A Probabilistic Interaction Model for Multipitch Tracking With Factorial Hidden Markov Models

Michael Wohlmayr, *Student Member, IEEE,*
Michael Stark, *Student Member, IEEE,* and Franz Pernkopf, *Member, IEEE*

Abstract

We present a simple and efficient feature modeling approach for tracking the pitch of two simultaneously active speakers. We model the spectrogram features of single speakers using Gaussian mixture models in combination with the minimum description length model selection criterion. To obtain a probabilistic representation for the speech mixture spectrogram features of both speakers, we employ the mixture maximization model (MIXMAX) and, as an alternative, a linear interaction model. A factorial hidden Markov model is applied for tracking pitch over time. This statistical model can be used for applications beyond speech, whenever the interaction between individual sources can be represented as MIXMAX or linear model. For tracking, we use the loopy max-sum algorithm, and provide empirical comparisons to exact methods. Furthermore, we discuss a scheduling mechanism of loopy belief propagation for online tracking. We demonstrate experimental results using Mocha-TIMIT as well as data from the speech separation challenge provided by Cooke et al. We show the excellent performance of the proposed method in comparison to a well known multipitch tracking algorithm based on correlogram features. Using speaker dependent models, the proposed method improves the accuracy of correct speaker assignment, which is important for single-channel speech separation. In particular, we are able to reduce the overall tracking error by 51% relative for the speaker dependent case. Moreover,

Manuscript received December 28, 2009; revised April 19, 2010. This work was supported by the Austria sciene fund, project number P22488-N23 and S10604-N13.

M. Wohlmayr, M. Stark and F. Pernkopf are with the Signal Processing and Speech Communication Laboratory (SPSC), Graz University of Technology, Austria.

we use the estimated pitch trajectories to perform single-channel source separation, and demonstrate the beneficial effect of correct speaker assignment on speech separation performance.

Index Terms

Speech analysis, multipitch tracking, factorial hidden Markov model (FHMM), mixture maximization, Gaussian mixture model (GMM).

I. Introduction

Estimation and tracking of pitch is an important and ongoing research area in speech and audio signal processing over the last decades.[1] Speech applications based on single pitch estimates involve speech coding [4], prosody analysis [5], speaker identification [6], speech enhancement for hearing aids [7], and speech recognition for tonal languages [8]. Moreover, pitch may serve deaf people as an additional cue for lipreading [9]. Some of the best performing algorithms for single pitch estimation are RAPT [2] and YIN [10]. RAPT extracts a set of candidate peaks from the normalized autocorrelation function (NACF) and tracks the most likely pitch trajectory using the Viterbi algorithm. YIN proposes a series of steps to modify and improve the autocorrelation method used for pitch estimation. Likewise, many other algorithms are based on extracting local maxima from short-time periodicity measures such as the NACF, the average magnitude difference function (AMDF), or modifications thereof. For an early comparison of single pitch estimation methods, we refer the interested reader to [3].

This paper is concerned with estimation of pitch in the more challenging scenario of multiple concurrent speakers. Applications of multipitch tracking of speech involve single-channel speech separation (SCSS) [11], [12] and co-channel speaker identification [13]. Beyond speech, a prominent application of multipitch tracking is the automatic transcription of music (see [14] and references therein). Methods for multipitch tracking of speech include [1], [15], [16], [17] and recently [18]. Since we compare our approach to the method of Wu et al. [1], we shortly

[1] *Pitch* refers to a perceptual quality, i.e. the frequency of a pure sinusoid perceived with the same tone as the given signal segment under investigation. In contrast to this, the *fundamental frequency* f_0 refers to the inverse of the smallest period of a quasi-periodic signal. Although the fundamental frequency correlates well with the perceived pitch of a signal, certain perceptual phenomena such as pitch doubling/halving find no explanatory support by f_0. Hence, the term pitch is mostly used in psychoacoustics. Nevertheless, we use the term pitch in this paper since it is more consistent with previous literature [1], [2], [3].

