Nonparallel Training for Voice Conversion Based on a Parameter Adaptation Approach

Athanasios Mouchtaris, *Member, IEEE*, Jan Van der Spiegel, *Fellow, IEEE*, and Paul Mueller

***Abstract*—The objective of voice conversion algorithms is to modify the speech by a particular source speaker so that it sounds as if spoken by a different target speaker. Current conversion algorithms employ a training procedure, during which the same utterances spoken by both the source and target speakers are needed for deriving the desired conversion parameters. Such a (parallel) corpus, is often difficult or impossible to collect. Here, we propose an algorithm that relaxes this constraint, i.e., the training corpus does not necessarily contain the same utterances from both speakers. The proposed algorithm is based on speaker adaptation techniques, adapting the conversion parameters derived for a particular pair of speakers to a different pair, for which only a nonparallel corpus is available. We show that adaptation reduces the error obtained when simply applying the conversion parameters of one pair of speakers to another by a factor that can reach 30%. A speaker identification measure is also employed that more insightfully portrays the importance of adaptation, while listening tests confirm the success of our method. Both the objective and subjective tests employed, demonstrate that the proposed algorithm achieves comparable results with the ideal case when a parallel corpus is available.**

***Index Terms*—Gaussian mixture model, speaker adaptation, text-to-speech synthesis, voice conversion.**

I. Introduction

VOICE conversion methods attempt to modify the characteristics of speech by a given source speaker, so that it sounds as if it was spoken by a different target speaker. Applications for voice conversion include "personalization" of a text-to-speech (TTS) synthesis system so that it "speaks" with the voice of a particular person, as well as creating new voices for a TTS system without the need of retraining the system for every new voice. More generally, the work in voice conversion can be extended and find applications in many areas of speech processing where speaker individuality is of interest. As an example we mention interpreted telephony [1], user-centric speech enhancement [2], and possibly even speech compression.

A number of different approaches have been proposed for achieving voice conversion. Based on research results on speech individuality (we mention for example [3]), it is generally accepted that voice conversion can be sufficiently achieved by converting certain segmental and suprasegmental features of the source speaker into those of the target speaker. Various experiments have shown that an average transformation of the pitch and speaking rate of the source speaker can produce convincing conversion in the suprasegmental level (more details can be found in [1] and related references within), whereas most efforts in the area focus on the segmental level information. Early attempts were based on vector quantization (VQ) approaches, where a correspondence between the source and target spectral envelope codebooks is derived during the training phase [4]–[7], as well as artificial neural networks for deriving the spectral mapping of the source to the target formant locations (followed by a formant synthesizer) [8]. Regarding the VQ-based approaches, during the conversion phase the aforementioned correspondence is used for converting the source short-time spectral envelope into an estimated envelope that is close to the desired. The conversion is achieved as a linear combination of the target codebook centroids, which is a limited set of vectors and this results in limited spectral variability. Thus, while the conversion can be considered as successful, the resulting speech quality is degraded. Conversion methods based on Gaussian mixture models (GMMs) [1], [9], [10], while based on the same codebook philosophy, do not suffer from this drawback, since the conversion function is not merely a linear combination of a limited set of vectors as in the VQ case. Besides this advantage, GMMs have been successfully applied to modeling of the spectral envelope features of speech signals in the area of speaker identification [11], which is closely related with the area of voice conversion since speaker identity is of central importance. It should be noted that the short-time spectral envelope of the speech signal is modeled as a vector of few coefficients such as cepstral coefficients, line spectral frequencies (LSFs), etc. [12]. The modeling error, usually referred to as residual signal, contains important information for speech individuality and naturalness. As is the case for the majority of the research on voice conversion, we concentrate on modifications of the spectral vectors and do not attempt to modify the residual signal, due to its quasi-random nature. The interested reader is referred to [7], [13], and [14], for more information on this challenging subject.

The common characteristic of all the voice conversion approaches is that they focus on the short-term spectral properties of the speech signals, which they modify according to a

Manuscript received May 21, 2004; revised March 19, 2005. This work was supported by the Catalyst Foundation. The work of the first author was supported in part by a grant from the General Secretariat of Research and Technology of Greece (GSRT), Program ΕΠΑΝ Code 01EP111, and in part by the Marie Curie International Reintegration Grant within the 6th European Community Framework Programme under Contract 516093. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Nick Campbell.

A. Mouchtaris was with the Electrical and Systems Engineering Department, University of Pennsylvania, Philadelphia, PA 19104 USA. He is now with the Foundation for Research and Technology—Hellas (FORTH), Institute of Computer Science, Crete, GR-71110, Greece (e-mail: mouchtar@ieee.org).

J. Van der Spiegel is with the Electrical and Systems Engineering Department, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: jan@seas.upenn.edu).

P. Mueller is with Corticon, Inc., King of Prussia, PA 19406 USA (e-mail: cortion@aol.com).

Digital Object Identifier 10.1109/TSA.2005.857790

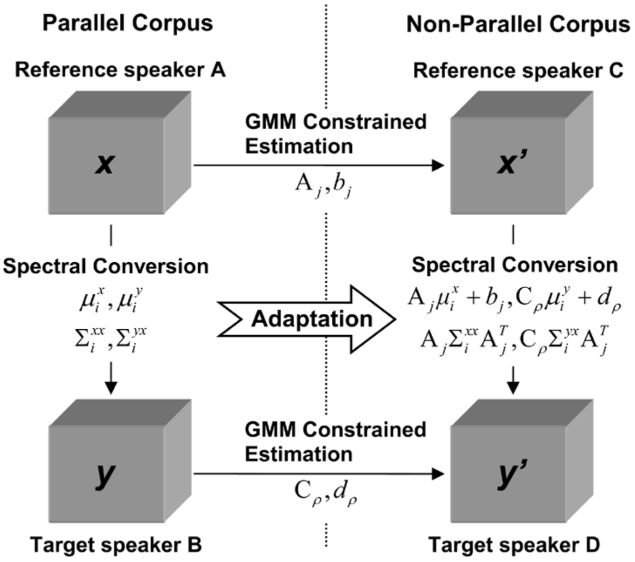


Fig. 1. Block diagram outlining spectral conversion for a parallel and nonparallel corpus. In the latter case, spectral conversion is preceded by adaptation of the derived parameters from the parallel corpus to the nonparallel corpus.

conversion function designed during the training phase. During training, the parameters of this conversion function are derived based on minimizing some error measure. In order to achieve this, however, a speech corpus is needed that contains the same utterances (words, sentences, etc.) from both the source and target speakers. The disadvantage of this method is that for many cases it is difficult or even impossible to collect such a corpus. If, for example, the desired source or target speaker is not a person directly available, it is evident that collecting such a corpus would probably be impossible, especially since a large number of data are needed in order to obtain meaningful results. This is especially important for possible extensions of voice conversion to other areas of speech processing, such as those briefly mentioned in the first paragraph of this section. Recently, an algorithm that attempted to address this issue was proposed [15], by concentrating on the phonemes spoken by the two speakers. The objective was to derive a conversion function that can transform the phonemes of the source speaker into the corresponding phonemes of the target speaker, thus not requiring a parallel corpus for training. However, accurately recognizing the phonemes spoken by the two speakers during training, as well as the phonemes spoken by the source speaker during conversion, is essential for this algorithm to operate correctly, and this can be a difficult requirement to meet in practice. Alternatively, phonemic transcriptions need to be available both during training and conversion as in [7].

Here we propose a conversion algorithm that relaxes the constraint of using a parallel corpus during training. Our approach, which is based on the first author's previous research on multichannel audio synthesis [16], [17], is to adapt the conversion parameters for a given pair of source and target speakers, to the particular pair of speakers for which no parallel corpus is available. Referring to Fig. 1, we assume that a parallel corpus is available for speakers A and B (in the left part of the diagram), and for this pair a conversion function is derived by employing one of the conversion methods that are given in the literature [9]. For the particular pair that we focus on, speakers C and D (in the right part of the diagram), a nonparallel corpus is available for training. Our approach is to adapt the conversion function derived for speakers A and B to speakers C and D, and use this new adapted conversion function for these speakers. Adaptation is achieved by relating the nonparallel corpus to the parallel corpus, as shown in the diagram and detailed in Sections II–IV. Note that the final result will depend on the initial error obtained by simply applying the conversion function for pair A-B to pair C-D, i.e., the error with no adaptation. Adaptation can improve on that error and reduce it significantly, but if this error is too large then the final result may not be as good as desired. Regarding the underlying model necessary for performing the required transformations of pitch and speech-rate, in our work the pitch-synchronous overlap-add (PSOLA) framework is applied [18], while it holds that the algorithm remains unaltered for any other model, such as sinusoidal models [19]–[21].

The adaptation among speaker pairs that, as explained in the previous paragraph, is central to our algorithm, is based on existing algorithms [22], [23] that have been developed for parameter adaptation within the speech recognition area. Parameter adaptation is important for speech recognition when there is a need for applying a recognition system to different conditions (speaker, environment, language) than those present during system training. Parameter adaptation allows for improving recognition performance in these cases, without the task of retraining the system for the new conditions. Parameter adaptation and voice conversion are highly related in many respects. Early work on parameter adaptation suggested using conversion methods as a means for adaptation (i.e., converting the source speaker characteristics into those of the target speaker for speaker adaptation) [24]–[27]. The disadvantage of these methods is that they require a parallel training corpus for achieving adaptation, which is something that is avoided in more recent adaptation algorithms. In our case, we attempt the opposite task, i.e., we are interested to apply adaptation methods to voice conversion, motivated by the fact that many recent adaptation algorithms do not require a parallel corpus. Since most of these algorithms (such as the one employed here) adapt the GMM parameters of a system and not directly the features, we found that our solution should be based on an existing set of GMM parameters for voice conversion, which can be available from a different conversion pair, as explained in the previous paragraph.

