

# A Study of Bilinear Models in Voice Conversion

Victor Popa<sup>1</sup>, Jani Nurminen<sup>2</sup>, Moncef Gabbouj<sup>1</sup>

<sup>1</sup>Department of Signal Processing, Tampere University of Technology, Tampere, Finland; <sup>2</sup>Nokia Devices R&D, Tampere, Finland.  
Email: victor.popa@tut.fi, jani.k.nurminen@nokia.com, moncef.gabbouj@tut.fi

Received February 7<sup>th</sup>, 2011; revised March 30<sup>th</sup>; accepted April 7<sup>th</sup>, 2011.

## ABSTRACT

*This paper presents a voice conversion technique based on bilinear models and introduces the concept of contextual modeling. The bilinear approach reformulates the spectral envelope representation from line spectral frequencies feature to a two-factor parameterization corresponding to speaker identity and phonetic information, the so-called style and content factors. This decomposition offers a flexible representation suitable for voice conversion and facilitates the use of efficient training algorithms based on singular value decomposition. In a contextual approach (bilinear) models are trained on subsets of the training data selected on the fly at conversion time depending on the characteristics of the feature vector to be converted. The performance of bilinear models and context modeling is evaluated in objective and perceptual tests by comparison with the popular GMM-based voice conversion method for several sizes and different types of training data.*

**Keywords:** Line Spectral Frequencies (LSF), Gaussian Mixture Model (GMM), Bilinear Models (BL), Singular Value Decomposition (SVD), Temporal Decomposition (TD), Factor Analysis

## 1. Introduction

Voice conversion is a transformation process applied to a speech signal to change the speaker identity to resemble a predetermined target speaker identity while leaving the speech content unaltered. The motivation for creating such a technology is related to its real life applications, among them the possibility to create personalized voices for text-to-speech systems (TTS) or to use it to recover the original voice identity in movie dubbing and speech-to-speech translations. There is also a big potential for other entertainment and security related applications.

The topic has received a great research interest and various methods, such as Gaussian mixture modeling (GMM) [1], frequency warping [2], artificial neural networks [3], hidden Markov models (HMM) [4], linear transformation [5], codebook based conversion [6], eigenvoices [7], maximum likelihood estimation of spectral parameter trajectory [8], partial least squares regression [9], have been proposed in the literature. Voice conversion remains an open issue as all the current methods have some weaknesses. For example, the GMM based methods, while being very popular for spectral conversion and considered to produce a good identity mapping,

This work was supported by the Academy of Finland, (application number 129657, Finnish Programme for Centres of Excellence in Research 2006-2011).

suffer from a known over-smoothing problem and result in relatively poor speech quality. Over-smoothing is a major issue in voice conversion and also affects the method proposed in this paper to some extent. On the other hand, the frequency warping produces good speech quality at the cost of a compromised identity conversion. The bilinear models represent a factor analysis technique introduced originally in [10] which attempts to model observations as a result of two underlying factors. This concept originated from the observation that living organisms are capable of separating “style” and “content” in their perception. The separation into these two factors gives a flexible representation and facilitates the generalization to unseen styles or content classes. Furthermore, this framework provides efficient training algorithms based on singular value decomposition (SVD). In [11] we have demonstrated with early results that bilinear models are a viable solution also for voice conversion, by studying the voice conversion in terms of style (speaker identity) and content (the phonetic information) with small parallel sets of training data.

In parallel training data, the speakers utter the same sentences. In contrast, if each speaker utters a different set of sentences, the data is referred to as text independent data. The term non-parallel will be used in this paper to denote a text independent data in which all speakers

use the same phonetic alphabet and usually the same language. The extreme case of text independent data where the speakers speak in different languages that typically have different phoneme sets is commonly referred to as the cross-lingual case. The greatest challenge in dealing with text independent data is to find an alignment between the source and target features. By proper alignment, the text independent case can be reduced to the parallel scenario and similar conversion methods can be used.

In this paper, we propose a spectral transformation technique for voice conversion based on bilinear models and we also propose an alignment scheme for text independent data based on [12]. Due to their capability for reconstructing missing data, we hypothesize that bilinear models may be particularly useful in text independent cases and especially in cross-lingual voice conversion. The proposed conversion technique based on bilinear models is compared with the widely used GMM based method using both parallel and text independent data with very small to very large sizes of the training sets. Our results offer a comprehensive perspective over the performance and the limitations that bilinear models have in voice conversion. In addition, we also try to answer the question whether fitting conversion models to contextual data (a subset of the training data) is more appropriate for capturing details than the usual models optimized globally over the entire training data.

The paper is organized as follows. In the next section, we introduce the method based on asymmetric bilinear models and explain how it can be applied in voice conversion with parallel training data. In Section 3, we present the challenges of the non-parallel and cross-lingual voice conversion and give a practical solution to the alignment problem. We also introduce in a separate subsection the concept of contextual conversion that can be utilized with both non-parallel and cross-lingual data. In Section 4, we describe the practical experiments and discuss the objective measurements and listening test results. Finally, concluding remarks and potential directions for future research are presented in Section 5.

## 2. Voice Conversion with Asymmetric Bilinear Models

The general style and content framework originally presented in [10] can be successfully utilized for spectral transformation in voice conversion. This section describes the asymmetric bilinear models following the notations used in [10], and discusses the properties of the technique from the voice conversion perspective. In the following, we will use the terms *style* and *content* to refer to the *speaker identity* and *phonetic information*, respectively, which constitute the two independent factors un-

derlying our observations. In this paper, the observations are represented as line spectral frequency (LSF) vectors.

### 2.1. Asymmetric Bilinear Models

In a symmetric model, the style  $s$  (the speaker identity) and content  $c$  (the phonetic information) are represented as parameter vectors denoted  $a^s$  and  $b^c$  of dimension  $I$  and  $J$ , respectively. Let  $y^{sc}$  denote an observation vector in style  $s$  and content class  $c$ , and let  $K$  denote its dimension. In our case,  $y^{sc}$  is an LSF vector of one speaker and it represents the spectral envelope of a particular speech frame.  $y^{sc}$  as a bilinear function of  $a^s$  and  $b^c$ , in its most general form, is given by [10]

$$y_k^{sc} = \sum_{i=1}^I \sum_{j=1}^J w_{ijk} a_i^s b_j^c \quad (1)$$

where  $i, j$  and  $k$  denote elements of the style, content and observation vectors. The terms  $w_{ijk}$  describe the interaction between the content (phonetic information) and style (speaker identity) factors and are independent of both of these factors.

Asymmetric bilinear models are derived from the symmetric bilinear models by allowing the interaction terms  $w_{ijk}$  to vary with the style leading to a more flexible style description [10]. Equation (1) becomes

$$y_k^{sc} = \sum_{i,j} w_{ijk}^s a_i^s b_j^c \quad (2)$$

Combining the style(identity)-specific terms in (2) into

$$a_{jk}^s = \sum_i w_{ijk}^s a_i^s \quad (3)$$

gives

$$y_k^{sc} = \sum_j a_{jk}^s b_j^c \quad (4)$$

By denoting as  $A^s$  the  $K \times J$  matrix with entries  $a_{jk}^s$ , (4) can be rewritten as

$$y^{sc} = A^s b^c. \quad (5)$$

In this formulation the  $a_{jk}^s$  terms can be interpreted as a style (identity) specific linear map from the content (phonetic info) space to the observation space (LSF). It is worth to note that unlike the face image case presented in [10] in which basis vectors appeared to have some concrete interpretations, no obvious patterns could be observed and no meaningful interpretation could be attributed to the parameter vectors and matrices in our particular application.

### 2.2. Model Fitting Procedure

The objective of the model fitting procedure is to train the parameters of the asymmetric model to minimize the total squared error over the entire training dataset. This is equivalent to maximum likelihood (ML) [13] estimation

of the style and content parameters based on the training data, with the assumption that the data was produced by the models plus independently and identically distributed (i.i.d.) Gaussian noise [10].

The model fitting is described for  $S$  speakers (styles) and  $C$  content classes which could correspond to phonetically justified units. Our training material consists of  $R_1$  LSF vectors of speech uttered by speaker  $s_1$  in style  $s = 1$ ,  $R_2$  LSF vectors of speech uttered by speaker  $s_2$  in style  $s = 2$ , and so on. The individual (speaker based) parametric sequences are pooled together in a training sequence of size  $R = R_1 + R_2 + \dots + R_S$ . Let  $y(r)$  denote the  $r$ -th training observation ( $r = 1, \dots, R$ ) from the pooled data. Each  $y(r)$  is an LSF vector coming from a certain speaker (style) and from one of  $C$  content classes. The binary indicator variable  $h^{sc}(r)$  takes the value 1 if  $y(r)$  is in style  $s$  and content class  $c$  and the value 0, otherwise. The total squared error  $E$  of the asymmetric model given in (5) is computed over the training set using

$$E = \sum_{r=1}^R \sum_{s=1}^S \sum_{c=1}^C h^{sc}(r) \|y(r) - A^s b^c\|^2 \quad (6)$$

In the case of parallel training data, the speech sequences of the  $S$  speakers can be time aligned and each  $S$ -tuple of aligned LSF vectors will be assumed to represent a distinct class. Consequently, there will be only one LSF vector from each speaker (style) falling into each content class.