summarize their method in the following: It is based on the unitary model of pitch perception [19], upon which several improvements are introduced to yield a probabilistic representation of the periodicities in the signal. First, the input signal is decomposed into 128 subbands using a gammatone filterbank, and the amplitude envelope is extracted for high frequency channels (center frequency above 800Hz). The NACF is then computed on frames for every channel. Subsequently, a scheme is employed to discard channels whose periodicity information is likely to be unreliable due to noise. For selecting a low frequency channel, the maximum peak at nonzero lags must exceed a threshold. For high frequency channels, the periodicity information must be consistent with the autocorrelation computed on a larger time frame. If a high frequency channel is selected, an additional peak selection routine is employed. The final set of peaks selected from various channels serves as a basis to create a probabilistic representation of zero, one or two pitch periodicity values at each time frame. In brief, the method calculates the likelihood of pitch periodicities under the given observation for the hypothesis of one and two pitch values. Semi-continuous pitch trajectories are then obtained by tracking these likelihoods using a hidden Markov model (HMM). Although this model provides an excellent performance in terms of accuracy, it is not possible to correctly link each pitch estimate to its source speaker.

In this paper, we aim to follow a quite different approach for multiple pitch tracking which is solely based on a statistical model. In contrast to auditory-based approaches such as [1], explicit heuristics such as peak or channel selection are hidden in the statistical model. In particular, the proposed system for multipitch tracking consists of several modules: i) The spectrogram of each single speaker is modelled using Gaussian mixture models (GMMs), whereas the minimum description length (MDL) [20] criterion is applied to find the optimal number of Gaussian components. Training of GMMs can be based on a large set of different speakers, which results in a speaker independent model. On the other hand, a priori knowledge of speaker specific characteristics can be incorporated to obtain a speaker dependent model. As we will show in the experimental section, speaker dependent models have the advantage to allow a correct assignment of pitch trajectories to their corresponding speakers, which is an important cue for single-channel speech separation. ii) Two different interaction models – the MIXMAX model [21] and a linear model – are explored to obtain a probabilistic representation of the observed speech mixture of both speakers.[2] iii) The statistical observation model of a speech mixture is then used within the framework of factorial hidden Markov models (FHMMs) [22], which provide the natural means for tracking the pitch trajectories of both speakers. FHMMs enable the tracking of the states of multiple hidden Markov processes evolving in parallel over time, where the available observations are considered as a joint effect of all single Markov processes. The explicit factorial nature among the various Markov chains allows the use of more efficient inference algorithms compared to an equivalent HMM. We use the loopy max-sum algorithm to obtain approximate solutions to the inference problem, and discuss a scheduling mechanism for online tracking. Furthermore, we empirically compare the results to exact inference.

The paper is organized as follows: Section II introduces FHMMs for multipitch tracking and establishes the terminology for subsequent sections. Section III presents the MIXMAX as well as the linear interaction model, and details the speaker model based on GMMs. Section IV describes the belief propagation methods used for tracking. Section V discusses the experimental setup and performance results on two different databases, namely Mocha-TIMIT [23] and the GRID corpus [24]. Section VI presents a simple approach to perform SCSS based on the estimated pitch trajectories, and demonstrates performance results. Finally, section VII concludes.

II. Factorial Hidden Markov Models

Factorial hidden Markov models enable the tracking of the states of multiple Markov processes evolving in parallel over time, where the available observations are considered as a joint effect of all single Markov processes. For simplicity, we present the case of two Markov chains depicted as the factor graph in Fig. 1. The hidden state random variables are denoted by $x_k^{(t)}$, where k indicates the Markov chain and t the time frame from 1 to T. Similarly, realizations of observed random variables at t are collected in a D-dimensional vector $\boldsymbol{y}^{(t)} \in \mathbb{R}^D$.[3] Each $x_k^{(t)}$ represents a discrete random variable with state space X and cardinality $|X|$. The edges between nodes indicate a conditional dependency between random variables. Specifically, the dependency of hidden variables between two consecutive time instances is defined for each Markov chain by the transition probability $p(x_k^{(t)}|x_k^{(t-1)})$. The dependency of the observed variable $\boldsymbol{y}^{(t)}$ on hidden

[2] Note that our approach can in general be extended to more than two speakers.