Finally, it is of interest to note that parameter adaptation has been used for voice conversion previously in the context of HMM speech synthesis [28], [29]. In that case, the method applies only in that particular context, i.e., synthesized speech by the particular HMM synthesis method. In our case, the proposed method applies to any recorded speech waveform, natural or synthesized.

The remainder of the paper is organized as follows. In Section II a description of the GMM-based spectral conversion of [9] is given, which is mostly of interest here. In Section III our algorithm for applying parameter adaptation to the voice conversion problem is described. The algorithm is based on multichannel audio synthesis research [16], [17], but it is presented here for completeness, especially since this algorithm was originally developed in a different context than speech synthesis. In

Section IV, results of the proposed algorithm are given based on both objective and subjective measures, with the goal of demonstrating that our algorithm for nonparallel voice conversion can achieve comparable performance with the parallel conversion algorithm on which it has been based. Finally, in Section V concluding remarks are made.

II. Spectral Conversion

Voice conversion in the segmental level is essentially achieved by spectral conversion. The objective of spectral conversion is to derive a function that can convert the short-term spectral properties of a reference waveform into those of a desired signal. A training dataset is created from the existing reference and the target speech waveforms by applying a short sliding window and extracting the parameters that model the short-term spectral envelope (in this paper we use the line spectral frequencies—LSFs—due to their desirable interpolation properties [9]). This procedure results in two vector sequences, $[\boldsymbol{x}_1\boldsymbol{x}_2\ldots\boldsymbol{x}_n]$ and $[\boldsymbol{y}_1\boldsymbol{y}_2\ldots\boldsymbol{y}_n]$, of reference and target spectral vectors respectively. A function $\mathcal{F}(\cdot)$ can be designed which, when applied to vector $\boldsymbol{x}_k$, produces a vector close in some sense to vector $\boldsymbol{y}_k$. Recent results have clearly demonstrated the superiority of the algorithms based on GMMs for the voice conversion problem [1], [9]. GMMs approximate the unknown probability density function (pdf) of a random vector $\boldsymbol{x}$ as a mixture of Gaussians whose parameters (mean vectors, covariance matrices, and prior probabilities of each Gaussian class), can be estimated from the observed data using the expectation–maximization (EM) algorithm [11]. A GMM is often collectively represented as $\{p(\omega_i), \boldsymbol{\mu}_i^x, \boldsymbol{\Sigma}_i^{xx}\}$ where ω_i denotes a particular Gaussian class $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_i^x, \boldsymbol{\Sigma}_i^{xx})$ (i.e., a Gaussian pdf with mean $\boldsymbol{\mu}_i^x$ and covariance $\boldsymbol{\Sigma}_i^{xx}$), and is given by the following equation:

$$g(\boldsymbol{x}) = \sum_{i=1}^{M} p(\omega_i)\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_i^x, \boldsymbol{\Sigma}_i^{xx}). \tag{1}$$

We focus on the spectral conversion method of [9], which offers great insight as to what the conversion parameters represent. Assuming that $\boldsymbol{x}$ and $\boldsymbol{y}$ are jointly Gaussian for each class ω_i, then, in mean-squared sense, the optimal choice for the function $\mathcal{F}$ is

$$\begin{aligned}\mathcal{F}(\boldsymbol{x}_k) &= E(\boldsymbol{y}|\boldsymbol{x}_k)\\ &= \sum_{i=1}^{M} p(\omega_i|\boldsymbol{x}_k)\left[\boldsymbol{\mu}_i^y + \boldsymbol{\Sigma}_i^{yx}\boldsymbol{\Sigma}_i^{xx^{-1}}(\boldsymbol{x}_k - \boldsymbol{\mu}_i^x)\right]\end{aligned} \tag{2}$$

where $E(\cdot)$ denotes the expectation operator and the conditional probabilities $p(\omega_i|\boldsymbol{x}_k)$ are given from

$$p(\omega_i|\boldsymbol{x}_k) = \frac{p(\omega_i)\mathcal{N}(\boldsymbol{x}_k; \boldsymbol{\mu}_i^x, \boldsymbol{\Sigma}_i^{xx})}{\sum_{j=1}^{M} p(\omega_j)\mathcal{N}(\boldsymbol{x}_k; \boldsymbol{\mu}_j^x, \boldsymbol{\Sigma}_j^{xx})}. \tag{3}$$

All the parameters in the two above equations are estimated using the EM algorithm on the joint model $\boldsymbol{z}$ of $\boldsymbol{x}$ and $\boldsymbol{y}$, i.e., $\boldsymbol{z} = [\boldsymbol{x}^T\boldsymbol{y}^T]^T$ (where ^{T} denotes transposition). In practice this means that the EM algorithm is performed during training on the sequence of concatenated vectors $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$. A time-alignment procedure is required in this case, and this is only possible when a parallel corpus is used.

Given the series of vectors $\boldsymbol{z}_k = \left[\boldsymbol{x}_k^T\boldsymbol{y}_k^T\right]^T$, the EM algorithm iteratively produces the maximum-likelihood estimates of the GMM for $\boldsymbol{z}$. For the convenience of the reader, we briefly review the basic formulas of the EM algorithm for a GMM pdf. The parameters needed to fully describe the pdf of $\boldsymbol{z}$ are the prior probabilities $p(\omega_i)$, the mean vectors $\boldsymbol{\mu}_i^z$, and the covariance matrices $\boldsymbol{\Sigma}_i^{zz}$, for each Gaussian class ω_i. The values of these parameters are initialized usually by a clustering procedure such as k-means. During the tth iteration of the EM algorithm, the expectation step of the algorithm (E Step) involves calculating the following conditional probabilities:

$$p^{(t)}(\omega_i|\boldsymbol{z}_k) = \frac{p^{(t)}(\omega_i)\mathcal{N}\left(\boldsymbol{z}_k; \boldsymbol{\mu}_i^{(t)z}, \boldsymbol{\Sigma}_i^{(t)zz}\right)}{\sum_{j=1}^{M} p^{(t)}(\omega_j)\mathcal{N}\left(\boldsymbol{z}_k; \boldsymbol{\mu}_j^{(t)z}, \boldsymbol{\Sigma}_j^{(t)zz}\right)}. \tag{4}$$

During the maximization step (M Step) that follows the E Step, the GMM parameters are reestimated and will be used at the E Step of the next ($t+1$th) iteration

$$p^{(t+1)}(\omega_i) = \frac{1}{n}\sum_{k=1}^{n} p^{(t)}(\omega_i|\boldsymbol{z}_k) \tag{5}$$

$$\boldsymbol{\mu}_i^{(t+1)z} = \frac{\sum_{k=1}^{n} p^{(t)}(\omega_i|\boldsymbol{z}_k)\boldsymbol{z}_k}{\sum_{k=1}^{n} p^{(t)}(\omega_i|\boldsymbol{z}_k)} \tag{6}$$

$$\begin{aligned}&\boldsymbol{\Sigma}_i^{(t+1)zz}\\ &\quad= \frac{\sum_{k=1}^{n} p^{(t)}(\omega_i|\boldsymbol{z}_k)\left(\boldsymbol{z}_k - \boldsymbol{\mu}_i^{(t+1)z}\right)\left(\boldsymbol{z}_k - \boldsymbol{\mu}_i^{(t+1)z}\right)^T}{\sum_{k=1}^{n} p^{(t)}(\omega_i|\boldsymbol{z}_k)}.\end{aligned} \tag{7}$$

This estimation is iterated until a convergence criterion is reached, while monotonic increase in the likelihood is guaranteed. Finally, the covariance matrices $\boldsymbol{\Sigma}_i^{xx}$, $\boldsymbol{\Sigma}_i^{yx}$ and the means $\boldsymbol{\mu}_i^x$, $\boldsymbol{\mu}_i^y$ in (2) and (3) can be directly obtained from the estimated covariance matrices and means of $\boldsymbol{z}$, since

$$\boldsymbol{\Sigma}_i^{zz} = \begin{bmatrix}\boldsymbol{\Sigma}_i^{xx} & \boldsymbol{\Sigma}_i^{xy}\\ \boldsymbol{\Sigma}_i^{yx} & \boldsymbol{\Sigma}_i^{yy}\end{bmatrix} \quad \boldsymbol{\mu}_i^z = \begin{bmatrix}\boldsymbol{\mu}_i^x\\ \boldsymbol{\mu}_i^y\end{bmatrix}. \tag{8}$$

The GMM-based spectral conversion algorithm of (2) can be implemented with the covariance matrices having no structural restrictions or restricted to be diagonal, denoted as full and diagonal conversion respectively. Full conversion is of prohibitive complexity when combined with the adaptation algorithm for the nonparallel corpus conversion problem examined in Section III, thus here we concentrate on diagonal conversion. Note that the covariance matrix of $\boldsymbol{z}$ for the conversion method cannot be diagonal because this method is based on the cross-covariance of $\boldsymbol{x}$ and $\boldsymbol{y}$ which is found from (8). This will be zero if the covariance of $\boldsymbol{z}$ is diagonal. Thus, in order to obtain an efficient structure, we must restrict *each* of the matrices $\boldsymbol{\Sigma}_i^{xx}$, $\boldsymbol{\Sigma}_i^{yy}$, $\boldsymbol{\Sigma}_i^{xy}$, and $\boldsymbol{\Sigma}_i^{yx}$ in (8) to be diagonal. For achieving this restriction, the EM algorithm for full conversion must be modified accordingly, and the details can be found in [17].