If the training set contains an equal number of observations in each style and in each content class (in our case one observation), a closed form procedure exists for fitting the asymmetric model using singular value decomposition (SVD) [10].

In the proposed case of parallel and aligned training data, in order to work with standard matrix algorithms, we stack the  $SC (= R)$  LSFs ( $K$  dimensional column) vectors into a single  $SK \times C$  matrix, similarly as in [10],

$$Y = \begin{bmatrix} y^{11} & \dots & y^{1C} \\ \dots & \dots & \dots \\ y^{S1} & \dots & y^{SC} \end{bmatrix}. \quad (7)$$

We can express now the asymmetric model in the following very compact matrix form

$$Y = AB, \quad (8)$$

where the  $(SK) \times J$  matrix  $A$  and the  $J \times C$  matrix  $B$  represent the stacked style and content parameters,

$$A = \begin{bmatrix} A^1 \\ \dots \\ A^S \end{bmatrix}, \quad (9)$$

$$B = [b^1 \quad \dots \quad b^C]. \quad (10)$$

To find the optimal style and content parameters for (8) in the least square sense, we can compute the SVD of  $Y = UZV^T$  [10] with complexity  $O(\min((SK)^2C, (SK)C^2))$ . ( $Z$  is considered to have the diagonal eigenvalues in decreasing order.) By definition, we choose the style parameter matrix  $A$  to be the first  $J$  columns of  $UZ$  and the content parameter matrix  $B$  to be the first  $J$  rows of  $V^T$ . There are many ways to choose the model dimensionality  $J$  e.g. from prior knowledge, by requiring a desired level of approximation of data, or by identifying an ‘‘elbow’’ in the singular value spectrum [10].

Note that using a relatively small model order  $J = S \cdot K$  prevents overfitting and that potential numerical problems due to very large matrices can be avoided by computing an *economy size* decomposition (in Matlab).

An important aspect in cases with very high dimensional features is the selection of the model dimensionality ( $J$ ) since high model dimensionalities could cause overfitting problems. Our experiments with  $K = 16$  and  $K = 10$  dimensional LSFs produced similar results with the difference that error decreases and stabilizes quicker for  $K = 10$  as fewer parameters require less data for a reliable training.

### 2.3. Application in Parallel Voice Conversion

One of the tasks that fall under the framework proposed in [10] and which is of particular interest in voice conversion is *extrapolation* illustrated in **Table 1**. In this character example the letters  $D$  and  $E$  (content classes) do not exist and need to be generated in the new font (style) based on the labeled training set (first two rows) [10].

The term extrapolation refers to the ability to produce equivalent content in a new style, in our case to produce speech as that uttered by a source speaker but with a target speaker’s voice. Therefore, voice conversion is a direct analogy of the extrapolation task. Extending a bit the concept of voice conversion we can also define it as the generation of speech with a target voice, reproducing content uttered by multiple source speakers.

We can formulate the problem of parallel voice conversion as an extrapolation task as follows. Given a training set of parallel speech data from  $S$  source speakers

**Table 1. The extrapolation task illustrated for characters.**

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$	$\mathcal{D}$	$\mathcal{E}$
A	B	C	D	E
<b>A</b>	<b>B</b>	<b>c</b>	?	?

and the target speaker, the task is to generate any test sentence in the target voice starting from  $S$  utterances of the test sentence corresponding to each of the  $S$  source voices (styles).

The alignment of the training data ( $S$  source + one target speakers) is a prerequisite step for model estimation and is usually done with DTW. On the other hand, the alignment of the test data ( $S$  utterances of the source speakers) is also required if  $S > 1$ . The test data is aligned to a target utterance of the test sentence which exists in this study for evaluative purposes. In real applications, where such a target utterance does not exist, the test data should be aligned to one of its  $S$  source utterances, preferably a source speaker (denoted as main source speaker) whose speaking style resembles that of the target speaker. Choosing the alignment in this way has at least two advantages: provides a natural speaking style for the converted utterance which is close to the target one and reduces alignment problems because at least the main source speaker's utterance does not have to be interpolated in the alignment process.

A so-called *complete* data is formed by concatenating the aligned training and test data of the  $S$  source speakers. The *complete* data is assumed to have as many classes as LSF vectors per speaker and is used to fit the asymmetric bilinear model of (8) to the  $S$  source styles following the closed-form SVD procedure described in Section 2.2. This yields a  $K \times J$  matrix  $A^s$  for each source style (voice)  $s$  and a  $J$  dimensional vector  $b^c$  for each LSF class  $c$  in the *complete* data (hence producing also the  $b^c$ - $s$  of the test utterance).

The model adaptation to the incomplete new style  $t$  (the target voice) can be done in closed form using the content vectors  $b^c$  learned during training. Suppose the aligned training data from our target speaker (style  $t$ ) consists of  $M$  LSF vectors which by convention we considered to be in  $M$  different content classes  $C_T = \{c_1, c_2, \dots, c_M\}$ . We can derive the style matrix  $A^t$  that minimizes the total squared error over the target training data,

$$E^* = \sum_{c \in C_T} \|y^{tc} - A^t b^c\|^2. \quad (11)$$

The minimum of  $E^*$  is found by solving the linear system

$$\frac{\partial E^*}{\partial A^t} = 0. \quad (12)$$

The missing observations (LSFs) in the style  $t$  and a content class  $c$  of the test sentence can be then synthesized from  $y^{tc} = A^t b^c$ . This means we can estimate the target version of the test sentence by multiplying the target style matrix  $A^t$  with the content vectors corresponding to the test sentence.

## 2.4. The Proposed Algorithm

The proposed technique is summarized in the following algorithm in which we assume that LSF features are available.

1) Time align the training data (source speakers and target speaker) and the test sentence (source speakers only) which is to be converted to the target voice. The alignment will respect the timeline or prosody of the main source speaker.

2) Form the *complete* data of the source speakers by combining their training data with their test sentence data.

3) Run SVD to fit the asymmetric bilinear model to the *complete* data. This step will find the style matrices  $A^s$  for all the source speakers and the content vectors  $b^c$  for all the content (LSF) classes, including the classes (LSFs) in the test sentence.

4) Find the style matrix  $A^t$  of the target voice by minimizing (11), thus solving (12).

5) Synthesize the converted LSF vectors as  $y^{tc} = A^t b^c$  with  $A^t$  found at step 4 and the content vectors  $b^c$  of the test sentence found at step 3.

## 3. Non-Parallel and Cross-Lingual Voice Conversion

In contrast to the parallel scenario, in text independent corpora the speakers need not utter the same sentences. If the speakers use the same phonetic set in their utterances, we refer to it as the non-parallel case. In a cross-lingual case even the phoneme sets used by speakers are different. In our experiments, we have designed a so called simulated cross-lingual corpus from originally parallel intra-lingual data by ensuring that the target speaker utilizes in his utterances only a subset of the phonemes used by the source speakers.

By observing that the correspondence of speech content between speakers has moved from the utterance level to the phoneme level we propose an alignment scheme based on temporal decomposition and phonetic segmentation of the speech signal starting from [12]. The role of the alignment scheme in this work is to facilitate the use of parallel voice conversion algorithms with text independent data allowing us to focus on the evaluation of GMM and bilinear models, therefore here we limit ourselves at presenting the scheme and leave further analysis for future study.

The first part of this section introduces a theoretical framework for speech modeling based on articulatory phonetics that justifies our alignment scheme for text independent data presented in the second part of the section. Finally, the last part of the section introduces the idea of context modeling.

### 3.1. Temporal Decomposition

In the temporal decomposition (TD) model [14], speech is represented as a sequence of articulatory gestures that produce acoustic events. An acoustic event is associated with a so called event target and with an event function. The event target can be regarded as a spectral parameter vector and the event function denotes the activation level of that acoustic event as a function of time. The mathematical formulation of this model was given in [14] as:

$$\hat{y}(n) = \sum_{l=1}^L z_l \phi_l(n), \quad 1 \leq n \leq N, \quad (13)$$

where  $z_l$  denotes the  $l$ -th event target,  $\phi_l(n)$  describes the temporal evolution of this target,  $\hat{y}(n)$  is an approximation of the  $n$ -th spectral parameter vector  $y(n)$ ,  $N$  is the number of frames in the speech segment and  $L$  represents the number of event functions, ( $N \gg L$ ).

In the original formulation of the TD model [14] several event functions may overlap at any given location in the speech signal. A simplification of the original model was proposed in [15,16] in which only adjacent event functions are allowed to overlap leading to a second order TD model:

$$\hat{y}(n) = z_l \phi_l(n) + z_{l+1} \phi_{l+1}(n), \quad n_l \leq n < n_{l+1} \quad (14)$$

where  $n_l$  and  $n_{l+1}$  represent locations of events  $l$  and  $(l+1)$ .