[3] Note that boldface symbols denote vectors throughout the manuscript.

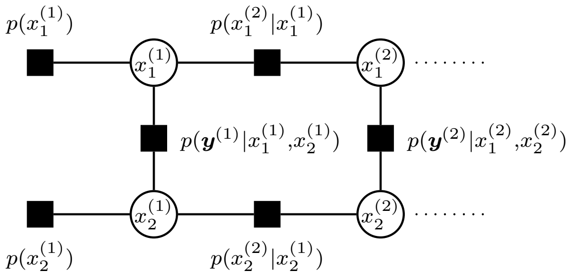

Fig. 1. A factorial HMM shown as a factor graph [25]. Factor nodes are depicted as shaded rectangles together with their functional description. Hidden variable nodes are shown as circles. Here, observed variables $\boldsymbol{y}^{(t)}$ are absorbed into factor nodes.

variables $x_1^{(t)}$ and $x_2^{(t)}$ is defined by the observation probability $p(\boldsymbol{y}^{(t)}|x_1^{(t)}, x_2^{(t)})$. Finally, the prior distribution of the hidden variables in every chain is denoted by $p(x_k^{(1)})$. Denoting the whole sequence of variables, i.e. $\{x^{(t)}\} = \bigcup_{t=1}^{T}\{x_1^{(t)}, x_2^{(t)}\}$ and $\{\boldsymbol{y}^{(t)}\} = \bigcup_{t=1}^{T}\boldsymbol{y}^{(t)}$, the joint distribution of all variables is given by

$$p(\{x^{(t)}\}, \{\boldsymbol{y}^{(t)}\}) = p(\{x^{(t)}\})p(\{\boldsymbol{y}^{(t)}\}|\{x^{(t)}\}) =$$
$$\prod_{k=1}^{2}\left[p(x_k^{(1)})\prod_{t=2}^{T}p(x_k^{(t)}|x_k^{(t-1)})\right]\prod_{t=1}^{T}p(\boldsymbol{y}^{(t)}|x_1^{(t)}, x_2^{(t)}).$$

The number of possible hidden states per time frame is $|X|^2$. As pointed out in [22], this could also be accomplished by an ordinary HMM. The main difference, however, is the constraint placed upon the transition structure. While an HMM with $|X|^2$ states would allow any $|X|^2\times|X|^2$ transition matrix between two hidden states, the FHMM is restricted to two $|X| \times |X|$ transition matrices.

As in most previous work for multipitch tracking [1], [26], [27], we restrict ourselves to two simultaneously speaking subjects, i.e. two Markov chains. Each Markov chain models the pitch trajectory of one speaker, hence the hidden variable $x_k^{(t)}$ denotes the pitch state of speaker k at time t. Each hidden variable has $|X| = 170$ states, where state value '1' refers to 'no pitch' (i.e. unvoiced or silent), and state values '2'-'170' encode different pitch frequencies ranging from 80 to 500Hz. Specifically, the pitch value corresponding to state $x \in \{2, ..., 170\}$ is $f_0 = \frac{f_s}{30+x}$, where the sampling rate $f_s = 16$kHz. Similar to [1], this results in a nonuniform quantization of the pitch interval, where low pitch values have a more fine-grained resolution than high pitch values.

The observation probabilities $p(\boldsymbol{y}^{(t)}|x_1^{(t)}, x_2^{(t)})$ are obtained by using speaker interaction models, as described in the next section. The construction of transitions $p(x_k^{(t)}|x_k^{(t-1)})$ and priors $p(x_k^{(1)})$ will be described in section V-B.

