INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

An exemplar selection algorithm for native-nonnative voice conversion

Christopher Liberatore, Ricardo Gutierrez-Osuna

Texas A&M University, College Station, Texas, USA
cliberatore@tamu.edu, rgutier@cse.tamu.com

Abstract

We present an algorithm for selecting exemplars for native-to-nonnative voice conversion (VC) using a Sparse, Anchor-Based Representation of speech (SABR). The algorithm uses phoneme labels and clustering to learn optimal exemplars when source and target speakers are affected by poor time alignment, as is common in in native-to-nonnative voice conversion. We evaluate the method on speech from the ARCTIC and L2-ARCTIC corpora and compare it to a baseline exemplar-based VC algorithm. The proposed algorithm significantly improves synthesis quality and more than doubles that of a baseline exemplar-based VC system while using two orders of magnitude fewer atoms. Additionally, the proposed algorithm significantly reduces the VC error and improves the synthesis quality as compared to unoptimized SABR models. We discuss the implications of both optimization algorithms for SABR and broader exemplar-based VC systems.Index terms should be included as shown below.

Index Terms: sparse coding, voice conversion, dictionary learning, exemplar-based

1. Introduction

Accent Conversion (AC) is the task of converting the speech of a nonnative speaker (L2) to have the pronunciation pattern (e.g. prosodic and segmental content) of a native speaker (L1) while retaining the voice quality of the L2 [1]. AC methods have application to pronunciation training, where prior research has shown that it can be made more effective by matching the learner to model voices that resemble the learner's voice [2-5]. A number of studies have applied voice conversion (VC) techniques to the problem of AC, by transforming utterances from a native speaker to match the voice quality of the L2 learner [1, 6]. However, these techniques can have onerous training requirements, making them impractical for instructional settings. Low-resource VC methods such as exemplar-based VC require significantly fewer amounts of training data, but also require time-aligned source and target utterances [7-9], which is challenge when pronunciation errors (e.g., phoneme substitutions, additions, and deletions) are present, as is common in L2 speech, which ultimately reduces synthesis quality [6, 10, 11].

To address this issue, we recently developed a low-resource, exemplar-based VC technique which used phoneme-based "anchors" (i.e. exemplars) to model a speaker's voice. Termed a sparse, anchor-based representation of speech (SABR) [12-14], the technique decomposes a source speaker's spectrum into "weights," (i.e., sparse codes) that separate linguistic content from speaker identity. SABR performs VC by taking the source speaker's weights (i.e., linguistic content) and combines them with anchors from the L2 learner. SABR anchors are learned from phoneme labels gathered via forced alignment, so issues arising from time-aligning source and target speakers (e.g., in the case of native-to-nonnative voice conversion) are largely avoided. However, SABR anchors cannot represent phonemes with multiple acoustic states (i.e. diphthongs or stops), or instances where a nonnative speaker mispronounced the phoneme. Optimizing anchor selection, as well as allowing for multiple anchors per phoneme, would improve synthesis quality and reduce the VC error.

In this paper, we propose a clustering-based exemplar selection algorithm called Anchor Removal and Splitting (ARS) to address the issue of phoneme selection in the SABR model. ARS builds a hierarchical cluster for each phoneme in a training dataset and greedily *removes* or *splits* anchors to reduce the VC error, allowing multiple anchors to represent a phoneme or the anchor to be removed entirely. We evaluate the algorithm using a dataset of speech recordings from native and non-native speakers in the ARCTIC [15] and L2-ARCTIC [16] corpora, respectively, and compared them against an exemplar-based VC baseline that relied on time-aligned source and target dictionaries [8]. ARS had significantly higher acoustic quality than the baseline system in native-to-nonnative conversions while needing two orders of magnitude fewer anchors. These improvements in acoustic quality came at no loss in VC performance, measured by the ability of the algorithms to capture the voice quality of the target speaker. Finally, a detailed analysis of ARS shows that it preferentially chooses to split phonemes that have multiple sub-phoneme units, and phonemes that are frequently substituted (i.e., mispronounced) by L2 speakers. These results show that ARS can be used as a tool to select appropriate exemplars, especially in low-resource settings.