A restriction on this model suggested in [17] requires the event functions to sum up to one. Furthermore, in order to better explain the temporal structure of speech, a modified restricted temporal decomposition (MRTD) was proposed in [18] and assumes that all event functions first grow from 0 to 1 and then decrease from 1 to 0. An illustration of the event functions with all the above restrictions is given in **Figure 1**.

The above assumptions are practically equivalent to saying that any spectral vector located between two event targets can be computed from the event targets by interpolation.

The MRTD algorithm [18] can be used to determine the event locations and the event targets. Interestingly, [19] suggests that these event targets convey speaker identity. However, MRTD cannot guarantee a fixed correspondence between the (number of) acoustic events and the phonetic units. Such a property is desirable for alignment purposes [12] but requires another method to find the event locations.

A method based on phonemes was proposed in [20,21] to represent phonemes with a fixed number of event targets. The method uses labeled utterances to segment the speech signal into phonemes. Each phoneme is divided into  $Q-1$  equal segments by  $Q$  equally spaced points which are used as event locations.  $Q$  is a free parameter

depending on the application. In [12]  $Q = 5$  was used.

In our work we distinguish between the middle stationary part of phonemes and phonetic transitions and segment the speech into stationary phonetic units and transient phonetic units as described in the next section. Each phonetic unit is divided by four equally spaced points ( $Q_{pu} = 4$ ), corresponding to seven event targets per phoneme ( $Q = 7$ ), and event targets are computed at those locations from an LSF representation of the phonetic unit as follows [18].

First the event functions are computed as:

$$\phi_l(n) = \begin{cases} 1 - \phi_{l-1}(n), & \text{if } n_{l-1} < n < n_l \\ 1, & \text{if } n = n_l \\ \min(\phi_l(n-1), \max(0, \hat{\phi}_l(n))), & \\ & \text{if } n_l < n < n_{l+1} \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

where  $l = 1:Q_{pu}$  and

$$\hat{\phi}_l(n) = \frac{\langle (y(n) - z_{l+1}), (z_l - z_{l+1}) \rangle}{\|z_l - z_{l+1}\|^2} \quad (16)$$

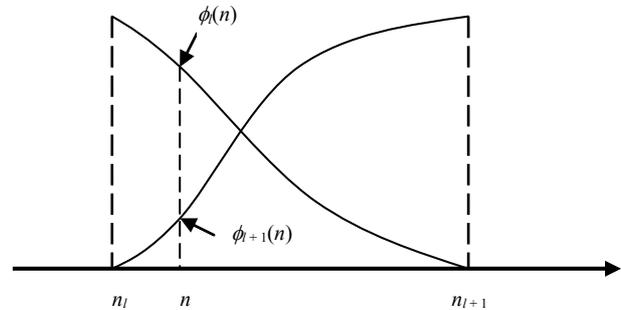
in which  $y(n)$  represents the  $n$ -th vector of the LSF sequence and the initial event targets  $z_l$  and  $z_{l+1}$  are vectors sampled from the LSF trajectories at the defined target locations  $n_l$  and  $n_{l+1}$ ,  $z_l = y(n_l)$ ,  $z_{l+1} = y(n_{l+1})$ .

The actual event target vectors are then calculated in least mean square sense using:

$$Z = Y\Phi^T (\Phi\Phi^T)^{-1} \quad (17)$$

and since these event target vectors may violate the frequency ordering property of LSFs a further refinement scheme is applied as in [18].

The event targets are used for alignment and used in conversion while the reconstruction of the phonetic unit to its original or to a desired number of feature vectors is done based on the event functions.



**Figure 1. Two adjacent event functions in the second order TD model.**

### 3.2. The Proposed Alignment Scheme

The next scheme requires phonetically labeled training data.

Let  $\Theta = \{\theta_1, \dots, \theta_p\}$  denote a phonetic set consisting of phonemes *common* to all source and target speakers and *rare* phonemes spoken only by the source speakers.

If  $\theta_j \in \Theta$  is the  $j$ -th phoneme (also denoted as  $p_j$ ) of an utterance occupying the time interval  $[\tau_{j-1} \ \tau_j]$  in the speech signal we define a stationary phonetic unit  $p_j p_j (= \theta_j \theta_j)$  as occupying time intervals  $[\tau_{1,j} \ \tau_{2,j}]$  where:

$$\tau_{1,j} = \tau_{j-1} + 0.25 \cdot (\tau_j - \tau_{j-1}) \quad (18)$$

and

$$\tau_{2,j} = \tau_{j-1} + 0.75 \cdot (\tau_j - \tau_{j-1}) \quad (19)$$

The transient phonetic unit  $p_{j-1} p_j (= \theta_g \theta_f)$  occupies intervals like  $[\tau_{2,(j-1)} \ \tau_{1,j}]$  instead.

Following the procedure in Section 3.1, the (LSFs of) phonetic units are decomposed into  $Q_{pu} = 4$  equally spaced event targets which can be concatenated into a phonetic unit based feature vector

$$Z = \begin{bmatrix} z_1^T & z_2^T & \dots & z_{Q_{pu}}^T \end{bmatrix}, \quad (20)$$

where  $z_q$  ( $1 \leq q \leq Q_{pu}$ ) denotes the  $q$ -th event target in a speech segment (phonetic unit). If we consider the frequency ordering property of the ‘‘LSF-like’’ event targets, this representation can be further normalized to

$$Z = \begin{bmatrix} z_1^T & z_2^T + \pi & \dots & z_{Q_{pu}}^T + (Q_{pu} - 1) \cdot \pi \end{bmatrix} \quad (21)$$

which is an ordered vector of frequencies within  $(0 \ Q_{pu}\pi)$ .

All the phonetic unit based feature vectors are then grouped by phonetic unit and speaker. We represent our training data in the form of a  $P \times P$  matrix  $D$ , structured in multiple layers (one for each speaker) having at node  $(g, f)$ :

- the stationary phonetic unit  $\theta_f \theta_f$  corresponding to phoneme  $\theta_f$ , if  $g = f$
- the transient phonetic unit  $\theta_g \theta_f$  between phonemes  $\theta_g$  and  $\theta_f$ , if  $g \neq f$

Aligned data is build for each phonetic unit by grouping phonetic unit based vectors from each layer of the unit’s node  $(g, f)$  into triples  $(Z_{s1}, Z_{s2}, Z_t)$  minimizing the distance  $SD_3(Z_{s1}, Z_{s2}, Z_t) = sd(Z_{s1}, Z_{s2}) + sd(Z_{s1}, Z_t) + sd(Z_{s2}, Z_t)$  over all combinations of the remaining vectors at node  $(g, f)$  until one layer runs out of phonetic unit based vectors. Here  $sd$  represents a spectral distortion measure. Consequently, we end up with an equal number of phonetic unit based vectors in each layer. A node  $(g, f)$  in which at least one of  $\theta_g$  or  $\theta_f$  is a *rare* phoneme cannot contain data from the target speaker, therefore we align

phonetic unit based vectors as pairs  $(Z_{s1}, Z_{s2})$  only between source speakers’ layers. This is done in a similar way minimizing the distance  $sd(Z_{s1}, Z_{s2})$  over all the remaining combinations until one layer runs out of data.

### 3.3. Contextual Modeling

The traditional GMM based voice conversion methods fit a GMM to the aligned training data globally without any explicit consideration of the various phoneme classes. It is natural to question whether GMM is able to capture the fine details of each phonetic class when the training optimizes a global fitting. It is also natural to wonder whether these details are influenced or not by the local context. It is not practical though to train a different model for each different context or even for each different phoneme due to the large amount of data necessary for such training. The research conducted so far has not been able to give clear answers to these questions.

To shed some light on the above issues more closely, we have studied the use of contextual modeling in voice conversion. By contextual modeling we refer to a scheme in which multiple models are optimized on possibly overlapping subsets of the training data denoted as contexts. We hypothesize that such a modeling could potentially offer more accuracy and partially alleviate the known over-smoothing problem of the traditional GMM based techniques.

Each feature vector  $y_i$  in the parameterized speech sequence  $[y_1, y_2, \dots, y_N]$  can be regarded as belonging to a context and is associated with a context descriptor  $\alpha_i$ . For simplicity  $\alpha_i$  can be regarded as the phonetic unit to which  $y_i$  belongs but in a broader sense the context descriptor can be any meaningful parameter (e.g.  $dy/dt$ , the time derivative of  $y$ ).

For the conversion of a feature vector  $y_i$  we first select the appropriate conversion model based on its context descriptor  $\alpha_i$ . A potentially different model is selected for the conversion of a different feature vector  $y_j$ .

Since it is not practical to train and store models for thousands of contexts beforehand, we can perform model training on context data selected on the fly for each feature vector  $y_i$  based on  $\alpha_i$ .