III. The Speaker Interaction Model

At each time frame t, the FHMM models the feature vector $\boldsymbol{y}^{(t)}$ extracted from the mixture signal by the observation probability $p(\boldsymbol{y}^{(t)}|x_1^{(t)}, x_2^{(t)})$. Recently, we modelled the spectrogram features for each pitch pair with one individual GMM [28]. In this work, however, the design of $p(\boldsymbol{y}^{(t)}|x_1^{(t)}, x_2^{(t)})$ is guided by the insight that feature vectors based on the magnitude spectrogram or log-spectrogram can be approximated by an *interaction model* f: $\boldsymbol{y}^{(t)} \approx f(\boldsymbol{s}_1^{(t)}, \boldsymbol{s}_2^{(t)})$, where $\boldsymbol{s}_i^{(t)}$ is the corresponding feature vector resulting from a single speaker. Thus we can approximate the desired observation probability using statistical models of single speakers. In the following, we provide details on two specific interaction models.

A. The Mixture-Maximization Model

The mixture-maximization (MIXMAX) model was originally proposed in [21] for noise robust speech recognition. Since then, it has been used for simultaneous recognition of cochannel speech [29], speech enhancement [30], SCSS [31], speaker identification [32] and joint single-channel speech separation and recognition [33]. It is based on the insight that the log-spectrum of two speakers can be approximated by their elementwise maximum [21]. Specifically, for each time instant t,

$$\boldsymbol{y}^{(t)} \approx \max(\boldsymbol{s}_1^{(t)}, \boldsymbol{s}_2^{(t)}), \tag{1}$$

where $\boldsymbol{s}_i^{(t)}$ is the log-spectrum of speaker i. The underlying assumption is the sparse nature of speech in time-frequency representations, i.e. each particular time-frequency cell of a mixed-speech spectrogram is dominated by a single speaker – this is valid with high probability. This leads to the notion of binary masks in computational auditory scene analysis (CASA) [34] and SCSS [31]. In [35], it is shown that (1) is a nonlinear minimum mean square error (MMSE) estimator of the mixture log-spectrum assuming that the phase of both sources has uniform distribution. We might think of $\boldsymbol{y}^{(t)}$ being generated by the stochastic model shown in Fig. 2. For a pitch value related to state $x_i^{(t)}$, speaker i generates a log-spectrum, $\boldsymbol{s}_i^{(t)}$, that is randomly

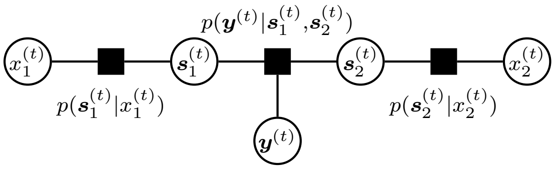
$p(\boldsymbol{y}^{(t)}|\boldsymbol{s}_1^{(t)},\boldsymbol{s}_2^{(t)})$
$x_1^{(t)}$
$\boldsymbol{s}_1^{(t)}$
$\boldsymbol{s}_2^{(t)}$
$x_2^{(t)}$
$p(\boldsymbol{s}_1^{(t)}|x_1^{(t)})$
$p(\boldsymbol{s}_2^{(t)}|x_2^{(t)})$
$\boldsymbol{y}^{(t)}$

Fig. 2. Pitch dependent generation of the mixture log-spectrum $\boldsymbol{y}^{(t)}$. Both speakers produce a log-spectrum $\boldsymbol{s}_i^{(t)}$ in dependency on pitch state $x_i^{(t)}$. The observed log-spectrum $\boldsymbol{y}^{(t)}$ of the speech mixture is approximated by the elementwise maximum of both single speaker log-spectra.

drawn from the *single speaker model* $p(\boldsymbol{s}_i^{(t)}|x_i^{(t)})$. Both log-spectra are then combined via the elementwise maximum operator to form the observable log-spectrum $\boldsymbol{y}^{(t)}$. Thus, $p(\boldsymbol{y}^{(t)}|\boldsymbol{s}_1^{(t)}, \boldsymbol{s}_2^{(t)}) = \delta\left(\boldsymbol{y}^{(t)} - \max(\boldsymbol{s}_1^{(t)}, \boldsymbol{s}_2^{(t)})\right)$, where $\delta(\cdot)$ denotes the Dirac delta.