The remainder of this paper is organized as follows. Section 2 reviews prior work on Accent Conversion exemplar-based VC. Section 3 describes the VC algorithm used in this paper (SABR) and ARS. Section 4 describes the experimental design and speech corpora used in our study. Section 5 presents objective and subjective experimental results. The article concludes with a summary of our findings and directions for future work.

2. Prior work

Accents are characterized by prosodic and segmental differences with respect to a norm, so AC methods must account for these to capture the target speaker's voice quality. Aryal et al. [6] proposed an alternative "acoustic similarity" alignment for use in a GMM-based VC algorithm. The authors used Vocal Tract Length Normalization (VTLN) to account for coarse physiological differences between the L1 and L2 speakers. Then, they paired each L2 frame to the closest L1 frame (according to Mel-Cepstral Distortion) and, likewise, paired each L1 frame to the closest L2 frame, forming a lookup

http://dx.doi.org/10.21437/Interspeech.2021-1740

table from which a GMM was trained. More recently, Zhao et al. [11] presented another alignment method to account for pronunciation differences between the L1 and L2 speakers. Instead of VTLN, the authors computed a Phonetic Posteriorgram (PPG) for each L1 and L2 frame, then paired L1 and L2 frames to minimize the KL-divergence of their respective PPGs. However, both of these methods require significant amounts of training data, making them infeasible for pronunciation training contexts.

Exemplar-based voice conversion was originally developed as a method to account for over-smoothing of parametric statistical voice conversion methods [7, 9, 17]. These methods are well-suited for low-resource scenarios, as they typically require only a few dozen training utterances. Aihara et al. [7] proposed a method for building a phoneme-categorized dictionary, which added a penalty to the objective function so that the conversion algorithm was forced to select target exemplars from the same phonetic class as the source. Sisman et al. [18] demonstrated an exemplar-based VC system that appended PPG to the selected exemplar dictionaries to encode additional phonetic information. These results suggest that selecting exemplars to retain similar phonetic content will significantly improve synthesis quality.

Phoneme labels derived from ASR are also often used in neural voice conversion [19-22]. In [23], the authors used a latent phoneme embedding in a vector quantized VAE system. They found that including the embedding significantly improved the synthesis quality of VQ-VAE VC. In [24], Zhou et al. used PPGs learned from two speech recognition systems to perform voice conversion on speakers of two different languages, allowing the conversion system to identify linguistic content from the source speaker that did not resemble the target speaker's language. The inclusion of both PPGs significantly improved the synthesis quality and speaker identity.

3. Methods

3.1. SABR: Spares, Anchor-Based Representation of Speech

The voice conversion algorithm that underlies this work (SABR) represents an utterance as a sparse linear combination of speaker-dependent phonemic anchors [13]. The intuition behind the method is that the sparse code of an utterance relative to these anchors encodes the linguistic content.

Given a speech spectrum X_S, SABR decomposes it as:

$$X_S = A_S W_S, \tag{1}$$

where W_s is a sparse set of weights, and A_S is a set of speaker-dependent phoneme "anchors." For an utterance with T frames, N acoustic spectral features (e.g., MFCCs), and K anchors, $X_S \in \mathbb{R}^{N\times T}$, $A_S \in \mathbb{R}^{N\times K}$, and $W_S \in \mathbb{R}^{K\times T}$. SABR uses a single anchor per phoneme, which is computed by selecting the centroid of all frames that have the corresponding phoneme label in the speaker's training corpus.

SABR uses the Lasso [25] to estimate the weights (sparse codes) $W_S \in \mathbb{R}^{K\times T}$:

$$\min_{W_S} \|X - A_S W_S\|_2^2 + \|W_S\|_1, \qquad s.t. \|W_S\|_1 \le \lambda \tag{2}$$

where $\|\cdot\|_1$ is the L1 norm and λ is the maximum sum allowed for each W_S, which acts as a regularization term.