Context data may be considerably small depending on the selection rule (it is not practical to gather sufficient data to train e.g. a reliable phoneme model) therefore the trained models need to be robust with small data, fast and computationally efficient because they are trained repeatedly on different contexts. Our results presented in [11] recommend bilinear models for this task.

### 3.4. Practical Implementation

The proposed algorithm requires aligned event target representations of the test utterance from all the source

speakers. Furthermore we use phonetic annotations in order to segment the aligned representations into phonetic units as defined in Section 3.2. Blocks of  $S \times Q_{pu}$  event targets representing one phonetic unit are converted one at a time generating  $Q_{pu}$  converted event targets as suggested in **Figure 2** below, with  $S$  being the number of source speakers and  $Q_{pu} = 4$  the number of equally spaced event targets used to represent one phonetic unit.

Let  $\theta_g, \theta_f \in \Theta$  be the  $(j - 1)$ -th and  $j$ -th phonemes of the test utterance, alternatively denoted as  $p_{j-1}$  and  $p_j$  respectively. We note that each phonetic unit (e.g.  $p_{j-1}p_j = \theta_g\theta_f$ ) corresponds to a node of the matrix  $D$  (e.g.  $(g, f)$ ) representing the full training data as introduced in Section 3.2.

For each phonetic unit of the test utterance a context data  $context_{common}$  is extracted from the full training set using the multilayer matrix structure  $D$ . To illustrate the selection we describe next how this is done for the phonetic unit  $p_{j-1}p_j = \theta_g\theta_f$ .

- 1) Start with an empty  $context$  data.  $context = \emptyset$ .
- 2) Add the data corresponding to the current phonetic unit ( $p_{j-1}p_j = \theta_g\theta_f$ ).  $context = context \cup D(g, f)$ .
- 3) If  $size(context_{common}) < Thr$  then  $context = context \cup D(k, f)$ ,  $1 \leq k \leq P$ ,  $k \neq g$  and  $context = context \cup D(g, k)$ ,  $1 \leq k \leq P$ ,  $k \neq f$ . By *common* data we refer to any  $D(l, m)$  for which both  $\theta_l$  and  $\theta_m$  represent phonemes common to both source and target speakers,  $context_{common}$  represents the common part of the data included in the current *con-*

*text*, and  $Thr$  denotes a size threshold.

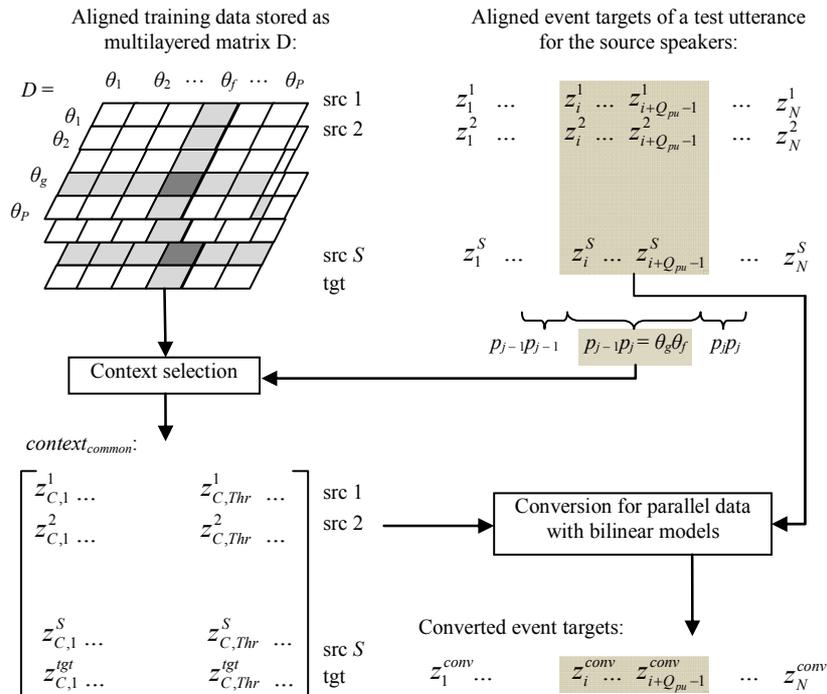
- 4) For  $p_{j-1}p_{j-1}, p_jp_j, p_{j-2}p_{j-1}, p_jp_{j+1}, p_{j-2}p_{j-2}, p_{j+1}p_{j+1} \dots$  until  $size(context_{common}) \geq Thr$  do step 2 and step 3 (if such a unit is within the utterance bounds), but in the context building skip the nodes of  $D$  that have already been collected.

By construction  $context_{common}$  is an aligned dataset of event targets of all the  $S$  source speakers and the target speaker. The block of source event targets corresponding to the phonetic unit for which  $context_{common}$  was built can be converted using this context data and the bilinear models framework for parallel data from Section 2.

After the conversion of the event targets the desired number of feature vectors can be reconstructed using event functions.

### 4. Experiments and Results

This work extends the study of voice conversion with bilinear models from the case of parallel and limited training sets [11] to non-parallel and simulated cross-lingual cases evaluating how the size of the training data and the contextual modeling influence the performance. Unlike in [11], the bilinear model is now compared against a GMM whose number of mixture components is optimized for the amount of available training data. Both objective metrics and listening test results are used. The GMM is chosen as a reference because it has been well studied and its performance level should be familiar in the field of voice conversion.



**Figure 2.** Context selection for the current phonetic unit and the conversion of its corresponding block of event vectors.

#### 4.1. The Experimental Set-up

The present study is concerned only with the spectral conversion and does not discuss prosodic nor energy conversion. We use 16-dimensional LSF vectors for the representation of the spectral envelope as proposed in [22]. LSFs relate closely to formant frequencies but unlike formant frequencies they can be reliably estimated [23,24]. They have also favorable interpolation properties and local spectral sensitivity which means that a badly estimated component affects only a small portion of the spectrum around that frequency [25,26]. Interestingly, LSFs have also been used with MRTD due to these beneficial characteristics.

We used in our experiments two source speakers (male and female) and one target speaker (male) selected out of four US English speakers available in the CMU Arctic database. The Arctic database is a parallel corpus of 16 kHz speech samples provided with phonetic labels and it is publicly available at [http://festvox.org/cmu\\_arctic/](http://festvox.org/cmu_arctic/). The samples consist of short utterances with an average duration of 3 seconds.

The number of three speakers is not meant to be an optimal lower limit, they were chosen with the purpose of ensuring sufficiently large and phonetically balanced text independent partitions of this parallel database. Another criterion was to have an equal number of male and female source speakers. It is assumed ([7]) that an increased number of speakers would be beneficial for the proposed bilinear method leading to a better separation of the style and content factors.

Phonetically balanced sets of utterances were selected from each speaker to form parallel, non-parallel and simulated cross-lingual training corpora with 3, 10, 70, 140 and 264 utterances.

In parallel and non-parallel training data all speakers cover the full US English phoneme set but in the case of simulated cross-lingual data only the two source speakers use the full phoneme set. We simulate a cross-lingual corpus by defining a set of 5 *rare* phonemes and selecting in the training data only target utterances in which none of the *rare* phonemes occurs. The benefit from doing so is that, unlike in the real cross-lingual case, we can evaluate the conversion of phonemes *unseen* in the target training data against real target instances of these *rare* phonemes. The selected *rare* phonemes are those with the lowest rate of occurrence in the database: “zh” as in “mirage”/m-er-aa-zh, “oy” as in “joy”/jh-oy, “uh” as in “could”/k-uh-d, “ch” as in “charge”/ch-aa-r-jh, “th” as in “author”/ao-th-er. This selection attempts to make efficient use of the full parallel data by maximizing the size of its cross-lingual partition and does not guarantee a minimal acoustic similarity between the *rare* phonemes

and other *common* phonemes used in training. The resemblance is possible to some extent (e.g. “th”  $\approx$  “t”) but seems to be rather limited. In our study the transcriptions are assumed to be accurate and no special handling is provided for pronunciation differences.

The alignment of the LSF vectors from parallel data is accomplished using dynamic time warping (DTW) on Mel-frequency cepstral coefficients (MFCC) extracted at the same time locations as the LSFs. For non-parallel and simulated cross-lingual data, event target vectors are aligned following the procedure described in Section 3.2.

The bilinear method presented in Section 2 and the contextual modeling methods described in Section 3.3 are compared against a modified GMM based method. The modified GMM method uses data from two source speakers to predict the target speaker’s voice in the same way as the original GMM method uses data from one source speaker to predict the target voice. Our tests indicate that the modified approach outperforms the original model in terms of mean squared error. The modified method requires aligned data from the three speakers to train a conversion model whose input is a concatenation of two aligned feature vectors from the two source speeches and whose output is a feature vector of the target speech. With the above specification the GMM training and conversion are done as described in [1]. It is worth to observe that in the simulated cross-lingual case only the *common* phonemes are represented in the data used to train the GMM. To keep comparisons between GMMs meaningful, the initialization of the GMM training is done always from the same list of data points in the same order. This way two GMMs with the same number of mixtures trained on different datasets would still be initialized identically.