In general, we obtain the observation probability by marginalizing over the unknown single speaker log-spectra:

$$p(\boldsymbol{y}^{(t)}|x_1^{(t)}, x_2^{(t)}) = \int\int p(\boldsymbol{y}^{(t)}|\boldsymbol{s}_1^{(t)}, \boldsymbol{s}_2^{(t)})p(\boldsymbol{s}_1^{(t)}|x_1^{(t)})p(\boldsymbol{s}_2^{(t)}|x_2^{(t)})d\boldsymbol{s}_1^{(t)}d\boldsymbol{s}_2^{(t)}. \tag{2}$$

For the sake of brevity, we omit the explicit dependence of random variables on t, where appropriate. We use GMMs to model the state-conditional single speaker spectra of each speaker, $i \in \{1, 2\}$, according to

$$p(\boldsymbol{s}_i|x_i) = p(\boldsymbol{s}_i|\boldsymbol{\Theta}_{i,x_i}) = \sum_{m=1}^{M_{i,x_i}} \alpha_{i,x_i}^m \mathcal{N}\left(\boldsymbol{s}_i|\boldsymbol{\theta}_{i,x_i}^m\right), \tag{3}$$

where $M_{i,x_i} \geq 1$ is the number of mixture components, and α_{i,x_i}^m corresponds to the weight of each component $m = 1, \dots, M_{i,x_i}$. These weights are constrained to be positive, $\alpha_{i,x_i}^m \geq 0$, and $\sum_{m=1}^{M_{i,x_i}} \alpha_{i,x_i}^m = 1$. The corresponding GMM for pitch x_i is fully specified by the parameter set $\boldsymbol{\Theta}_{i,x_i} = \left\{\alpha_{i,x_i}^m, \boldsymbol{\theta}_{i,x_i}^m\right\}_{m=1}^{M_{i,x_i}}$, where $\boldsymbol{\theta}_{i,x_i}^m = \left\{\boldsymbol{\mu}_{i,x_i}^m, \boldsymbol{\Sigma}_{i,x_i}^m\right\}$. We assume that the covariance matrices $\boldsymbol{\Sigma}_{i,x_i}^m$ are diagonal.

Given a set of N_i log-spectra from speaker i, $\mathcal{S}_i = \{\boldsymbol{s}_i^{(1)}, ..., \boldsymbol{s}_i^{(N_i)}\}$, together with corresponding reference pitch labels, $\{x_i^{(1)}, ..., x_i^{(N_i)}\}$ we can easily learn a speaker dependent GMM $p\left(\boldsymbol{s}_i|\boldsymbol{\Theta}_{i,x_i}\right)$ for each pitch state x_i, and each speaker i, using the EM algorithm [36]. Accordingly, we have to determine 170 GMMs for each speaker, i.e. one GMM for each pitch state x_i. Further, we use MDL [20], [37] to determine the number of components of each GMM automatically. We denote the set of training samples for pitch state x_i as $\mathcal{S}_{i,x_i} = \{\boldsymbol{s}_i^{(k)}|x_i^{(k)} = x_i\}$, and $|\mathcal{S}_{i,x_i}|$ is the size of the set. For each $\mathcal{S}_{i,x_i}$, we train a range of candidate GMMs with different number of components, and select the GMM which minimizes

$$\mathrm{MDL}(\boldsymbol{\Theta}_{i,x_i}) = -\log p\left(\mathcal{S}_{i,x_i}|\boldsymbol{\Theta}_{i,x_i}\right) + \frac{(2D+1)M_{i,x_i}}{2}\log|\mathcal{S}_{i,x_i}|,$$

where the first term denotes the log-likelihood for the training data, i.e. $\log p\left(\mathcal{S}_{i,x_i}|\boldsymbol{\Theta}_{i,x_i}\right) = \sum_{\boldsymbol{s}_i\in\mathcal{S}_{i,x_i}}\log p(\boldsymbol{s}_i|\boldsymbol{\Theta}_{i,x_i})$, and the second term relates to the complexity of the model with respect to the available data. Indeed, MDL is a method to find the optimal tradeoff between data-fit and model complexity.