III. ML Constrained Adaptation

The majority of spectral conversion methods that have been described so far in the literature, including the GMM-based methods, assume a parallel speech corpus for obtaining the spectral conversion parameters for every pair of reference and target speakers. Our objective here is to derive an algorithm that relaxes this constraint. In other words, we propose in this section an algorithm that derives the conversion parameters from a speech corpus in which the reference and target speakers do not necessarily utter the same words or sentences. In order to achieve this result, we apply the maximum-likelihood constrained adaptation method [22], [23], which offers the advantage of a simple probabilistic linear transformation leading to a mathematically tractable solution.

In addition to the pair of speakers for which we intend to derive the nonparallel training algorithm, we also assume that a parallel speech corpus is available for a *different* pair of speakers. From this latter corpus, we obtain a joint GMM model, derived as explained in Section II. In the following, the spectral vectors that correspond to the reference speaker of the parallel corpus are considered as realizations of random vector $\boldsymbol{x}$, while $\boldsymbol{y}$ corresponds to the target speaker of the parallel corpus. From the nonparallel corpus, we also obtain a sequence of spectral vectors, considered as realizations of random vector $\boldsymbol{x}'$ for the reference speaker and $\boldsymbol{y}'$ for the target speaker. We then attempt to relate the random variables $\boldsymbol{x}'$ and $\boldsymbol{x}$, as well as $\boldsymbol{y}'$ and $\boldsymbol{y}$, in order to derive a conversion function for the nonparallel corpus based on the parallel corpus parameters.

We assume that the target random vector $\boldsymbol{x}'$ is related to reference random vector $\boldsymbol{x}$ by a probabilistic linear transformation

$$\boldsymbol{x}' = \begin{cases} \mathbf{A}_1\boldsymbol{x} + \boldsymbol{b}_1 & \text{with probability } p(\lambda_1|\omega_i) \\ \mathbf{A}_2\boldsymbol{x} + \boldsymbol{b}_2 & \text{with probability } p(\lambda_2|\omega_i) \\ \vdots & \vdots \\ \mathbf{A}_N\boldsymbol{x} + \boldsymbol{b}_N & \text{with probability } p(\lambda_N|\omega_i). \end{cases} \tag{9}$$

This equation corresponds to the GMM constrained estimation that relates $\boldsymbol{x}'$ with $\boldsymbol{x}$ in the block diagram of Fig. 1. Each of the component transformations λ_j is related with a specific Gaussian ω_i of $\boldsymbol{x}$ with probability $p(\lambda_j|\omega_i)$ satisfying

$$\sum_{j=1}^{N} p(\lambda_j|\omega_i) = 1, \qquad i = 1, \ldots, M. \tag{10}$$

In the above equations M is the number of Gaussians of the GMM that corresponds to the joint vector sequence of the parallel corpus, $\mathbf{A}_j$ is a $K \times K$ matrix (K is the dimensionality of $\boldsymbol{x}$), and $\boldsymbol{b}_j$ is a vector of the same dimension with $\boldsymbol{x}$. Random vectors $\boldsymbol{y}'$ and $\boldsymbol{y}$ are related by another probabilistic linear transformation, similar to (9), as follows:

$$\boldsymbol{y}' = \begin{cases} \mathbf{C}_1\boldsymbol{y} + \boldsymbol{d}_1 & \text{with probability } p(\kappa_1|\omega_i) \\ \mathbf{C}_2\boldsymbol{y} + \boldsymbol{d}_2 & \text{with probability } p(\kappa_2|\omega_i) \\ \vdots & \vdots \\ \mathbf{C}_L\boldsymbol{y} + \boldsymbol{d}_L & \text{with probability } p(\kappa_L|\omega_i) \end{cases} \tag{11}$$

$$\sum_{\rho=1}^{L} p(\kappa_\rho|\omega_i) = 1, \qquad i = 1, \ldots, M. \tag{12}$$

Note that classes ω_i are the same for $\boldsymbol{x}$ and $\boldsymbol{y}$ by design in Section II.

All the unknown parameters (i.e., the matrices $\mathbf{A}_j$ and $\mathbf{C}_\rho$, and the vectors $\boldsymbol{b}_j$ and $\boldsymbol{d}_\rho$) can be estimated by use of the nonparallel corpus based on the GMM of the parallel corpus, by applying the EM algorithm. In essence, it is a linearly constrained maximum-likelihood estimation of the GMM parameters of $\boldsymbol{x}'$ and $\boldsymbol{y}'$. Concentrating on (9), it clearly follows that the pdf of $\boldsymbol{x}'$ given a particular class ω_i and λ_j will be

$$g(\boldsymbol{x}'|\omega_i, \lambda_j) = \mathcal{N}\left(\boldsymbol{x}'; \mathbf{A}_j\boldsymbol{\mu}_i^x + \boldsymbol{b}_j, \mathbf{A}_j\mathbf{\Sigma}_i^{xx}\mathbf{A}_j^T\right) \tag{13}$$

resulting in the pdf of $\boldsymbol{x}'$

$$g(\boldsymbol{x}') = \sum_{i=1}^{M}\sum_{j=1}^{N} p(\omega_i)p(\lambda_j|\omega_i)\mathcal{N} \cdot \left(\boldsymbol{x}'; \mathbf{A}_j\boldsymbol{\mu}_i^x + \boldsymbol{b}_j, \mathbf{A}_j\mathbf{\Sigma}_i^{xx}\mathbf{A}_j^T\right) \tag{14}$$

which is a GMM of $M \times N$ mixtures. In other words, the EM algorithm is applied in this case for estimating the matrices $\mathbf{A}_j$ and the vectors $\boldsymbol{b}_j$ in the same manner as described in the previous section, but now the means and covariance matrices of the pdf of $\boldsymbol{x}'$ are restricted to be linearly related to the GMM parameters of $\boldsymbol{x}$. For convenience, the formulas of the EM algorithm as applied to this problem in [23] are given here (it is interesting to compare these equations that follow with (4)–(7) in the previous section). The parameters that are estimated iteratively (an initial estimate of these parameters is needed and this is discussed in [23]) are the matrices $\mathbf{A}_j$, the vectors $\boldsymbol{b}_j$, and the conditional probabilities $p(\lambda_j|\omega_i)$. During the tth iteration, the E-Step involves computation of the following parameters:

$$n_{ij}^{(t)} = \sum_{k=1}^{n} p^{(t)}\left(\omega_i|\boldsymbol{x}_k'\right)p^{(t)}\left(\lambda_j|\boldsymbol{x}_k', \omega_i\right) \tag{15}$$

$$\boldsymbol{\mu}_{ij}^{(t)x'} = \frac{1}{n_{ij}^{(t)}}\sum_{k=1}^{n} p^{(t)}\left(\omega_i\boldsymbol{x}_k'\right)p^{(t)}\left(\lambda_j|\boldsymbol{x}_k', \omega_i\right)\boldsymbol{x}_k' \tag{16}$$

$$\mathbf{\Sigma}_{ij}^{(t)x'x'} = \frac{1}{n_{ij}^{(t)}}\sum_{k=1}^{n} p^{(t)}\left(\omega_i|\boldsymbol{x}_k'\right)p^{(t)}\left(\lambda_j|\boldsymbol{x}_k', \omega_i\right) \cdot \left(\boldsymbol{x}_k' - \boldsymbol{\mu}_{ij}^{(t)x'}\right)\left(\boldsymbol{x}_k' - \boldsymbol{\mu}_{ij}^{(t)x'}\right)^T \tag{17}$$

where

$$p^{(t)}(\omega_i|\boldsymbol{x}_k') = \frac{p(\omega_i)\sum_{j=1}^{N} p^{(t)}(\lambda_j|\omega_i)g^{(t)}\left(\boldsymbol{x}_k'|\omega_i, \lambda_j\right)}{\sum_{i=1}^{M}\sum_{j=1}^{N} p(\omega_i)p^{(t)}(\lambda_j|\omega_i)g^{(t)}\left(\boldsymbol{x}_k'|\omega_i, \lambda_j\right)} \tag{18}$$

$$p^{(t)}(\lambda_j|\boldsymbol{x}_k', \omega_i) = \frac{p^{(t)}(\lambda_j|\omega_i)g^{(t)}\left(\boldsymbol{x}_k'|\omega_i, \lambda_j\right)}{\sum_{j=1}^{N} p^{(t)}(\lambda_j|\omega_i)g^{(t)}\left(\boldsymbol{x}_k'|\omega_i, \lambda_j\right)} \tag{19}$$

and $g^{(t)}\left(\boldsymbol{x}'|\omega_i,\lambda_j\right)$, similarly with (13), is given from

$$g^{(t)}\left(\boldsymbol{x}'|\omega_i,\lambda_j\right)=\mathcal{N}\left(\boldsymbol{x}';\mathbf{A}_j^{(t)}\boldsymbol{\mu}_i^x+\boldsymbol{b}_j^{(t)},\mathbf{A}_j^{(t)}\boldsymbol{\Sigma}_i^{xx}\mathbf{A}_j^{(t)T}\right). \qquad (20)$$

Subsequently, the M-Step involves the computation of the needed parameters using (21)–(23), shown at the bottom of the page. The above equations are applied for $i=1,\ldots,M$ and $j=1,\ldots,N$, i.e., for all the different classes. The procedure is iterated until a convergence criterion is met, and again it holds that the likelihood is monotonically increased after each iteration. Note that (22) is greatly simplified when a diagonal GMM for $\boldsymbol{x}$ is assumed and when matrices $\mathbf{A}_j$ are assumed to be diagonal. This is the reason that the diagonal GMM conversion problem was especially examined in Section II. Thus, in the experiments that follow in this work, the covariance matrices for the conversion task, as well as the matrices $\mathbf{A}_j$ in (9) and $\mathbf{C}_\rho$ in (11) for the adaptation procedure, are restricted to be diagonal. More information on this issue can be found in [17], [22], and [23].