To simplify the alignment in the test set, but also for a more meaningful evaluation of the conversion result, we design the test set as a phonetically balanced set of ten parallel utterances covering the entire phoneme set (including the *rare* phonemes). Including the rare phonemes is important especially for the evaluation of the simulated cross-lingual voice conversion.

Even though in real applications the test sentence does not exist in the target voice, in our study such an utterance exists and is used to align the test utterances of the source speakers to the speaking rate of the target speaker. This facilitates distance measurements in the feature domain between the converted LSF and real target LSF vectors and allows the converted LSFs to be used along with the rest of the original target parameters for the synthesis of a converted waveform. Hence the converted waveforms mimic the case when all other features except LSFs are ideally converted allowing the evaluation to more effectively focus on the performance of the spectral

LSF conversion.

The contextual model experiment is run only once for the largest cross-lingual dataset (264 utterances) which is believed to ensure sufficient data for the training contexts. The conversion is done one phonetic unit at a time and for every phonetic unit a context is built by requiring at least 1000 aligned *common* frames (event targets). The size of 1000 was selected based on preliminary experiments and corresponds usually to 2-3 neighboring phonetic units in the speech sequence (*i.e.* *offset* =  $\pm 1$  or  $\pm 2$ ). About 3 h 40 min of contextual training was needed for the conversion of a 3 sec test utterance with a simulated cross-lingual training set of 264 utterances. For the same data the typical time required to train a GMM with 8 mixtures is 2 min while a bilinear model or a GMM with 1 mixture take about 2 sec to train. The times are reported for an Intel Core2 CPU 6300 @ 1.86 GHz with 1 Gb of memory.

## 4.2. Metrics for Objective Evaluation

The first objective metric used is the *mean squared error* (MSE) which is computed between a converted and a target LSF vector using the formula

$$MSE(lsf_c, lsf_t) = \frac{\sum_{i=1}^N (lsf_c(i) - lsf_t(i))^2}{N} \quad (22)$$

where  $lsf_c$  and  $lsf_t$  denote the converted and target LSF vectors and  $N$  represents the LSF order. The frame-wise MSE figures are then averaged over the entire test data.

*Spectral distortion* (SD) is computed between a converted spectral envelope (derived from the converted LSF) and the corresponding target spectral envelope. The SD is measured only for a selected frequency range of the spectrum, using

$$SD^2 = \frac{1}{(f_u - f_l)} \int_{f_l}^{f_u} \left( 20 \log_{10} \frac{|H(e^{j2\pi f/f_s})|}{|\hat{H}(e^{j2\pi f/f_s})|} \right)^2 df \quad (23)$$

where  $H$  and  $\hat{H}$  represent the target and converted spectra, respectively,  $f_s$  is the sampling frequency, and  $f_l$  and  $f_u$  denote the frequency limits of the integration. For better perceptual relevance, SD is computed between 0 and 4 kHz.

## 4.3. The Relationship between the Size of the Training Data and the Number of Mixture Components for GMMs

We studied with parallel training data how the GMM performance is related to the number of mixtures and the size of the training set.

For reduced datasets (3 utts.) the best GMM perform-

ance is attained using one mixture component. Objective results in **Figure 3** and indeed perceptual ones presented in Section 4.5, indicate a close tie between this configuration and the bilinear approach.

On the other hand, four mixtures achieve optimal or close to optimal performance for larger sets (70 - 264 utts.). With 264 utts. for instance, a degradation to a lesser or larger extent is produced for less than 4 or more than 8 mixture components.

It is difficult to know beforehand what number of mixture components is optimal for a given amount of training data. Too few components, although reliably estimated, would give an inaccurate approximation of the training data while estimating too many components becomes unreliable and may cause over-fitting problems.

The result obtained with bilinear models was superimposed in **Figure 3** for comparison revealing an interesting similarity with the one mixture case of the GMM. Both models outperform all other GMM configurations for small training sets but remain slightly behind for large data. It is worth to notice that the proposed bilinear model does not require preliminary order tuning.

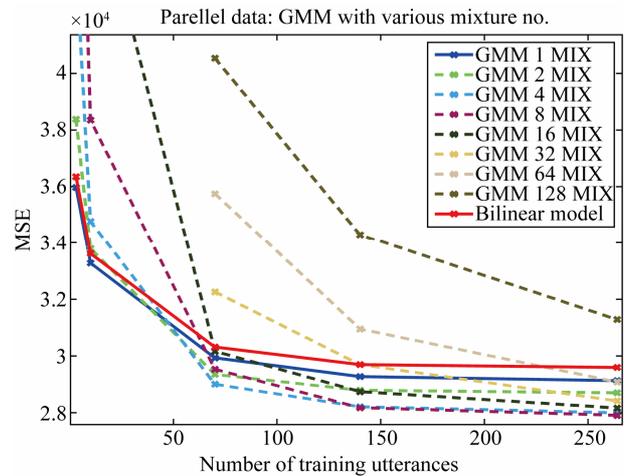
The result presented in this section was used to determine an “optimal” number of mixture components for the GMMs involved in the next sections depending on the amount of aligned LSF vectors in the training data.

## 4.4. Objective Results

The objective results obtained for a training set of 3 parallel utterances are shown in **Table 2**. The “optimal” number of components for GMM in this case is 1.

It is worth observing that these figures are extremely close.

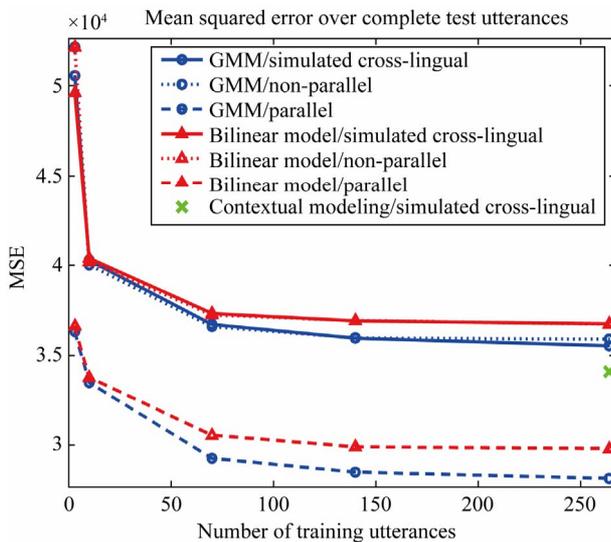
**Figure 4** presents MSE for both GMM and bilinear



**Figure 3. MSE for GMMs with different mixture numbers and training sizes demonstrated with parallel training data; a similar figure for the bilinear approach is superimposed.**

**Table 2. Mean squared error (MSE) and spectral distortion (SD) results for 3 parallel training utterances**

	Bilinear model	GMM (1 mix)
MSE	36625	36329
SD (dB)	5.51	5.50

**Figure 4. Mean squared error results over the set of test utterances.**

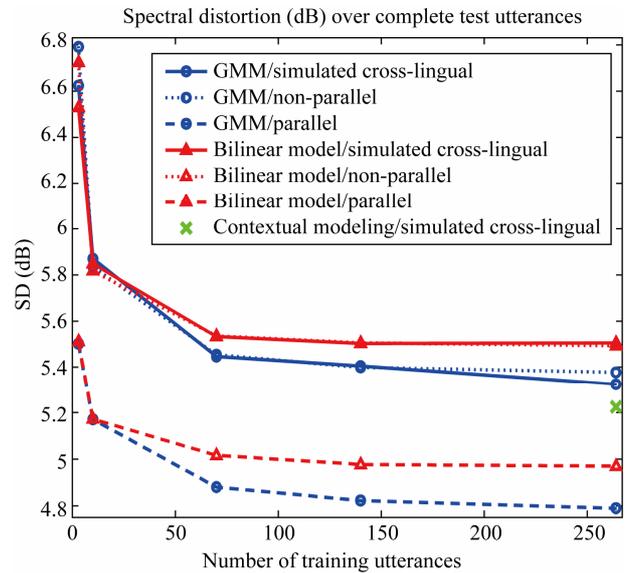
methods for parallel, non-parallel and simulated cross-lingual cases and for various sizes of the training data. The contextual modeling was evaluated only for 264 simulated cross-lingual utterances.

For GMMs, the “optimal” mixture numbers corresponding to 3, 10, 70, 140, 264 utterances were found to be 1, 1, 4, 8, 8 for parallel data and 1, 1, 4, 4, 8 for non-parallel and cross-lingual data.

The two techniques compare to each other similarly in all three scenarios. Their objective performance is very similar for small training sets while the “optimal” GMM gains some advantage for larger training sets. It is important to observe that this performance gain for the GMM is obtained at the cost of increased computational complexity corresponding to a larger number of mixture components. As reflected in Section 4.5 in the listening tests this difference in objective measurements seems to be very small from a perceptual point of view.

We can also see that the contextual modeling brings a sensible improvement compared to the “optimal” GMM and bilinear models fitted globally on full cross-lingual training data (264 utterances).