Hence, by introducing speaker specific GMMs in (2) and marginalizing over $\boldsymbol{s}_i$, we obtain the pitch conditional observation probability

$$p(\boldsymbol{y}|x_1,x_2) = \sum_{m=1}^{M_{1,x_1}}\sum_{n=1}^{M_{2,x_2}}\alpha_{1,x_1}^{m}\alpha_{2,x_2}^{n}\prod_{d=1}^{D}\left\{\mathcal{N}(y_d|\theta_{1,x_1}^{m,d})\Phi(y_d|\theta_{2,x_2}^{n,d}) + \Phi(y_d|\theta_{1,x_1}^{m,d})\mathcal{N}(y_d|\theta_{2,x_2}^{n,d})\right\}, \quad (4)$$

where y_d gives the d^{th} element of $\boldsymbol{y}$, $\theta_{i,x_i}^{m,d}$ gives the d^{th} element of the corresponding mean and variance, and $\Phi(y|\theta) = \int_{-\infty}^{y}\mathcal{N}(x|\theta)dx$ denotes the univariate cumulative normal distribution. A detailed derivation of (4) is provided in the appendix.

B. The Linear Interaction Model

As an alternative to the MIXMAX approach, we can directly model the magnitude spectrum of a speech mixture. Denoting the short-time magnitude spectrum of speaker $i \in \{1,2\}$ at time t by $\tilde{\boldsymbol{s}}_i^{(t)}$, we approximate the resulting short-time magnitude spectrum of the speech mixture by

$$\tilde{\boldsymbol{y}}^{(t)} \approx \tilde{\boldsymbol{s}}_1^{(t)} + \tilde{\boldsymbol{s}}_2^{(t)}.$$

To obtain an observation model, we make use of the fact that the sum of two independent random variables is modelled by the convolution of their individual probability densities, i.e. $p(\tilde{\boldsymbol{y}}|x_1,x_2) = p(\tilde{\boldsymbol{s}}_1|x_1) * p(\tilde{\boldsymbol{s}}_2|x_2)$ [38], where $*$ denotes the convolution operator. Further, the convolution of two Gaussian densities results again in a Gaussian, with mean and covariance matrix being the sum of the individual means and covariances, respectively. Hence, $\mathcal{N}(\tilde{\boldsymbol{y}}|\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2) = \mathcal{N}(\tilde{\boldsymbol{s}}_1|\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1) * \mathcal{N}(\tilde{\boldsymbol{s}}_2|\boldsymbol{\mu}_2,\boldsymbol{\Sigma}_2)$. This easily extends to GMMs, as the convolution of two GMMs results in a mixture of all pairwise convolved component densities. Similar as in the MIXMAX model, we train speaker dependent GMMs to model the magnitude spectrum,

$p\left(\tilde{\boldsymbol{s}}_i|\boldsymbol{\Theta}_{i,x_i}\right)$. Then, we obtain the observation model as

$$p(\tilde{\boldsymbol{y}}|x_1,x_2)=\sum_{m=1}^{M_{1,x_1}}\sum_{n=1}^{M_{2,x_2}}\alpha_{1,x_1}^{m}\alpha_{2,x_2}^{n}\mathcal{N}\left(\tilde{\boldsymbol{y}}|\boldsymbol{\mu}_{1,x_1}^{m}+\boldsymbol{\mu}_{2,x_2}^{n},\boldsymbol{\Sigma}_{1,x_1}^{m}+\boldsymbol{\Sigma}_{2,x_2}^{n}\right). \tag{5}$$

IV. Tracking

Given the set of observations $\{\boldsymbol{y}^{(t)}\}$, the task of tracking involves searching the sequence of hidden states $\{x^{(t)}\}^{*}$ that maximizes the conditional distribution:

$$\{x^{(t)}\}^{*}=\arg\max_{\{x^{(t)}\}}p(\{x^{(t)}\}|\{\boldsymbol{y}^{(t)}\}). \tag{6}$$