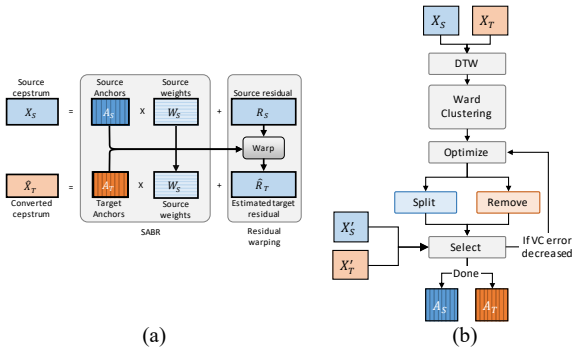


Figure 1: *Illustration of the overall SABR algorithm and the proposed optimization algorithm. (a) An overview of the SABR VC method. (b) A block diagram of the proposed Anchor Removal and Splitting (ARS) algorithm.*

To obtain an estimate of the target speaker's spectrum, a target anchor set A_T is built in the same manner as A_S: one spectral anchor per phoneme label, corresponding to the centroid of all target speaker training data with that label. An estimate of the target speaker's spectral envelope $\hat{X}_T$ is obtained as the product of the source weights and target anchors:

$$\hat{X}_T = A_T W_S. \tag{3}$$

This estimated spectrum lacks spectral detail, as the residual from eq. (2) has been discarded. In prior work [14], we proposed a method for transforming the source residual to the target speaker's space using frequency warps learned from the source and target anchor sets A_S and A_T, represented here as F_R:

$$\hat{X}_T = A_T W_S + F_R(X_S - A_S W_S; A_S, A_T, W_S) \tag{4}$$

For brevity, we refer the reader to [14] for details on the residual warping method. The overall approach is illustrated in Figure 1 (a).

3.2. Anchor Removal and Splitting (ARS) algorithm

The proposed optimization method, Anchor Removal and Splitting (ARS), addresses two issues. First, using single anchors per phoneme may not be enough to represent some phonemes classes, such as stops or affricates, which contain several sub-states. Second, the phoneme inventory of the L2 speaker may be different from that of the L1 speaker and will likely include mispronunciations of a phoneme. As a result, the source-target anchors may be mismatched, introducing distortions in the VC synthesis. To address these issues, at each iteration ARS either removes an anchor or "splits" it into sub-anchors, depending on which action most reduces the VC error against a set of validation utterances.

Initially, we compute a binary tree of cluster centroids for each phoneme using Ward's method [26]. These clusters are learned by concatenating time-aligned source and target training data: $[X_S^T, X_T^T]^T$. The root node of the binary cluster tree corresponds to the centroids of each phoneme. We then optimize the anchor sets iteratively by either *splitting* an anchor into the two child subtrees, or *removing* an anchor entirely. During the *split* operation, a given anchor is replaced with its two child nodes from the phoneme's cluster tree, representing the two higher-detail clusters in that phoneme. The *removal* operation simply removes the given anchor from the anchor set

and all child nodes of that anchor from the phoneme tree. These two operations address the two potential issues described previously. The split operation allows for multiple anchors per phoneme, aiding in the representation of certain phoneme classes. In the case that one pair of source-target anchors contains a mispronunciation and is mismatched, the removal operation allows for that anchor to be discarded.

On each iteration t, the *split* and *removal* operations are performed on the current anchor sets A_S^t and A_T^t, resulting in a temporary anchor sets $A_S^{k,f}$, $A_T^{k,f}$:

$$[A_s^{k,f}, A_T^{k,f}] = f(A_S^t, A_T^t), \quad (5)$$

where f are the two ARS operations and k is the anchor on which the operation was performed.

