M. Chalhoub-Deville and C. E. Turner, “What to look for in ESL admission tests: Cambridge Certificate Exams, IELTS, and TOEFL,” *System*, vol. 28, no. 4, pp. 523–539, 2000.

[45] ETS, “Linking TOEFL iBT scores to IELTS scores - a research report,” Educational Testing Service, Tech. Rep., 2010.

[46] Y. Cho and B. Bridgeman, “Relationship of TOEFL iBT scores to academic performance: Some evidence from American universities,” *Language Testing*, vol. 29, no. 3, pp. 421–442, 2012.

[47] H. Kawahara, A. de Cheveign, H. Banno, T. Takahashi, and T. Irino, “Nearly defect-free F0 trajectory extraction for expressive speech modifications based on STRAIGHT,” in *Proc. Interspeech*, 2005, pp. 537–540.

[48] M. Morise, “D4C, a band-aperiodicity estimator for high-quality speech synthesis,” *Speech Communication*, vol. 84, pp. 57–65, 2016.

[49] H. Kawahara, M. Morise, T. Takahashi, R. Nisimura, T. Irino, and H. Banno, “Tandem-STRAIGHT: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, F0, and aperiodicity estimation,” in *Proc. ICASSP*, 2008, pp. 3933–3936.

[50] S. Buchholz and J. Latorre, “Crowdsourcing preference tests, and how to detect cheating,” in *Proc. Interspeech*, 2011, pp. 3053–3056.

[51] M. J. Munro, T. M. Derwing, and C. S. Burgess, “The detection of foreign accent in backwards speech,” in *Proceedings of the 15th International Congress of Phonetic Sciences*, 2010, pp. 535–538.

[52] ——, “Detection of nonnative speaker status from content-masked speech,” *Speech Communication*, vol. 52, pp. 626–637, 2010.

[53] J. Yamagishi, Private Communication, Calgary, Alberta, Canada, April 2018.

[54] G. E. Henter, J. Lorenzo-Trueba, X. Wang, M. Kondo, and J. Yamagishi, “Cyborg speech: Deep multilingual speech synthesis for generating segmental foreign accent with natural prosody,” in *Proc. ICASSP*, April 2018, pp. 4799–4803.

[55] N. Umeda, “Vowel duration in American English,” *Journal of the Acoustical Society of America*, vol. 58, no. 2, pp. 434–445, 1975.

[56] ——, “Consonant duration in American English,” *Journal of the Acoustical Society of America*, vol. 61, no. 3, pp. 846–858, 1977.

[57] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen, and Z. Ling, “The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods,” in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 195–202.

[58] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3104–3112.

[59] T. Kaneko, H. Kameoka, K. Hiramatsu, and K. Kashino, “Sequence-to-sequence voice conversion with similarity metric learned using generative adversarial networks,” in *Proc. Interspeech*, 2017, pp. 1283–1287.

[60] H. Miyoshi, Y. Saito, S. Takamichi, and H. Saruwatari, “Voice conversion using sequence-to-sequence learning of context posterior probabilities,” in *Proc. Interspeech*, 2017, pp. 1268–1272.

[61] I. M. Quintanilha, L. W. P. Biscainho, and S. L. Netto, “Towards an end-to-end speech recognizer for Portuguese using deep neural networks,” *XXXV Simpósio Brasileiro de Telecomunicações e Processamento de Sinais, Sao Pedro, Brazil*, 2017.

[62] M. M. Azevedo, “A contrastive phonology of Portuguese and English,” *The Modern Language Journal*, vol. 66, no. 2, p. 222, 1982.

[63] S. Ding, C. Liberatore, G. Zhao, S. Sonsaat, E. Chukharev-Hudilainen, J. Levis, and R. Gutierrez-Osuna, “Golden speaker builder: an interactive online tool for L2 learners to build pronunciation models,” in *Proc. Pronunciation in Second Language Learning and Teaching (PSLLT)*, 2017, pp. 25–26.