**Figure 5** shows the corresponding spectral distortion results. An interesting aspect to note is that the minimum spectral distortion with 264 utterances is attained for the GMM method with parallel data (4.79 dB) while the

**Figure 5. Spectral distortion (dB) results over the set of test utterances.**

maximum is 5.50 dB recorded for the bilinear approach with non-parallel data. The gap of only 0.71 dB is perceptually small and this was also reflected in the listening tests.

**Figures 6** and **7** present consistent MSE and SD results for the conversion of the *rare/unseen* phonemes. It is interesting to observe in the simulated cross-lingual experiment the capability to restore phonemes *unseen* in the training data (*rare* phonemes). In the bottom plots of **Figures 6** and **7**, we observe that the bilinear approach and the GMM based method perform similarly independent of the size of training data. By comparison with the cross-lingual results over complete utterances presented in **Figures 4** and **5**, it is worth noticing that the error over the *unseen* phonemes is significantly larger.

The top and middle plots indicate for GMM and the bilinear method respectively that the accuracy of reconstruction is not depending much on whether the phoneme exists or not in the training data of the target speaker (minor differences between the results with non-parallel data including *rare* phonemes and those with simulated cross-lingual data lacking them) but rather on the alignment and type of data (parallel or text independent). Better reconstruction results are obtained with parallel data which is also an indicator of the best performance that could be achieved due to its precise alignment and because the *rare* phonemes are included in the training. By comparison with the results in **Figures 4** and **5** we notice that the gap between figures for *rare* phonemes and complete utterances is significantly smaller for the parallel case than it is for the nonparallel and cross-lingual cases.

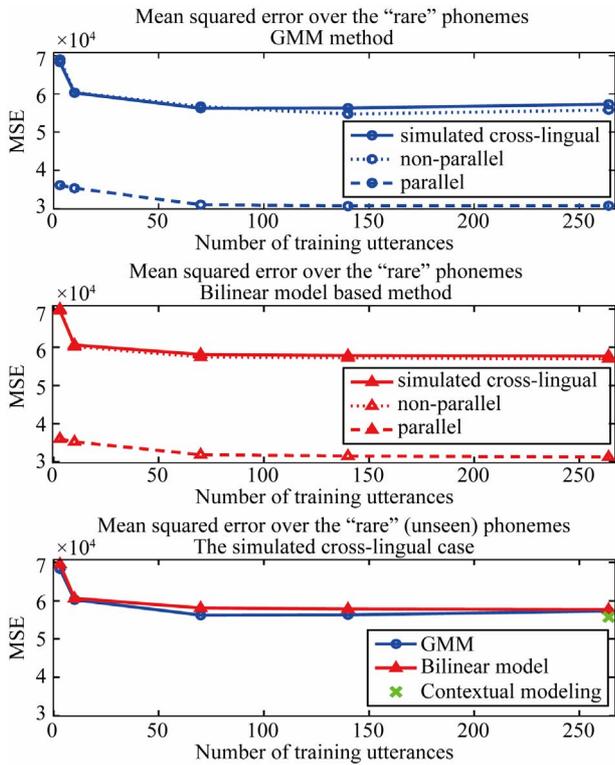


Figure 6. Mean squared error results over the rare/unseen phonemes existent in the test utterances.

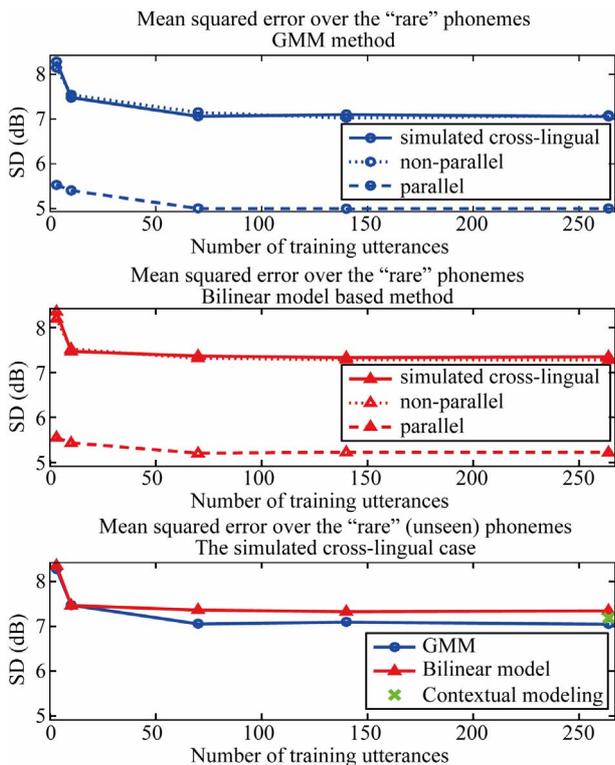


Figure 7. Spectral distortion (dB) results over the rare/unseen phonemes existent in the test utterances.

Interestingly, the result of the contextual modeling for the reconstruction of *unseen* phonemes is very similar to those obtained for the globally optimized GMM or bilinear models. This result is surprising considering that the missing phonemes are reconstructed based only on very small contexts of phonetic units. The bilinear model seems to be capable to generalize from a reduced subset of training data almost as well as it does when using the full data for training.

Figures 8 and 9 illustrate a direct comparison between the two methods for every conversion scenario separately by showing again MSE and SD results measured over the entire test set.

Independent of the scenario, the performance of the bilinear models is very similar to that of the “optimal” GMM particularly for small training sets. While the objective results show a small performance advantage of the GMM for larger training sets, the subjective listening results presented in Section 4.5 indicate that the methods are still very close perceptually even for large datasets.

The relatively small performance difference can be explained by observing the similarity in the MSE criteria that both methods optimize. The bilinear models optimize the criterion in (6) whereas the GMM optimizes a similar mean squared error criterion between the converted and target feature vectors. An interesting finding visible in the top panel reveals that contextual modeling slightly outperforms the two techniques based on globally optimized models. This confirms that contextual approach may indeed capture details better than globally optimized models even though the gain does not justify the additional computational effort.

Finally, for each method (GMM and bilinear) we compared between three conversion scenarios: parallel, non-parallel and simulated cross-lingual (in Figures 10 and 11 in terms of MSE and SD, respectively).

First, we notice that each method taken separately obtained very similar results for the non-parallel and simulated cross-lingual scenarios. This finding, in line with a similar result in the recovery of *rare/unseen* phonemes, indicates that the presence (in non-parallel data) or absence (from cross-lingual data) of the *rare* phonemes did not have a major influence on the results.

Secondly, both GMM and bilinear approaches perform clearly better with parallel training data than in the non-parallel or simulated cross-lingual cases and the difference is bigger for the small training sets (3 utts.).

The SD results shown in Figure 11 are again in line with the MSE scores presented above.

A concluding remark on the objective measurement experiments is that the performance of both systems is influenced by the amount of training data only up to a point beyond which adding more data does not bring

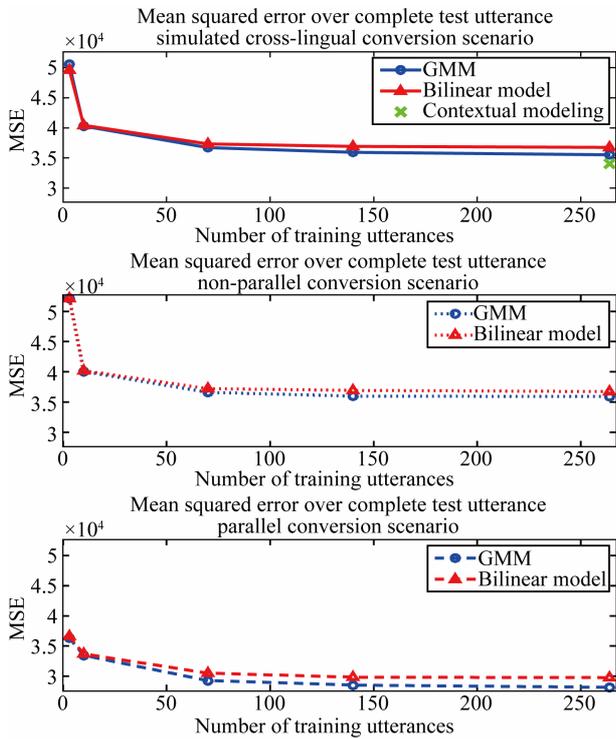


Figure 8. Comparative mean squared error results for the GMM, bilinear approach and contextual modeling in different conversion scenarios.