For each anchor-operation pair, the VC error (computed from eq. (3)) is measured against a validation data set X_S' and X_T':

$$[A_s^{t+1}, A_T^{t+1}] = \underset{A_S^{k,f},\, A_T^{k,f}}{\operatorname{argmin}}(X_T' - A_T^{k,f} W_S^{k,f}), \quad (6)$$

where $W_S^{k,f}$ are the weights computed from eq. (2) using anchors $A_S^{k,f}$ and the validation data X_S'. The temporary anchor set with the minimum VC error is used as the input to the next iteration, and the *split* and *removal* operation are tested again against the new anchor set. This iterates until the VC error does not decrease, or a maximum number of iterations is reached. A block diagram of the ARS algorithm is shown in Figure 1 (b).

4. Experiments

4.1. Data and implementation details

We evaluated the ARS proposed algorithm using the CMU ARCTIC speech corpus [15] and the L2-ARCTIC speech corpus (v 1.0) [16]. L2-ARCTIC is a corpus based on the prompts of the ARCTIC database, but with L2 speakers of English from six first languages: Mandarin, Hindi, Arabic, Spanish, Korean, and Vietnamese [1]. We conducted both subjective and objective evaluations in both native-to-native (ARCTIC to ARCTIC, or *A2A* for short) and native-to-nonnative (ARCTIC to L2-ARCTIC, *A2L2*) contexts. For objective experiments, we evaluated all possible speaker pairs in A2A and A2L2 conditions; but for perceptual experiments (synthesis quality and speaker identity tests), we only evaluated the speaker pairs in Table 1 (A2A) and Table 2 (A2L2). In a prior study [12], we observed that using SABR for Accent Conversion was very effective in reducing accentedness when converting from a native speaker to a nonnative speaker; because of this, we did not perform such tests in this study.

To illustrate the time-alignment difference between native and nonnative speaker pairs, we computed the average difference between DTW trajectories for A2A and A2L2 speaker pairs (see Table 3). As shown, alignment of A2L2 speaker pairs incurs nearly twice the error of alignment of A2A pairs, which *highlights the challenges of using conventional exemplar-based VC methods when the target speakers are non-native.*

Table 1: *A2A speaker pairs for perceptual experiments.*

Source speaker	Target speaker
BDL (M)	RMS (M)
SLT (F)	CLB (F)
RMS (M)	SLT (F)
CLB (F)	BDL (M)

Table 2: *A2L2 speaker pairs for perceptual experiments.*

Source speaker	Target speaker	First language
BDL (M)	HKK (M)	Korean
SLT (F)	SKA (F)	Arabic
RMS (M)	YDCK (F)	Mandarin
CLB (F)	EVBS (M)	Spanish

Table 3: *Average time alignment differences.*

Corpus	Average alignment error
L2-ARCTIC average	221 ms ± 36 ms
ARCTIC average	124 ms ± 15 ms

We performed time alignment using the MFCC features and dynamic time warping (DTW) [27]. We used STRAIGHT [28] with 1 ms frame steps and 80 ms window size to extract aperiodicity, fundamental frequency, and spectral envelope from each utterance. We then computed a 25-dimension MFCC vector. SABR models ignored $MFCC_0$, as it contains energy.

For synthesis, the energy from the source speaker was copied to the estimated target speaker's spectrum. We converted the pitch of the source utterance to match the pitch range of the target speaker using log mean-variance scaling [29]. To solve for the Lasso, we used the LARS solver from the SPAMS sparse coding toolbox [30].

4.2. Comparison systems

We evaluated the proposed algorithm against the original SABR system and a VC baseline system[2]:

- *ARS*: source and target anchor sets optimized by the ARS algorithm (Section 3.2). Twenty training utterances were split up into two sets of 10 utterances: one to train the anchors, and the other 10 to use as a validation set at the end of each ARS iteration.
- *SABR*: the default SABR anchors—one anchor per phoneme, selected by computing the centroid of all frames with that phoneme label (Section 3.1).
- *Baseline:* Exemplar-Based VC with Residual Compensation [8], a time-aligned exemplar-based VC approach. The dictionaries in this model were substantially larger than that of the SABR and ARS models (4720 atoms on average for the baseline models).