Matrices $\mathbf{C}_\rho$ and vectors $\boldsymbol{d}_\rho$ can be estimated in the same manner as above, and the pdf of $\boldsymbol{y}'$ will have a similar form with (14). It is now possible to derive the conversion function for the nonparallel training problem, based entirely on the parameters derived from a parallel corpus of a different pair of speakers. Based on the aforementioned assumptions, it holds that

$$\begin{aligned}E\left(\boldsymbol{y}'|\boldsymbol{x}'_k,\omega_i,\lambda_j,\kappa_\rho\right)&=\boldsymbol{\mu}_i^{y'}+\boldsymbol{\Sigma}_i^{y'x'}\boldsymbol{\Sigma}_i^{x'x'^{-1}}\left(\boldsymbol{x}'_k-\boldsymbol{\mu}_i^{x'}\right)\\&=\mathbf{C}_\rho\boldsymbol{\mu}_i^y+\boldsymbol{d}_\rho+\mathbf{C}_\rho\boldsymbol{\Sigma}_i^{yx}\boldsymbol{\Sigma}_i^{xx^{-1}}\mathbf{A}_j^{-1}\\&\quad\cdot\left(\boldsymbol{x}'_k-\mathbf{A}_j\boldsymbol{\mu}_i^x-\boldsymbol{b}_j\right) \qquad (24)\end{aligned}$$

since

$$\boldsymbol{\Sigma}_i^{y'x'}=\mathbf{C}_\rho\boldsymbol{\Sigma}_i^{yx}\mathbf{A}_j^T,\ \boldsymbol{\Sigma}_i^{x'x'}=\mathbf{A}_j\boldsymbol{\Sigma}_i^{xx}\mathbf{A}_j^T \qquad (25)$$

and

$$\boldsymbol{\mu}_i^{y'}=\mathbf{C}_\rho\boldsymbol{\mu}_i^y+\boldsymbol{d}_\rho,\ \boldsymbol{\mu}_i^{x\prime}=\mathbf{A}_j\boldsymbol{\mu}_i^x+\boldsymbol{b}_j. \qquad (26)$$

Finally, the conversion function for the nonparallel case becomes (see also [30])

$$\begin{aligned}\mathcal{F}\left(\boldsymbol{x}'_k\right)&=E\left(\boldsymbol{y}'|\boldsymbol{x}'_k\right) \qquad (27)\\&=\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{\rho=1}^{L}p\left(\omega_i|\boldsymbol{x}'_k\right)p\left(\lambda_j|\boldsymbol{x}'_k,\omega_i\right)p(\kappa_\rho|\omega_i)\\&\quad\cdot\Bigg[\mathbf{C}_\rho\boldsymbol{\mu}_i^y+\boldsymbol{d}_\rho+\mathbf{C}_\rho\boldsymbol{\Sigma}_i^{yx}\boldsymbol{\Sigma}_i^{xx^{-1}}\mathbf{A}_j^{-1}\\&\qquad\left(\boldsymbol{x}'_k-\mathbf{A}_j\boldsymbol{\mu}_i^x-\boldsymbol{b}_j\right)\Bigg]\end{aligned}$$

$$p\left(\omega_i|\boldsymbol{x}'_k\right)=\frac{p(\omega_i)\sum\limits_{j=1}^{N}p(\lambda_j|\omega_i)g\left(\boldsymbol{x}'_k|\omega_i,\lambda_j\right)}{\sum\limits_{i=1}^{M}\sum\limits_{j=1}^{N}p(\omega_i)p(\lambda_j|\omega_i)g(\boldsymbol{x}'_k|\omega_i,\lambda_j)} \qquad (28)$$

$$p(\lambda_j|\boldsymbol{x}'_k,\omega_i)=\frac{p(\lambda_j|\omega_i)g(\boldsymbol{x}'_k|\omega_i,\lambda_j)}{\sum\limits_{j=1}^{N}p(\lambda_j|\omega_i)g(\boldsymbol{x}'_k|\omega_i,\lambda_j)} \qquad (29)$$

and $g\left(\boldsymbol{x}'|\omega_i,\lambda_j\right)$ is given from (13).

IV. Results and Discussion

The spectral conversion method for the case of a nonparallel training corpus that was derived in the previous section is evaluated in this section both objectively and subjectively. Two different objective measures are employed for measuring the performance of the proposed algorithm, the mean-squared error (MSE), as well as the results obtained from a speaker identification system we implemented [11]. The latter is especially important for testing our algorithm, since it is expected to give us a better measure of the significance of the adaptation step of the algorithm, as opposed to the results obtained when no adaptation occurs (i.e., when a conversion function derived for a specific pair of source/target speakers is applied to a different pair). Both the MSE and the speaker identification results will give us better insight as to the algorithm's performance when compared to the parallel conversion algorithm which corresponds to the ideal case (when a parallel corpus is available). Thus, successful performance of the proposed algorithm will be indicated by objective results that are comparable to those obtained for the parallel case algorithm. Listening tests are essential for judging the performance of voice conversion algorithms. In Section IV-C we show that the subjective tests also indicate successful performance of the proposed algorithm, and comparable performance to the parallel case.

$$p^{(t+1)}(\lambda_j|\omega_i)=\frac{n_{ij}^{(t)}}{\sum\limits_{j=1}^{N}n_{ij}^{(t)}} \qquad (21)$$

$$\sum_{i=1}^{M}n_{ij}^{(t)}\left\{\mathbf{A}_j^{(t+1)}-\Sigma_i^{xx^{-1}}\left[\mathbf{A}_j^{(t+1)^{-1}}\left(\boldsymbol{\mu}_{ij}^{(t)x'}-\boldsymbol{b}_j^{(t+1)}\right)-\mu_i^x\right]\left(\boldsymbol{\mu}_{ij}^{(t)x'}-\boldsymbol{b}_j^{(t+1)}\right)^T-\Sigma_i^{xx^{-1}}\mathbf{A}_j^{(t+1)^{-1}}\boldsymbol{\Sigma}_{ij}^{(t)x'x'}\right\}=0 \qquad (22)$$

$$\boldsymbol{b}_j^{(t+1)}=\left[\sum_{i=1}^{M}n_{ij}\mathbf{A}_j^{(t+1)^{-T}}\Sigma_i^{xx^{-1}}\mathbf{A}_j^{(t+1)^{-1}}\right]^{-1}\left[\sum_{i=1}^{M}n_{ij}\mathbf{A}_j^{(t+1)^{-T}}\Sigma_i^{xx^{-1}}\mathbf{A}_j^{(t+1)^{-1}}\left(\boldsymbol{\mu}_{ij}^{(t)x'}-\mathbf{A}_j^{(t+1)}\mu_i^x\right)\right] \qquad (23)$$

As mentioned previously, the spectral vectors used here are the LSFs (22nd order) due to their favorable interpolation properties. The corpus used is the VOICES corpus, available from OGI's CSLU [31].[1] This is a parallel corpus and is used for both the parallel and nonparallel training cases that are examined in this section, in a manner explained in the next paragraphs. The sampling rate of this corpus is 22 050 Hz which is retained in our experiments. It is interesting to mention that this corpus was recorded using a "mimicking" approach. This means that during the corpus recording, all the speakers were asked to follow the timing, stress, and intonation patterns of a template speaker. The reason for using this approach was so that there is a high degree of natural time-alignment in the recorded speech of all the different speakers, which is very important for minimizing the signal processing needed during the training and testing tasks of the parallel conversion algorithms. In other words, the natural time-alignment of the recorded speech contributes toward a more successful performance of the time-alignment algorithm needed for conversion. Otherwise, this time-alignment algorithm might produce errors which would affect the final performance of the conversion task. For our nonparallel conversion algorithm this mimicking approach is very helpful as well, since the nonparallel training is based on the previously derived parameters from a parallel corpus, as explained earlier. Because of this dependence, it is expected that the performance of our algorithm is positively affected by this "mimicking" approach in the design of the corpus. This dependence, though, is not direct (since our algorithm includes the intermediate adaptation step), consequently further research is needed to evaluate the effect of the corpus design in the final algorithm performance.