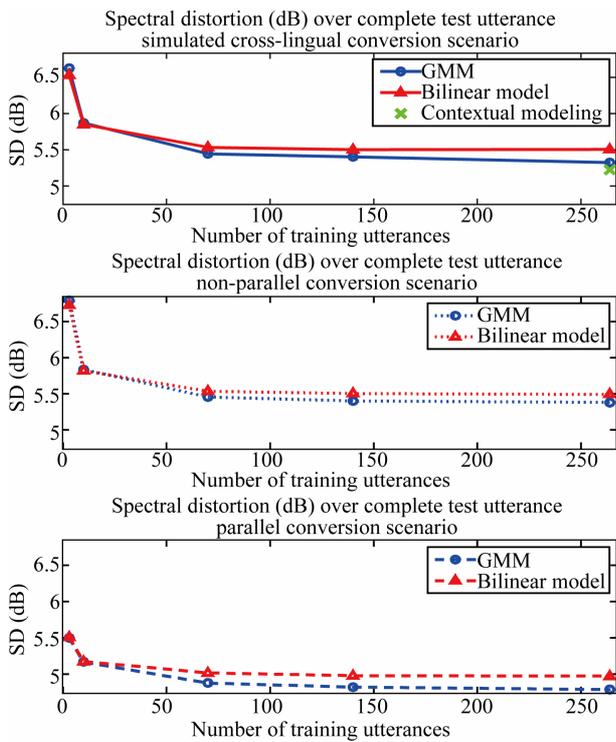


Figure 9. Comparative spectral distortion (dB) results for the GMM, bilinear approach and contextual modeling in different conversion scenarios.

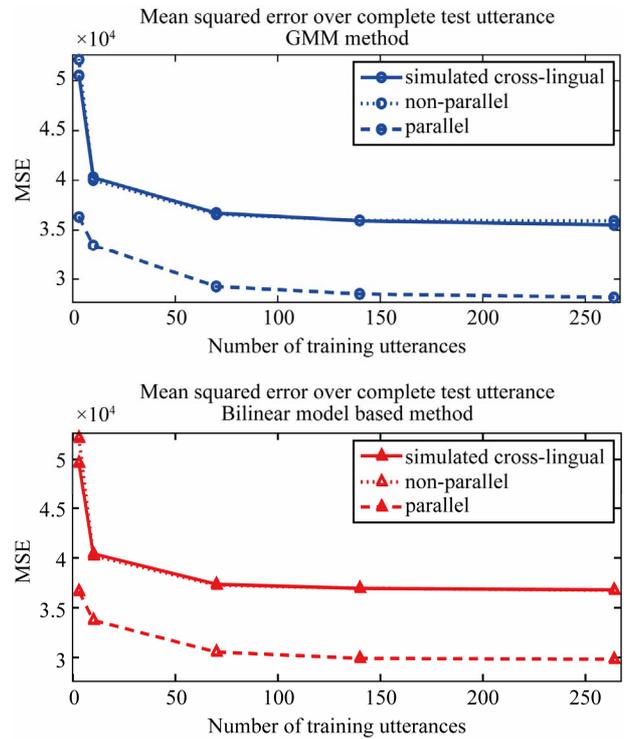


Figure 10. Comparative mean squared error results between different conversion scenarios for the GMM and bilinear approach.

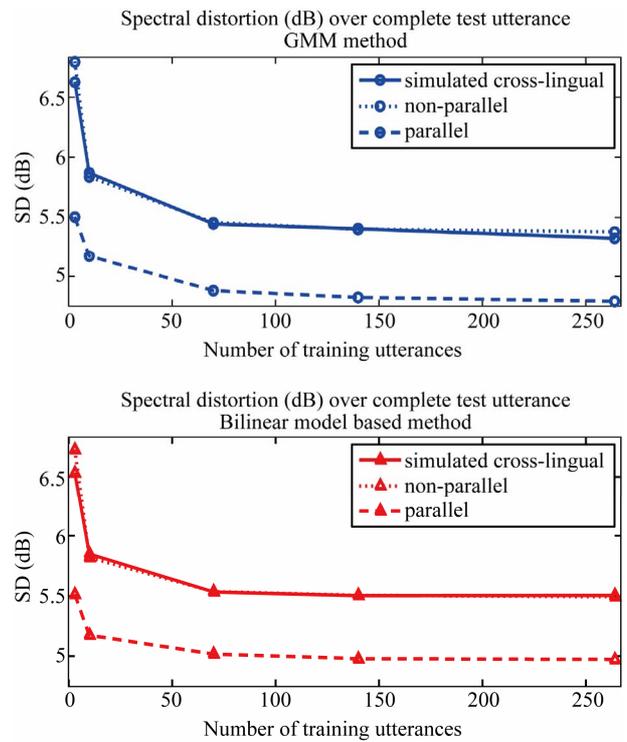


Figure 11. Comparative spectral distortion (dB) results between different conversion scenarios for the GMM and bilinear approach.

significant improvements of the performance. We also note that it is the size of the actual aligned data that influences the performance and not the number of training sentences. A training data consisting of text independent utterances will result in significantly less aligned data than the same number of parallel utterances using our alignment technique. This also explains the bigger differences between parallel and text independent scenarios in the range 3 to 10 training utterances.

#### 4.5. Listening Tests

For a meaningful validation of the objective measurements we present subjective results with both reduced (3 utterances) and large training sets (264 utterances).

The first test compares the bilinear and GMM methods for a training set of 3 parallel utterances. One mixture component is used for the GMM as found “optimal” for the data size.

The next tests are concerned with large training sets of 264 utterances (approx. 3 sec per utterance) and use GMMs with 8 mixture components. They evaluate the GMM and bilinear methods relative to one another using parallel or simulated cross-lingual training data but also evaluate how these two scenarios (parallel and simulated cross-lingual) compare to each other for each of the two methods. In the last test the contextual modeling is compared with the GMM based method for the cross-lingual training data. The results with 95% confidence intervals are given in **Table 3**.

In each test, ten listeners compare schemes A and B using ten test utterances and a modified MOS test. In the identity test a real target version of the test sentence is compared in terms of voice identity with the converted samples obtained with schemes A and B. The quality test is simply a comparison in terms of speech quality between the two converted samples A and B. The successfulness of identity conversion and the overall speech quality are evaluated separately with scores between -2 (scheme A is much better than B) and 2 (scheme B is much better than A). The 0 (zero) score is given for perceptually identical performance.

The first result in **Table 3** represents a comparison between the GMM based method with one mixture component and the bilinear approach for a training set of 3 parallel utterances. The very balanced score and its 95% confidence intervals indicate very similar performances for the two methods. This is in line with the objective results pointing out that the methods tend to have identical performance for small training sets. The SD figure shows a 0.01 dB difference (5.50 dB for GMM and 5.51 dB for the bilinear approach) which is not perceivable by humans.

Results for large training sets of 264 utterances are

**Table 3. Subjective listening test results.**

Utts.	A	B	Quality	Identity
3	GMM/parallel	BL/parallel	$0.02 \pm 0.08$	$0.01 \pm 0.07$
264	GMM/parallel	BL/parallel	$-0.08 \pm 0.12$	$-0.02 \pm 0.09$
264	GMM/simulated cross-lingual	BL/simulated cross-lingual	$-0.12 \pm 0.12$	$-0.05 \pm 0.09$
264	GMM/parallel	GMM/simulated cross-lingual	$-0.17 \pm 0.12$	$-0.11 \pm 0.10$
264	BL/parallel	BL/simulated cross-lingual	$-0.17 \pm 0.13$	$-0.01 \pm 0.10$
264	GMM/simulated cross-lingual	CM/simulated cross-lingual	$-0.05 \pm 0.12$	$-0.03 \pm 0.10$

presented on lines 2 to 6 as follows. The second line compares the bilinear model and the 8-mixture GMM method found “optimal” for the given parallel data on which both methods are trained. The 95% confidence interval could not indicate a clear winner showing that the methods are perceptually equivalent. The SD difference of 0.17 dB (4.79 dB for GMM and 4.97 dB for bilinear) is hardly observable by the human hearing.

On the third line, the result obtained for the simulated cross-lingual scenario is slightly in favor of the GMM but the perceptual difference seems to be, however, very small. The exact 95% confidence interval actually extends by 0.0006 to the other side of the 0 axis, so in a strict sense it is impossible to call a winner.

The fourth and fifth results of **Table 3** indicate that both the GMM based method and the bilinear approach perform clearly better with parallel data than with simulated cross-lingual data but interestingly the difference is very small. With -1 indicating that the parallel case is (clearly) better, and -2 for much better, our scores of  $\approx -0.17$  could be interpreted as “only slightly better”. This suggests that the type of the data may not be the essential factor for conversion as long as we have an efficient alignment scheme and that the proposed alignment scheme has been successful.

Finally, the last result of **Table 3** represents a comparison between the “optimal” (8-mixture) GMM based method and the contextual modeling technique in the simulated cross-lingual case. Not surprisingly the 0.1 dB margin by which the context method outperforms the GMM approach is perceptually insignificant and the listening test result is consistent with this objective finding, indicating that it is practically impossible to decide a winner.

The listening test results are largely consistent with the objective measurements additionally revealing that the objective differences between the bilinear model and GMM, especially for large datasets, are very small or

insignificant from a perceptual point of view. The small perceptual difference between parallel and cross-lingual scenarios is an indication of efficiency for our text-independent alignment. On the other hand the listening tests demonstrate that contextual modeling did not bring a perceptually meaningful gain.