We used 20 parallel utterances to train the three systems. The training utterances were selected in such a way as to maximize phoneme variability. For all systems, we measured the VC error using Mel-Cepstral Distortion (MCD) [29] on a test set of 200 utterances selected from the ARCTIC "*A"* set of utterances.

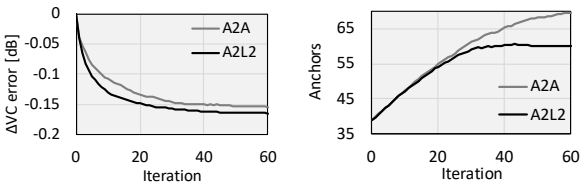
5. Results

5.1. Objective evaluation

To characterize the performance of ARS, we evaluated it from two perspectives: the per-iteration performance and the

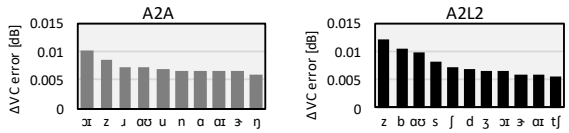
[1] At the time of our experiments, the Vietnamese speakers were not available and were not included in the objective experiments.

[2] Synthesis samples from the above systems can be found at https://cliberatore.github.io/samples/sabr-ars.html



(a) (b)

Figure 2*: Performance of the ARS algorithm by iteration in terms of (a) change in VC error from initial SABR anchor set (b) number of source/target anchors.*



(a) (b)

Figure 3: *Reduction in VC error for top ten phonemes "split" by the ARS algorithm. (a) A2A pairs. (b) A2L2 pairs.*

Table 4: *Results of the perceptual tests. Average and standard errors of the ratings are shown.*

	MOS		Identity	
Method	A2A	A2L2	A2A	A2L2
Baseline	2.19 ± 0.10	1.36 ± 0.08	89% ± 2.1%	89% ± 3.5%
SABR	3.04 ± 0.10	2.54 ± 0.10	88% ± 1.3%	84% ± 2.2%
ARS	3.18 ± 0.10	2.7 ± 0.09	84% ± 3.0%	84% ± 3.8%

phonemes the ARS algorithm selected for splitting, as that operation contributed most to reducing the VC error.

Figure 2 shows the per-iteration results. ARS reduces the VC error for both speaker pairs, but more for A2L2 pairs because of pronunciation differences between the source and target speakers. Effects of time-alignment differences are visible in the number of anchors selected (see Figure 2 (b)). Because A2A pairs are generally less affected by accent and time-alignment issues, the source and target clusters are more likely to contain similar phonetic information, so ARS favors the splitting operation. The A2L2 pairs reach an average of 60 anchors, whereas A2A pairs had an average of 69.8 anchors.

As the *split* operation had a stronger impact on the reduction in VC error, we examined which phonemes selected in that operation contributed to the greatest reduction in VC error. We computed the total amount the VC error decreased when phoneme k was split as:

$$\Delta e_k = \sum_{t=1}^{60} S_k^t (e_{t-1} - e_t), \qquad (7)$$

where Δe_k is the change in VC error for phoneme k, t is the ARS iteration, S_k^t is an indicator variable that is 1 when phoneme k was selected for splitting on iteration t and 0 otherwise. For e_0, we used the VC error of the initial SABR models. Figure 3 shows the results of eq. (7) computed for all A2A and A2L2 speaker pairs, (a) shows A2A speakers and (b) shows A2L2 speakers. First, ARS favored voiced phonemes and vowels on A2A pairs more than on A2L2 pairs. Second, on A2L2 pairs, ARS often split phonemes with known voicing substitution errors (e.g., /s/ and /z/, common in the L2-ARCTIC corpus [16]). Finally, on A2L2 pairs, the most-split phonemes were those where the phoneme labels contained multiple states over the production of the phoneme (e.g., diphthongs and stops) or had the same articulation, but different voicing.