A. MSE Results

The error measure used in this section is the mean-squared error normalized by the initial distance between the reference and target speakers, i.e.,

$$\mathcal{E} = \frac{\frac{1}{n}\sum_{k=1}^{n} \|\boldsymbol{y}_k - \mathcal{F}(\boldsymbol{x}_k)\|^2}{\frac{1}{n}\sum_{k=1}^{n} \|\boldsymbol{y}_k - \boldsymbol{x}_k\|^2} \tag{30}$$

where $\boldsymbol{x}_k$ is the reference vector at instant k, $\boldsymbol{y}_k$ is the target vector at instant k, and $\mathcal{F}(\cdot)$ denotes the conversion function used, which can be the one of (2) or (27) depending whether training is performed in a parallel or nonparallel manner. For all results given in this section, the number of GMM classes for the parameters obtained from the parallel corpus is 16. The number of vectors for the parallel and the nonparallel training corpus for a 30-ms window is about 19 000 (denoted here as full corpus), which corresponds to 40 out of the 50 sentences available in the corpus. The results given in this section are the averages of the remaining 10 sentences.

The results described in this section can be found in Tables I and II. These two tables contain the same type of results as explained in the following Item 1, and are different only regarding the training data used, as explained in the following Item 2.

TABLE I
NORMALIZED ERROR FOR FOUR DIFFERENT PAIRS OF PARAMETERS DERIVED FROM A PARALLEL CORPUS, WHEN APPLIED TO TWO DIFFERENT SPEAKER PAIRS OF A NONPARALLEL CORPUS (**DIFFERENT** SENTENCES IN PARALLEL AND NONPARALLEL TRAINING)

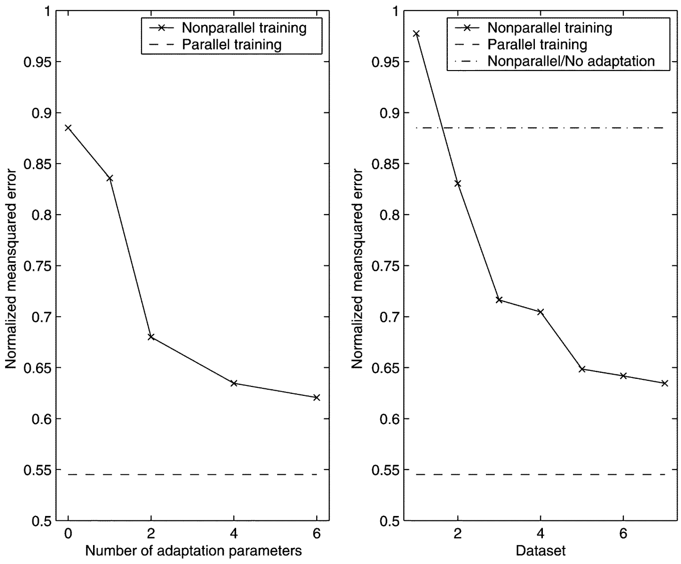
Conversion Method	Normalized Error			
	Test1 (M-F)		Test2 (M-M)	
	None	Adapt.	None	Adapt.
Case 1 (M-M)	0.7879	0.6137	0.7897	0.7977
Case 2 (M-M)	0.9340	0.8644	0.8314	0.7330
Case 3 (M-F)	0.8882	0.6809	1.0264	0.6980
Case 4 (M-F)	0.7307	0.6761	0.8342	0.7073
Parallel	0.5221		0.5453	

TABLE II
NORMALIZED ERROR FOR FOUR DIFFERENT PAIRS OF PARAMETERS DERIVED FROM A PARALLEL CORPUS, WHEN APPLIED TO TWO DIFFERENT SPEAKER PAIRS OF A NONPARALLEL CORPUS (**SAME** SENTENCES IN PARALLEL AND NONPARALLEL TRAINING)

Conversion Method	Normalized Error			
	Test1 (M-F)		Test2 (M-M)	
	None	Adapt.	None	Adapt.
Case 1 (M-M)	0.7957	0.6148	0.7749	0.7154
Case 2 (M-M)	0.8749	0.7510	0.8147	0.7017
Case 3 (M-F)	0.8512	0.6368	1.0371	0.7462
Case 4 (M-F)	0.7252	0.6169	0.8850	0.6346
Parallel	0.5221		0.5453	

1) Both Tables I and II give the normalized mean-squared error for two different pairs of nonparallel reference and target speakers (Test 1 and Test 2 in the tables) for four different adaptation cases (i.e., four different pairs of speakers in parallel training, Cases 1–4). Test 1 corresponds to male-to-female (M-F) conversion, while Test 2 corresponds to male-to-male (M-M) conversion. Similarly, Cases 1–2 correspond to male-to-male conversion while Cases 3–4 correspond to male-to-female. The column denoted as "None" in each of these tables corresponds to no adaptation, i.e., when the derived parameters from the parallel corpus are directly applied to the speaker pair from the nonparallel corpus, while the column "Adapt." corresponds to the conversion function of (27), for four adaptation parameters for both the reference and the target speaker [$L = N = 4$ in (27)]. The last row of each table gives the error when the conversion parameters are derived by parallel training (i.e., the ideal case).
2) These two tables correspond to two different choices of the training corpus. For Table I the corpus for the parallel pair (speakers A and B in Fig. 1) is chosen to be sentences 1–10 of the full corpus, while for adaptation, sentences 11–25 for relating speaker C with speaker A and sentences 26–40 for relating speaker D with speaker B. This means that all sentences are different for the different tasks. For the second choice of corpus (Table II), the full training corpus is used for all tasks. Inevitably for this latter case, the sentences in parallel and nonparallel training will be the same. In parallel training, the fact that the same sentences are used is essential since the reference and target vectors are aligned, and this vector-to-vector correspondence is required during training. In contrast, for nonparallel training the corpus is used as explained here for adaptation of the spectral conversion parameters, thus the fact that the corpus was created in a parallel manner is not exploited and is not expected to influence the results. The

[1] See also http://www.cslu.ogi.edu/corpora/voices.



(a) (b)

Fig. 2. Normalized error (a) when using different number of adaptation parameters (0 corresponds to no adaptation) and (b) for various choices of training dataset (see Table III). The dashed line corresponds to the error when a parallel corpus is used for training. The dashed–dotted line corresponds to no adaptation.

results in Table I, derived with different sentences as explained, are included in order to further support this argument. In total, 10 out of the 12 speakers of the corpus were used in order to test the performance of the algorithm with a variety of speaker pairs.

It is apparent from Tables I and II that the adaptation methods proposed result in a large error decrease compared to simply applying the conversion parameters of a given pair to a different pair of speakers. This improvement can reach the level of 30% when the initial distance is large, which is exactly what is desired. This is true *both* when the sentences are different or the same (Table I versus Table II) and this supports our previous argument. The performance for the latter case is on the average better compared to the former, due to the fact that when the full corpus is used for adaptation, more vectors are available and adaptation is more accurate (40 versus 15 sentences). The fact that more data will produce better results for the same number of estimated parameters is intuitive and has been shown for the parallel conversion algorithms (e.g., in [9]). This is also shown later in this section, when the results in Fig. 2(b) are discussed. The performance that we obtain when the conversion parameters are derived by parallel training is always better, compared with nonparallel training (although in most cases the two are comparable). This is an expected and intuitive result since in parallel training we exploit a particular advantage of the speech corpus which is not available in a nonparallel corpus. The methods proposed here intend to address the lack of a parallel corpus and are suitable only for this case. It is also of interest to note that the use of conversion parameters derived from a pair in the parallel corpus that is of same gender to the one in the nonparallel corpus (e.g., derived parameters from a male-to-male pair applied to a male-to-male pair) does not seem to perform better than when the genders are not the same. The error does not seem to display any particular patterns when no adaptation is performed, but it is interesting that in most cases we examined the initial distance is decreased (i.e., error less than one). The results obtained by the speaker identification system are of particular interest in this case, as the discussion that follows in Section IV-B clearly demonstrates.

TABLE III
NUMBER OF VECTORS (THOUSANDS) IN NONPARALLEL TRAINING FOR THE DATASETS IN FIG. 2(b)

Dataset	1	2	3	4	5	6	7
kVectors	0.25	0.5	1	2.5	5	10	19

In Fig. 2(a), the performance of the algorithm for a different number of adaptation parameters is shown, using the full corpus both for parallel (dashed line in the figure) and nonparallel (solid line in the figure) training. As mentioned previously, by the term adaptation parameters we refer to the values that correspond to L and N in (27). The number of adaptation parameters that is given is the same for the adaptation of the reference speaker and that of the target speaker, although a different number can be used for each case. Adaptation of zero parameters in this figure corresponds to the case when no adaptation of the parameters is performed. From this figure it is evident that, as expected, there is a significant error decrease when increasing the number of adaptation parameters, since this corresponds to a more accurate modeling of the statistics of the spectral vectors. On the other hand, when increasing the number of adaptation parameters above 4, the error remains approximately constant, concluding that this number of parameters is sufficient to model the statistics of the spectral vectors and further increase does not offer any advantage. In fact, further increase of the adaptation parameters might result in an error increase, which is something that we often noticed, and can be attributed to the effect of overtraining. This is an issue that is evident in Fig. 2(b), and consequently is discussed in the next paragraph.