## 5. Conclusions

This paper presented a comprehensive study of bilinear models applied in voice transformation and explored their capability to reconstruct phonetic content in a new voice. The paper also proposed a new conversion technique called contextual modeling that benefits from the efficient computation algorithms and the robust performance of the bilinear models with reduced data.

Objective and subjective evaluations of the bilinear model were reported in relationship to the traditional GMM-based technique with “optimal” number of mixture components determined based on the size of the training data. The objective figures of the two methods are particularly close in the range of small data while for larger sets the GMM seems to gain advantage. However, the listening tests indicated that the two methods perform equivalently or comparable from a perceptual point of view for both small and large training sets.

The gain in objective performance of the GMM for large data is achieved at the cost of an important increase of the computational complexity due to a larger number of mixture components. It is worth noticing that the bilinear model does not need any tuning.

Section 4.3 and [11] suggest that the bilinear model may have an important advantage in the range of small datasets over GMMs with more than one mixture component, both objectively and subjectively. This is demonstrated in [11] for a GMM with 4 mixtures. Section 4.3 also reflects with objective figures an interesting similarity between the bilinear model and a GMM with only one mixture.

The reconstruction capability of *unseen* data appears to be similar for the two methods independently of the data size.

Both in a global evaluation over the entire test set and exclusively over the *rare* phonetic units the non-parallel result is very similar to the simulated cross-lingual result, leading to an interesting finding that the performance is not influenced or influenced only marginally by incomplete data if sufficient training data is provided and sufficient *common* phonetic units are represented in the training data.

The performance seems to be much more affected by the type of data (parallel or text independent). Both methods perform better with parallel data than they do with text independent data but as the amount of data is in-

creased the differences reduce for each of the two methods. The perceptual closeness between parallel and cross-lingual scenarios with large data is also reflected in listening tests which gave scores of only  $-0.17$  (on a scale  $-2$  to  $2$ ) in favor of the parallel case. Such a small difference between parallel and text independent results indicates a certain degree of efficiency of the proposed alignment scheme.

The contextual modeling is conceptually interesting and obtained slightly better results than the other methods. Our experiments answer the questions posed in Section 3.3 showing that a contextual modeling can be better than models optimized globally on the full training data. We could not find clear evidence that the contextual modeling solved the over-smoothing problem. In fact, it could be argued that over-smoothing is partially caused by the MSE based criteria optimized by all these methods, in the sense that such criteria do not focus on details but on averages. In contextual modeling, however, this averaging is applied to a restricted “local” dataset.

Future research could try finding and optimizing a perceptually motivated criterion or study new ways to separate style and content in speech, e.g. by modeling it as product of more than two underlying factors.

## REFERENCES

- [1] A. Kain and M. Macon, “Spectral Voice Conversion for Text-to-Speech Synthesis,” *Proceedings of International conference on Acoustics, Speech and Signal Processing*, Vol. 1, Seattle, 12-15 May 1998, pp. 285-288.
- [2] Z. Shuang, R. Bakis and Y. Qin, “Voice Conversion Based on Mapping Formants,” *Proceeding of TC-STAR Workshop on Speech-to-Speech Translation*, Barcelona, 19-20 June 2006, pp. 219-223.
- [3] M. Narendranath, H. Murthy, S. Rajendran and N. Yegnanarayana, “Transformation of Formants for Voice Conversion Using Artificial Neural Networks,” *Speech Communication*, Vol. 16, No. 2, 1995, pp. 207-216. [doi:10.1016/0167-6393\(94\)00058-1](https://doi.org/10.1016/0167-6393(94)00058-1)
- [4] E. K. Kim, S. Lee and Y. Oh, “Hidden Markov Model Based Voice Conversion Using Dynamic Characteristics of Speaker,” *5th Proceedings of European Conference on Speech Communication and Technology*, Rhodes, 1997.
- [5] Y. Stylianou, O. Cappe and E. Moulines, “Continuous Probabilistic Transform for Voice Conversion,” *IEEE Transaction on Speech and Audio Processing*, Vol. 6, No. 2, 1998, pp. 131-142. [doi:10.1109/89.661472](https://doi.org/10.1109/89.661472)
- [6] L. Arslan and D. Talkin, “Voice Conversion by Codebook Mapping of Line Spectral Frequencies and Excitation Spectrum,” *5th Proceedings of European Conference on Speech Communication and Technology*, Rhodes, 1997.
- [7] T. Toda, Y. Ohtani and K. Shikano, “Eigenvoice Conversion Based on Gaussian Mixture Model,” *Proceedings of*

- ICSLP*, Pittsburgh, September 2006, pp. 2446- 2449.
- [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, 2007, pp. 2222-2235. [doi:10.1109/TASL.2007.907344](https://doi.org/10.1109/TASL.2007.907344)
- [9] E. Helander, T. Virtanen, J. Nurminen and M. Gabbouj, "Voice Conversion Using Partial Least Squares Regression," *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 18, No. 5, 2010, pp. 912-921. [doi:10.1109/TASL.2010.2041699](https://doi.org/10.1109/TASL.2010.2041699)
- [10] J. B. Tenenbaum and W. T. Freeman, "Separating Style and Content with Bilinear Models," *Neural Computation*, Vol. 12, No. 6, 2000, pp. 1247-1283. [doi:10.1162/089976600300015349](https://doi.org/10.1162/089976600300015349)
- [11] V. Popa, J. Nurminen and M. Gabbouj, "A Novel Technique for Voice Conversion Based on Style and Content Decomposition with Bilinear Models," *Interspeech 2009*, Brighton, 6-10 September 2009.
- [12] B. P. Nguyen, "Studies on Spectral Modification in Voice Transformation," Ph.D. Thesis, School of Information Science, Japan Advanced Institute of Science and Technology, Japan, March 2009.
- [13] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of Royal Statistical Society B*, Vol. 39, No. 1, 1977, pp. 1-38.
- [14] B. S. Atal, "Efficient Coding of LPC Parameters by Temporal Decomposition," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'83)*, 1983, pp. 81-84.
- [15] C. N. Athaudage, A. B. Brabley and M. Lech, "Optimization of a Temporal Decomposition Model of Speech," *Proceedings of the International Symposium on Signal Processing and Its Applications (ISSPA'99)*, Brisbane, 22-25 August 1999, pp. 471-474.
- [16] M. Niranjana and F. Fallside, "Temporal Decomposition: A Framework for Enhanced Speech Recognition," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'89)*, Glasgow, 23-26 May 1989, pp. 655-658.
- [17] P. J. Dix and G. Bloothoof, "A Breakpoint Analysis Procedure Based on Temporal Decomposition," *IEEE Transactions on Speech and Audio Processing*, Vol. 2, No. 1, 1994, pp. 9-17. [doi:10.1109/89.260329](https://doi.org/10.1109/89.260329)
- [18] P. C. Nguyen, T. Ochi and M. Akagi, "Modified Restricted Temporal Decomposition and Its Application to Low Bit Rate Speech Coding," *IEICE Transactions on Information and Systems*, Vol. E86-D, 2003, pp. 397-405.
- [19] P. C. Nguyen, M. Akagi and T. B. Ho, "Temporal Decomposition: A Promising Approach to VQ-Based Speaker Identification," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'03)*, Baltimore, 6-9 July 2003, pp. 184-187.
- [20] B. P. Nguyen, T. Shibata and M. Akagi, "High-Quality Analysis/Synthesis Method Based on Temporal Decomposition for Speech Modification," *Proceedings of the International Speech Communication Association (Interspeech'08)*, Brisbane, 22-26 September 2008, pp. 662-665.
- [21] T. Shibata and M. Akagi, "A Study on Voice Conversion Method for Synthesizing Stimuli to Perform Gender Perception Experiments of Speech," *Proceedings of the RISP International Workshop on Nonlinear Circuits and Signal Processing (NCSP'08)*, 2008, pp. 180-183.
- [22] J. Nurminen, V. Popa, J. Tian and I. Kiss, "A Parametric Approach for Voice Conversion," *Proceedings of TCSTAR Workshop on Speech-to-Speech Translation*, Barcelona, 19-21 June 2006, pp. 225-229.
- [23] E. Helander, J. Nurminen and M. Gabbouj, "Analysis of LSF Frame Selection in Voice Conversion," *International Conference on Speech and Computer*, 2007, pp. 651-656.
- [24] E. Helander, J. Nurminen and M. Gabbouj, "LSF Mapping for Voice Conversion with Very Small Training Sets," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'08)*, Las Vegas, 31 March - 4 April 2008, pp. 4669-4672.
- [25] K. K. Paliwal, "Interpolation Properties of Linear Prediction Parametric Representations," *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech'95)*, 1995, pp. 1029-1032.
- [26] K. K. Paliwal and B. S. Atal, "Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame," *IEEE Transactions on Speech and Audio Processing*, Vol. 1, No. 1, 1993, pp. 3-14.