The average MCD of the three methods, evaluated on the test set, as well as the final dictionary sizes are shown in Figure 2. For both A2A and A2L2 pairs, ARS had significantly lower VC error than the original SABR model (A2A and A2L2, $p \ll 0.001$, paired t-test). Notably, there was no significant difference in the VC error of the proposed optimization method and the baseline model (A2A, $p \geq 0.05$, A2L2, $p \geq 0.35$, paired t-test), a positive result given that the dictionaries on the baseline model were more than two orders of magnitude larger.

5.2. Subjective evaluation

5.2.1. Mean Opinion Score

We performed a Mean Opinion Score (MOS) test to measure the synthesis quality of the three VC systems. We recruited participants ($n = 20$) on Amazon Mechanical Turk to rate the quality of an utterance on a 5-point scale (1 = "low quality"; 5 = "high quality"). For each synthesis method, we asked participants to rate 40 utterances (5 per speaker pair). Following [31], we included unmodified references to ensure participants were not randomly guessing. Results are shown in Table 4.

MOS ratings for ARS were approximately twice as high as those of the baseline system ($p \ll 0.01$, single-tailed t-test), a remarkable result given that they include far fewer anchors in their dictionaries. This MOS rating is significantly lower than those reported by the authors of the baseline system. We believe this is due to the difficulty in time-aligning native to non-native utterances, which is critical for the baseline system.

Notably, the baseline method had a significantly larger difference in MOS between A2A and A2L2 versus the SABR and ARS methods ($p < 0.01$, both methods, single-tailed t-test). However, ARS did not significantly improve on the baseline SABR synthesis quality in either A2A or A2L2 cases ($p > 0.13$, single-tailed t-test), suggesting that the additional anchors, while reducing VC error, do not impact synthesis quality as significantly.

5.2.2. Speaker identity test

We performed an XAB speaker identity test to evaluate the effect of ARS on the speaker identity of VC utterances. Participants ($n = 20$) were presented with three utterances: a VC utterance (X), and utterances from the source or target speaker (A, B), counterbalancing A and B. Following [32], utterances were played in reverse to mask the effects of accent. For each VC method, we performed 32 evaluations (4 per speaker pair). Results are shown in Table 4. There was no statistically significant difference between the three systems ($p > 0.12$, two-tailed t-test).

6. Conclusion and future work

In this paper, we proposed an optimization algorithm for selecting exemplars in exemplar-based VC. The proposed method reduced voice VC over the original SABR method and modestly improved synthesis quality of VC utterances. The phonemes selected by the algorithm reflected nonstationary phonemes (e.g., diphthongs and stops) as well as phonemes nonnative speakers were known to mispronounce or having difficulty producing. This would be useful in VC methods that use clustering to build models, e.g., the PPG-based clustering method proposed by [21]. Future work will examine extending SABR anchors to include multiple anchors for these phonemes.

7. Acknowledgements

This work was partially supported by NSF Grants 1619212 and 1623750.

8. References

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

[2] 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 *Australian Int. Conf. on Speech Science & Technology*, 2006, pp. 24-29.

[3] K. Nagano and K. Ozawa, "English speech training using voice conversion," in *Proc. of Int. Conf. on Spoken Language Processing (ICSLP)*, Kobe, Japan, 1990.

[4] M. Peabody and S. Seneff, "Towards automatic tone correction in non-native mandarin," in *International Symposium on Chinese Spoken Language Processing*, 2006, pp. 602-613: Springer.

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

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

[7] R. Aihara, T. Nakashika, T. Takiguchi, and Y. Ariki, "Voice conversion based on non-negative matrix factorization using phoneme-categorized dictionary," in *ICASSP*, 2014, pp. 7894-7898.

[8] Z. Wu, T. Virtanen, E. S. Chng, and H. Li, "Exemplar-based sparse representation with residual compensation for voice conversion," *IEEE/ACM Transactions on Audio, Speech and Language Processing,* vol. 22, no. 10, pp. 1506-1521, 2014.