In Fig. 2(b), the performance of the algorithm is given for different sizes of the nonparallel corpus, using the full corpus for parallel training, and four adaptation parameters for both the reference and target speaker. The dataset numbers in the figure correspond to the numbers of vectors given in Table III. The error when no adaptation is used (dashed–dotted line), as well as when the corpus is used in a parallel manner (dashed line), is also shown. From this figure we can see that there is a significant error decrease when the size of the corpus is increased. As is the case for the parallel corpus [9], the error decrease is less significant when the size of the corpus increases above 5 000–10 000 vectors. In Fig. 2(b), we can also notice the effect of overtraining, when we compare the performance of dataset 1 with the error obtained when simply applying the conversion parameters of one pair to a different pair (the dashed–dotted line in the figure corresponding to the Nonparallel/No adaptation case). From the figure we can see that the obtained error is less for the Nonparallel/No adaptation case than for the dataset 1 case. This might seem as a counterintuitive fact, since in the latter case more information has been used for obtaining the conversion parameters than in the former case (i.e., adaptation performs worse than no adaptation). This can be attributed to overtraining, which occurs when there are too many parameters in the model to be estimated from a comparatively small number of data. In

TABLE IV
NORMALIZED ERROR FOR THE HPT APPROACH (USING ONLY THE TRANSFORMATION WITH THE HIGHEST PROBABILITY), FOR FOUR DIFFERENT PAIRS OF PARAMETERS DERIVED FROM A PARALLEL CORPUS, WHEN APPLIED TO 2 DIFFERENT SPEAKER PAIRS OF A NONPARALLEL CORPUS (**SAME** SENTENCES IN PARALLEL AND NONPARALLEL TRAINING AS IN TABLE II)

Conversion Method	Normalized Error			
	Test1 (M-F)		Test2 (M-M)	
	None	Adapt.	None	Adapt.
Case 1 (M-M)	0.7957	0.6956	0.7749	0.7440
Case 2 (M-M)	0.8749	0.7995	0.8147	0.7529
Case 3 (M-F)	0.8512	0.6960	1.0371	0.8089
Case 4 (M-F)	0.7252	0.6766	0.8850	0.6506
Parallel	0.5221		0.5453	

such cases, the derived parameters do not perform well when applied to different data than those used during testing, i.e., they cannot be successfully generalized. In this case it is evident that the 250 vectors of dataset 1 are not enough for successfully estimating the four adaptation parameters (resulting in $4 \times 16 = 64$ linearly constrained GMM classes for both the source and the target speakers).

The nonparallel conversion method that has been proposed here is computationally demanding during training and during the actual conversion. The training procedure can be simplified if a small number of transformation parameters are used, but there is a tradeoff regarding the number of transformation parameters and the resulting mean-squared error. This has been shown when discussing the results of Fig. 2(a). Similarly, the conversion phase is computationally expensive since it includes the calculation and summation of $M \times N \times L$ linear terms in (27). In [23] a similar issue arises, and is addressed there by a method referred to as HPT, which reduces the total number of linear terms that are actually used. Following this approach for our conversion method, we constrain the probabilities $p(\lambda_j|\omega_i)$ and $p(\kappa_\rho|\omega_i)$, so that for given class ω_i

$$p(\lambda_j|\omega_i) = \begin{cases} 1, & \text{for } \lambda_j \text{ with the highest probability} \\ 0, & \text{elsewhere} \end{cases} \tag{31}$$

and similarly for $p(\kappa_\rho|\omega_i)$. In essence, this means that for each class ω_i we use only one of the available transformation components λ_j (corresponding to one matrix $\mathbf{A}_j$ and vector $\boldsymbol{b}_j$), and one of the transformation components κ_ρ (corresponding to one matrix $\mathbf{C}_\rho$ and one vector $\boldsymbol{d}_\rho$). This selection is based on the transformation probabilities $p(\lambda_j|\omega_i)$ and $p(\kappa_\rho|\omega_i)$, as implied by (31). For our conversion method, this constraint results in using only M terms in (27), which is the same number of terms required for parallel conversion as well. In Table IV we present some results for the HPT method as applied to our algorithm. The results shown there correspond to the same training and testing conditions as those in Table II, and for the convenience of the reader the results regarding the no adaptation case (column denoted as "None") have been included in this table as well. From these results we can see that the HPT method reduces the initial error (i.e., with no adaptation) significantly in most cases. On the other hand, by comparing Table IV with Table II, we can see that there is a performance tradeoff when comparing HPT to the unconstrained case. In other words, the HPT method, which reduces the complexity of the algorithm during conversion, also produces higher mean-squared error when compared to the—computationally more demanding—unconstrained case. From the results shown here for HPT, though, we can conclude that this method is a viable alternative for cases when complexity is of central importance.

B. Speaker Identification Results

In this section a speaker identification error measure is employed. Since voice conversion algorithms have the objective to modify the source speaker's identity into that of the target speaker, a speaker identification system is ideal for testing the conversion performance. We implemented the speaker identification system of [11], which is a simple but powerful system that has been shown to successfully perform this task. This is a GMM-based system, where for each one of the speakers in the database, a corpus is used to train a GMM model of the extracted sequences of (short-time) spectral envelopes. Thus, for a predefined set of speakers a sufficient amount of training data is assumed to be available, and identification is performed based on segmental-level information only. During the identification stage, the spectral vectors of the examined speech waveform are extracted and classified to one of the speakers in the database, according to a maximum *a posteriori* criterion. More specifically, a group of S speakers in the training dataset is represented by S different GMMs $\lambda_1, \lambda_2, \ldots, \lambda_S$,[2] a sequence (or segment) of n consecutive spectral vectors $X = [\boldsymbol{x}_1 \boldsymbol{x}_2 \ldots \boldsymbol{x}_n]$ is identified as spoken by speaker $\hat{S}$ based on

$$\hat{S} = \arg \max_{1 \le q \le S} p(\lambda_q|X) = \arg \max_{1 \le q \le S} \frac{p(X|\lambda_q)p(\lambda_q)}{p(X)}. \tag{32}$$

For equally likely speakers and since $p(X)$ is the same for all speaker models the above equation becomes

$$\hat{S} = \arg \max_{1 \le q \le S} p(X|\lambda_q) \tag{33}$$

and finally, for independent observations and using logarithms, the identification criterion becomes

$$\hat{S} = \arg \max_{1 \le q \le S} \sum_{k=1}^{n} \log p(\boldsymbol{x}_k|\lambda_q) \tag{34}$$

where

$$p(\boldsymbol{x}_k|\lambda_q) = \sum_{i=1}^{M} p_q(\omega_i) \mathcal{N}\left(\boldsymbol{x}_k; \boldsymbol{\mu}_{i,q}^{x}, \boldsymbol{\Sigma}_{i,q}^{xx}\right). \tag{35}$$

Note that this is a text-independent system, i.e., the sentences during the validation stage need not be the same as the ones used for training. We are not only interested in the final decision of the classification system, but also in a measure of "certainty" for that decision. As in [11], the error measure employed is the percentage of segments of the speech recording that were identified as spoken by the most likely speaker. As previously explained, a segment in this case is defined as a time-interval of prespecified duration containing n spectral vectors, during which these

[2] In this section λ_q denotes a particular GMM $\lambda_q = \{p_q(\omega_i), \boldsymbol{\mu}_{i,q}^{x}, \boldsymbol{\Sigma}_{i,q}^{xx}\}$, not to be confused with λ_j in (9) where it denotes a specific *class* of a particular GMM.

vectors are collectively classified based on (34), to one of the speakers by the identification system. If each segment contains n vectors (n depending on the prespecified duration of each segment), different segments overlap as shown below, where Segment #1 and Segment #2 are depicted

$$\overbrace{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n}^{\text{Segment \#1}}, \boldsymbol{x}_{n+1}, \boldsymbol{x}_{n+2}, \ldots$$
$$\boldsymbol{x}_1, \overbrace{\boldsymbol{x}_2, \ldots, \boldsymbol{x}_n, \boldsymbol{x}_{n+1}}^{\text{Segment \#2}}, \boldsymbol{x}_{n+2}, \ldots.$$

The resulting percentages are an intuitive measure of the performance of the system. There is a performance decrease when decreasing the segment duration, which is an expected result since the more data are available, the better the performance of the system. A large number of segments is also important for obtaining more accurate results; it should be noted, though, that an identification decision is taken for each different segment, independently of the other segments. In [11], a segment duration of 5 s was found to be a minimal value for accurate identification, and this is the value used in our system as well. We trained a diagonal GMM of 16 classes for each of the 12 speakers available in the OGI corpus. Note that a speaker identification measure for a voice conversion system was also employed in [7]. Here, the performance measure used is more insightful as compared to the likelihood measure in [7]. Additionally, the availability of 12 different speakers used for the identification task offers more reliable results than using only two speakers as in [7] (source and target speakers only).

Sentences 1–20 of the corpus were used for training the identification system, while the remaining sentences 21–50 were used for obtaining the identification results in the following manner. Identification results are obtained for the following waveforms.

1) Original speech by a particular speaker, available from the corpus.
2) Converted speech by the parallel conversion algorithm corresponding to (2), where the target speaker is the same as in Item 1).
3) Converted speech by the proposed nonparallel conversion algorithm corresponding to (27), where the target speaker is the same as in Item 1).

TABLE V

Normalized Error for the Sentences Used in the Speaker Identification Experiments, for Four Different Pairs of Parameters Derived From a Parallel Corpus, When Applied to Two Different Speaker Pairs of a Nonparallel Corpus (Testing Sentences 21–50, Same Training Data as in Table II)

Conversion Method	Normalized Error			
	Test1 (M-F)		Test2 (M-M)	
	None	Adapt.	None	Adapt.
Case 1 (M-M)	0.8316	0.6255	0.7984	0.7303
Case 2 (M-M)	0.9107	0.7510	0.8583	0.7306
Case 3 (M-F)	0.8981	0.6524	1.0762	0.7681
Case 4 (M-F)	0.7521	0.6275	0.8844	0.6567
Parallel	0.5138		0.5478	

TABLE VI

Speaker Identification Results (Speaker Identified and Percentage of Segments Identified as Spoken by This Speaker in Parentheses) for Four Different Pairs of Parameters Derived From a Parallel Corpus, When Applied to Two Different Speaker Pairs of a Nonparallel Corpus (Same Test and Training Data as in Table V)

Conversion Method	Speaker Identication Results			
	Test1 (M-F)		Test2 (M-M)	
	None	Adapt.	None	Adapt.
Case 1 (M-M)	B(98.94)	D(95.74)	B(100)	D(74.16)
Case 2 (M-M)	B(95.74)	D(85.11)	B(95.51)	D(75.28)
Case 3 (M-F)	B(81.91)	D(87.23)	B(87.64)	D(43.33)
Case 4 (M-F)	B(93.62)	D(88.30)	B(98.88)	D(80.90)
Parallel	D(97.54)		D(96.63)	
Original	D(100)		D(99.78)	

Thus, our objective is to obtain identification results for a particular speaker's original speech, compared to the synthesized (converted) speech from some other (source) speaker, both using the algorithm of (2) and our algorithm (27). As explained, a large number of sentences is important for more accurate results, and this is the reason that sentences 21–50 of the corpus were used for testing, rather than sentences 41–50 as in Section IV-A. As a result, some of the sentences (more specifically 21–40) are used both in training and testing. However, as is evident in [9], this issue does not influence the obtained results if a large number of vectors is available for training (as is the case here). In Table V, the MSE results for sentences 21–50 are presented. Note that the same training data used for obtaining the conversion results of Table II were also used for the results of Table V. Thus, the only difference between the results in these two tables is that sentences 41–50 were used for testing in Table II, while sentences 21–50 were used for testing in Table V. It is evident that the results in Tables II and V are very similar. In other words, we verify the fact that, although for the results in the latter table some sentences are used both during training and testing, this does not have any significant consequences in the obtained results. In Table VI, the identification results (percentage of segments identified as spoken by the most likely—from (34)—speaker) are given, corresponding to the MSE results of Table V for exactly the same sentences. In Table VI, the row denoted as "Original" corresponds to the identification results for the original recorded speech by the corresponding target speaker. In this table, the most likely speaker identified by the system is displayed, based on the notation of Fig. 1, while in the following parentheses the percentage of identification for this speaker is given. In Fig. 1 Speaker A and Speaker B represent the source and target speakers in the parallel corpus used to derive the initial conversion parameters, while Speaker C and Speaker D represent the source and target speakers in the nonparallel conversion task. We remind the reader that for each of the Cases 1–4 in the tables, a different pair of speakers is used from the corpus (four pairs in total). This means that with our notation, Speaker A and Speaker B correspond to a different speaker pair for each of the four different cases. Similarly, Test 1 and Test 2 in the tables correspond to two different pairs of speakers from the corpus, thus Speaker C and Speaker D in Table VI correspond to two different pairs for these two cases. The row in the tables that is denoted as "Parallel" corresponds to the ideal case when a parallel corpus is available for the same pair of speakers that the nonparallel conversion was applied, i.e., source speaker C and target speaker D using the notation of Fig. 1. With the aforementioned notation, identification of Speaker D in that table means that conversion is performed successfully. In Table VI we see that Speaker D is identified correctly in all cases, except when no adaptation is applied. In the latter case, there is a consistent identification of Speaker B, who is the target speaker in the first step of our algorithm, before adaptation is performed.

Based on the results of these tables, the following conclusions can be derived.

- Results for parallel conversion are very close to those for the natural recorded waveforms. In other words, for source speaker C and target speaker D, speaker D is identified with almost the same percentage as the natural recorded speech of speaker D.
- Waveforms from the nonparallel conversion system are also correctly identified, but with somewhat higher error when compared to parallel conversion. This is expected, since the MSE results also showed a somewhat higher error for the nonparallel case when compared to the parallel case, as discussed in Section IV-A. As explained there, the parallel conversion algorithm is expected to perform better than our nonparallel algorithm, since the parallel corpus has an additional property when compared to the nonparallel corpus, and this property is directly exploited during training. The focus here is on the fact that the nonparallel procedure that is proposed can produce successful results, that are comparable to those obtained when using a parallel conversion algorithm.
- The identification results for the nonparallel conversion with no adaptation are very revealing of the importance of adaptation. Referring to Fig. 1, if the conversion function derived for source speaker A and target speaker B is applied to source speaker C, the resulting waveform is identified as spoken by speaker B. In other words, a conversion function derived in a parallel manner results in identification of the target speaker used to train the conversion system, regardless if the source speaker is different than the one in training.

The last observation, that a high percentage of identification is obtained for parallel conversion even in the case when the source speaker is different than the one used for training the system, is an unexpected result and can be possibly attributed to the forced-choice nature of the algorithm. This is indicative of the importance of using both the MSE and identification measures when evaluating voice conversion algorithms. An additional note is the fact that we proposed a context-independent error measure, which does not guarantee that the phoneme sequence in the speech is retained. In turn, this means that the speech produced by the conversion algorithm might be completely different than the desired (e.g., as a result of errors in the phoneme mapping), but the measure would still indicate the conversion results as successful. In this sense, a context-dependent speaker identification system might be a better measure of performance. In our case, the fact that the phoneme sequence is retained is indicated by the MSE measure, and this is an additional reason why the context-independent measure proposed here should be used only when combined with the MSE measure.

C. Listening Tests Results

Subjective tests are essential for judging the performance of voice conversion algorithms, since the target users of such technologies will be human listeners. We conducted listening tests for both our nonparallel algorithm of (27), as well as for the parallel case algorithm of (2). In this manner, we not only measure the performance of the proposed algorithm, but we also compare its performance with the parallel case in exactly the same conditions (synthesized speech, listeners in the test, etc.). For these listening tests we synthesized three different sentences from the same source speaker and using the same target speaker, using the *VOICES* corpus. The test employed is the ABX test that has been mostly followed in voice conversion literature. In ABX tests, A and B are speech waveforms corresponding to the source and target speakers (in random order throughout the tests), while X is the corresponding waveform that has been synthesized with the voice conversion algorithm. We designed three ABX tests, one for each chosen sentence of the corpus. The same test is employed for both the parallel and nonparallel conversion algorithms presented here (total of six tests, three for the parallel and three for the nonparallel case). A total of 14 subjects participated in the tests.

One important difference that distinguishes the tests conducted for this work from other ABX tests in the literature is the choice of the target speech. In the majority of ABX tests for voice conversion in the literature, A and B correspond to speech from two different speakers in the corpus. One implication of this choice is that the final synthesized speech that is judged, includes spectral conversion as well as time-scale and pitch-scale modifications. We believe that since the central importance in the majority of voice conversion algorithms is on spectral conversion, it is important to derive a test that measures the performance of spectral conversion alone. For this purpose, we propose *synthesizing* the target speech as follows. Assuming the source speaker is a male speaker from the corpus, a female speaker is chosen from the corpus for synthesizing the target speaker. The target speech is synthesized by applying the sequence of spectral vectors obtained from the female speaker, to the corresponding residual signal from the source (male) speaker. In other words, the result is the “perfect” spectral conversion of the male speaker into the female speaker (i.e., corresponding to zero mean-squared error). In this way, the pitch and time characteristics are exactly the same in both the source and target speech, and the only difference lies in the spectral envelopes, which is the objective of spectral conversion to match. The reason for obtaining the source speech from a male speaker and the target speech from a female speaker is to create two distinct speakers for the task, given the fact that both the source and target speech will not differ regarding the time- and pitch-scale characteristics. Otherwise, it might be very difficult for the listeners to distinguish between the source and target speech, which in turn would produce incorrect results for the listening test. In fact, our ABX tests were preceded by a brief section of speaker identification, where all the listeners correctly identified the source and target speaker without difficulty.

The results of the ABX tests are given in Table VII. From this table we can conclude that the proposed method for nonparallel conversion produces satisfying results and can be con-

TABLE VII
RESULTS FROM THE ABX LISTENING TESTS, FOR THE ALGORITHM PROPOSED HERE (NONPARALLEL CASE) AS WELL AS THE IDEAL CASE WHEN A PARALLEL TRAINING CORPUS IS AVAILABLE (PARALLEL CASE)

	Non-parallel Case	Parallel Case
Results correct	74%	88%

sidered successful. It is also apparent that the parallel conversion method produces more convincing results than the nonparallel method, and this was also evident in the objective results of this section. As mentioned previously, this is an expected result given that the parallel conversion method directly exploits an additional property of the available corpus. The proposed method for nonparallel training attempts to address the lack of a parallel corpus, and is only meaningful in that scenario. Finally, it is of interest to note that the results for the parallel case are similar to other ABX tests for voice conversion that can be found in the literature (e.g., [1] and [9]).

Some examples of the proposed algorithm have been made available for the interested reader at http://www.seas.upenn.edu/-mouchtar/vc-demo/. There, three different directories can be found that correspond to three different examples. Each directory contains four speech waveforms. The one denoted as "src" corresponds to the source speech recording (natural recording of a speaker from the corpus). The recording denoted as "trg" is the target speech, which is synthesized as explained when describing the design of the listening test, earlier in this section. The recording denoted as "par" is the resulting waveform when using the parallel conversion method of [9], while the recording denoted as "adp" corresponds to the result obtained when using the nonparallel conversion method proposed here.