[9] Z. Wu, T. Virtanen, T. Kinnunen, E. S. Chng, and H. Li, "Exemplar-based voice conversion using non-negative spectrogram deconvolution," in *ISCA Workshop on Speech Synthesis*, 2013.

[10] S. Aryal and R. Gutierrez-Osuna, "Data driven articulatory synthesis with deep neural networks," *Computer Speech & Language,* vol. 36, pp. 260-273, 2016.

[11] G. Zhao, S. Sonsaat, J. Levis, E. Chukharev-Hudilainen, and R. Gutierrez-Osuna, "Accent Conversion Using Phonetic Posteriorgrams," in *ICASSP*, 2018, pp. 5314-5318: IEEE.

[12] S. Ding *et al.*, "Golden Speaker Builder - An interactive tool for pronunciation training," *Speech Communication,* vol. 115, pp. 51-66, 2019.

[13] C. Liberatore, S. Aryal, Z. Wang, S. Polsley, and R. Gutierrez-Osuna, "SABR: sparse, anchor-based representation of the speech signal," in *Interspeech*, 2015.

[14] C. Liberatore, G. Zhao, and R. Gutierrez-Osuna, "Voice conversion through residual warping in a sparse, anchor-based representation of speech," in *ICASSP*, 2018.

[15] J. Kominek and A. W. Black, "The CMU ARCTIC speech databases," in *ISCA Workshop on Speech Synthesis*, 2004, pp. 223-224.

[16] G. Zhao *et al.*, "L2-ARCTIC: A Non-Native English Speech Corpus," in *Interspeech*, 2018, pp. 2783-2787.

[17] 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.

[18] B. Sisman, M. Zhang, and H. Li, "A Voice Conversion Framework with Tandem Feature Sparse Representation and Speaker-Adapted WaveNet Vocoder," in *Interspeech*, 2018.

[19] Y. Saito, Y. Ijima, K. Nishida, and S. Takamichi, "Non-parallel voice conversion using variational autoencoders conditioned by phonetic posteriorgrams and d-vectors," in *ICASSP*, 2018, pp. 5274-5278: IEEE.

[20] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *Multimedia and Expo (ICME), 2016 IEEE International Conference on*, 2016, pp. 1-6: IEEE.

[21] F.-L. Xie, F. K. Soong, and H. Li, "A KL Divergence and DNN-Based Approach to Voice Conversion without Parallel Training Sentences," in *Interspeech*, 2016, pp. 287-291.

[22] G. Zhao and R. Gutierrez-Osuna, "Using phonetic posteriorgram based frame pairing for segmental accent conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 27, no. 10, pp. 1649-1660, 2019.

[23] S. Ding and R. Gutierrez-Osuna, "Group latent rmbedding for vector quantized variational autoencoder in non-parallel voice conversion," in *Interspeech*, 2019, pp. 724-728.

[24] Y. Zhou, X. Tian, H. Xu, R. K. Das, and H. Li, "Cross-lingual voice conversion with bilingual phonetic posteriorgram and average modeling," in *ICASSP*, 2019, pp. 6790-6794: IEEE.

[25] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society,* vol. Series B, pp. 267-288, 1996.

[26] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association,* vol. 58, no. 301, pp. 236-244, 1963.

[27] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *KDD workshop*, 1994, vol. 10, no. 16, pp. 359-370: Seattle, WA.

[28] H. Kawahara, "STRAIGHT, exploitation of the other aspect of VOCODER: Perceptually isomorphic decomposition of speech sounds," *Acoustical Science and Technology,* vol. 27, no. 6, pp. 349-353, 2006.

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

[30] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research,* vol. 11, no. Jan, pp. 19-60, 2010.

[31] S. Buchholz and J. Latorre, "Crowdsourcing preference tests and how to detect cheating," in *Interspeech*, 2011, pp. 3053-3056.

[32] 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.