V. CONCLUSION

Current voice conversion algorithms require a parallel speech corpus that contains the same utterances from the source and target speakers for deriving a conversion function. Here, we proposed an algorithm that relaxes this constraint and allows for the corpus to be nonparallel. Our results clearly demonstrate that the proposed method performs quite favorably and the conversion error is low and comparable with the error obtained with parallel training. It was shown that adaptation can reduce the initial mean-squared error, obtained by simply applying the conversion parameters developed for a specific pair of speakers to a different pair, by a factor that can reach 30%. The speaker identification results were also useful for this case, since they showed that adaptation is essential so that the desired target speaker is identified. The successful performance of our algorithm was also indicated by formal listening tests.

If the nonparallel corpus is large enough so that it contains a sufficient number of occurrences of all phonemes, the performance improvement will be large. On the other hand, the recording conditions for the two different corpora can influence algorithm performance. In the case examined here, the speech recordings were made in the same environment and using the same quality microphones. If, however, the parallel corpus is made in different conditions compared to the nonparallel corpus, then it is possible that the adaptation algorithm described here might not result in significant improvement, due to reasons such as microphone quality, reverberation, etc.

ACKNOWLEDGMENT

The authors would like to thank all the volunteers who participated in the listening tests.

REFERENCES

[1] Y. Stylianou, O. Cappe, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. Speech Audio Process.*, vol. 6, no. 2, pp. 131–142, Mar. 1998.

[2] A. Mouchtaris, J. Van der Spiegel, and P. Mueller, "A spectral conversion approach to the iterative Wiener filter for speech enhancement," presented at the *IEEE Int. Conf. Multimedia and Expo (ICME)*, 2004.

[3] S. Furui, "Research on individuality features in speech waves and automatic speaker recognition techniques," *Speech Commun.*, vol. 5, pp. 183–197, 1986.

[4] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, New York, Apr. 1988, pp. 655–658.

[5] H. Kuwabara and Y. Sagisaka, "Acoustic characteristics of speaker individuality: Control and conversion," *Speech Commun.*, vol. 16, no. 2, pp. 165–173, 1995.

[6] L. M. Arslan and D. Talkin, "Speaker transformation using sentence HMM based alignments and detailed prosody modification," in *Proc. IEEE Int. Conf Acoustics, Speech and Signal Processing (ICASSP)*, Seattle, WA, May 1998, pp. 289–292.

[7] L. M. Arslan, "Speaker transformation algorithm using segmental codebooks (STASC)," *Speech Commun.*, vol. 28, pp. 211–226, 1999.

[8] M. Narendranath, H. A. Murthy, S. Rajendran, and B. Yegnanarayana, "Transformation of formants for voice conversion using artificial neural networks," *Speech Commun.*, vol. 16, no. 2, pp. 207–216, 1995.

[9] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Seattle, WA, May 1998, pp. 285–289.

[10] G. Baudoin and Y. Stylianou, "On the transformation of the speech spectrum for voice conversion," in *IEEE Proc. Int. Conf. Spoken Language Processing (ICSLP)*, Philadephia, PA, Oct. 1996, pp. 1405–1408.

[11] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 1, pp. 72–83, Jan. 1995.

[12] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[13] D. G. Childers, "Glottal source modeling for voice conversion," *Speech Commun.*, vol. 16, no. 2, pp. 127–138, 1995.

[14] A. Kain and M. W. Macon, "Design and evaluation of a voice conversion algorithm based on spectral envelope mapping and residual prediction," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Salt Lake City, UT, May 2001, pp. 813–816.

[15] A. Kumar and A. Verma, "Using phone and diphone based acoustic models for voice conversion: A step toward creating voice fonts," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Hong Kong, Apr. 2003, pp. 720–723.

[16] A. Mouchtaris, S. S. Narayanan, and C. Kyriakakis, "Multichannel audio synthesis by subband-based spectral conversion and parameter adaptation," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 2, pp. 263–274, Mar. 2005.

[17] ——, "Maximum likelihood constrained adaptation for multichannel audio synthesis," in *Conf. Record 36th Asilomar Conf Signals, Systems and Computers*, vol. 1, Pacific Grove, CA, Nov. 2002, pp. 227–232.

[18] E. Moulines and F. Charpentier, "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech Commun.*, vol. 9, no. 5/6, pp. 453–467, 1990.

[19] R. J. McAulay and T. F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-34, pp. 744–754, Aug. 1986.

[20] Y. Stylianou, "Applying the harmonic plus noise model in concatenative speech synthesis," *IEEE Trans. Speech Audio Process.*, vol. 9, no. 1, pp. 21–29, Jan. 2001.

[21] E. B. George and M. J. T. Smith, "Speech analysis/synthesis and modification using an analysis-by-synthesis/overlap-add sinusoidal model," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 5, pp. 389–406, Sep. 1997.

[22] V. V. Digalakis, D. Rtischev, and L. G. Neumeyer, "Speaker adaptation using constrained estimation of Gaussian mixtures," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 5, pp. 357–366, Sep. 1995.

[23] V. D. Diakoloukas and V. V. Digalakis, "Maximum-likelihood stochastic-transformation adaptation of hidden Markov models," *IEEE Trans. Speech Audio Process.*, vol. 7, no. 2, pp. 177–187, Mar. 1999.

[24] C. Mokbel and G. Chollet, "Word recognition in the car—Speech enhancement/spectral transformation," in *Proc. IEEE lnt. Conf. Acoustics. Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada, Apr. 1991, pp. 925–928.

[25] L. Neumeyer and M. Weintraub, "Probabilistic optimum filtering for robust speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Adelaide, Australia, Apr. 1994, pp. 417–420.

[26] K. Shikano, S. Nakamura, and M. Abe, "Speaker adaptation and voice conversion by codebook mapping," in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, Jun. 1991, pp. 594–597.

[27] S. Nakamura and K. Shikano, "Speaker adaptation applied to HMM and neural networks," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Glasgow, U.K., May 1989, pp. 89–92.

[28] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Salt Lake City, UT, May 2001, pp. 805–808.

[29] K. Tokuda, H. Zen, and A. W. Black, "An HMM-based speech synthesis system applied to english," in *IEEE Workshop on Speech Synthesis*, Santa Monica, CA, Sep. 2002, pp. 227–230.

[30] A. Mouchtaris, J. Van der Spiegel, and P. Mueller, "Non-parallel training for voice conversion by maximum likelihood constrained adaptation," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Montreal, QC, Canada, May 2004, pp. 1–4.

[31] A. Kain, "High resolution voice transformation," Ph.D. dissertation, OGI School Sci. Eng., Oregon Health Sci. Univ., Portland, OR, 2001.

Athanasios Mouchtaris (S'02–M'04) received the Diploma degree in electrical engineering from Aristotle University of Thessaloniki, Thessaloniki, Greece, and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, in 1997, 1999, and 2003, respectively.

From 2003 to 2004, he was a Postdoctoral Researcher in the Electrical and Systems Engineering Department, University of Pennsylvania, Philadelphia. He is currently a Postdoctoral Researcher in the Institute of Computer Science of the Foundation for Research and Technology—Hellas (ICS-FORTH), Heraklion, Crete. He is also a Visiting Professor in the Computer Science Department, University of Crete. His research interests include signal processing for immersive audio environments, spatial audio rendering, multichannel audio modeling, speech synthesis with emphasis on voice conversion, and speech enhancement.

Dr. Mouchtaris is a member of Eta Kappa Nu.

Jan Van der Spiegel (M'72–SM'90–F'02) received the Masters degree in electromechanical engineering and the Ph.D. degree in electrical engineering from the University of Leuven, Leuven, Belgium, in 1974 and 1979, respectively.

He is currently a Professor of the Electrical and Systems Engineering Department, and the Director of the Center for Sensor Technologies at the University of Pennsylvania, Philadelphia. His primary research interests are in high-speed, low-power analog and mixed-mode VLSI design, biologically based sensors and sensory information processing systems, microsensor technology, and analog-to-digital converters. He is the author of over 160 journal and conference papers and holds four patents.

Dr. Van der Spiegel is the recipient of the IEEE Third Millennium Medal, the UPS Foundation Distinguished Education Chair, and the Bicentennial Class of 1940 Term Chair. He received the Christian and Mary Lindback Foundation and the S. Reid Warren Award for Distinguished Teaching, and the Presidential Young Investigator Award. He has served on several IEEE program committees (IEDM, ICCD, ISCAS, and ISSCC) and is currently the Technical Program Vice-Chair of the International Solid-State Circuit Conference (ISSCC2006). He is an elected member of the IEEE Solid-State Circuits Society and is also the SSCS chapters Chairs Coordinator and former Editor of Sensors and Actuators A for North and South America. He is a member of Phi Beta Delta and Tau Beta Pi.

Paul Mueller received the M.D. degree from Bonn University, Bonn, Germany.

He was formerly with the Rockefeller University, New York, and the University of Pennsylvania, Philadelphia, and is currently Chairman of Corticon, Inc. He has worked on ion channels, lipid bilayers, neural processing of vision, and acoustical patterns and VLSI implementation of neural systems.