For HMMs, the exact solution to this problem is found by the Viterbi algorithm. Although an FHMM could be expressed by an equivalent HMM, more efficient tracking algorithms exploit the explicit factorization into individual Markov chains. The junction tree algorithm [39] provides an exact solution for FHMMs. However, its computational complexity increases exponentially with the number of hidden Markov chains. Several algorithms are derived in [22] from the framework of variational inference to obtain approximate solutions for the sake of reduced complexity. The sum-product algorithm [25] can be derived under a similar setting of variational principles [40], although more intuitive derivations exist for graphs without loops. When applied on a graph with loops, as is the case of FHMMs, the solutions are in general not guaranteed to converge and can only approximate the optimal solution. For a detailed discussion, we refer the interested reader to [25], [41], [40].

In this work, we use both the exact junction tree algorithm as well as the max-sum algorithm (a variant of the sum-product algorithm) to solve (6). Moreover, we propose a message passing schedule for the max-sum algorithm to enable online tracking. In the experiments, we compare the performance of all presented inference methods in terms of accuracy and computation time.

A. Junction Tree Algorithm

Exact inference on arbitrary graphical models is usually accomplished by first transforming the graphical model into a junction tree, where then belief propagation is performed [41], [42]. For the problem of finding the marginal distribution

$$p(x_i^{(t)}|\{\boldsymbol{y}^{(t)}\})=\sum_{\{x^{(t)}\}\backslash x_i^{(t)}}p(\{x^{(t)}\}|\{\boldsymbol{y}^{(t)}\}), \tag{7}$$

Ghahramani and Jordan [22] provide an exact inference algorithm for FHMMs based on the junction tree algorithm.[4] We present the equivalent formulation on the max-sum semiring [43] in Fig. 3,[5] which provides an exact solution to (6). The computational complexity (without considering the computation of $p(\boldsymbol{y}|x_1, x_2)$) is $O(TK|X|^{K+1})$, where K is the number of Markov chains. For $K = 2$, as in our case, tracking is still tractable.

B. Max-Sum Algorithm

The max-sum algorithm is based on passing messages between nodes of a graph. Among various types of graphs, factor graphs [25] have become a popular tool to depict the mechanisms of message passing. Consider again Fig. 1, which shows an FHMM as factor graph. The functional dependency of each variable node, for brevity called x, is made explicit by "factor nodes", shown as shaded rectangles, i.e. each rectangle denotes a function $f(\{\hat{x}\})$ of its adjacent (i.e. neighboring) variable nodes $\{\hat{x}\}$. For the max-sum algorithm, each node sends to every neighbor a vector valued message $\boldsymbol{\mu}$, which is itself a function of the messages it received, (as well as $f(\{\hat{x}\})$, for the case of a factor node). When applied to factor graphs with loops, message passing results in an iterative procedure (*loopy* max-sum algorithm). A message from variable node x to factor node f is

$$\mu_{x\to f}(x) \quad = \sum_{g\in n(x)\backslash f} \mu_{g\to x}(x), \tag{8}$$

while a message from factor f to variable x is

$$\mu_{f\to x}(x) \quad = \quad \max_{\{\hat{x}\}\backslash x} \left(\ln f(\{\hat{x}\}) + \sum_{y\in\{\hat{x}\}\backslash x} \mu_{y\to f}(y) \right). \tag{9}$$

Here, $n(x)$ denotes the set of neighbor nodes of x. We re-normalize each message μ such that $\sum_{i=1}^{|X|} e^{\mu(i)} = 1$. Although this does not influence the final results, it ensures the numerical stability of the message passing scheme [44]. We restrict each node to send a maximum of 15 messages

[4] For two sets A and B, $A\backslash B$ refers to the set difference. The notation $\sum_{\{a_i\}}$ denotes a nested sum, where one summation is performed for each element in $\{a_i\}$. I.e. we sum in (7) over all hidden nodes except $x_i^{(t)}$.

[5] Informally, a semiring is an algebraic structure defined as a set K, together with two binary operations over elements of that set. Among other requirements, the binary operations must satisfy the distributive law. As shown in [43], the sum-product algorithm can be translated to a semiring involving other binary operations. In other words, the algorithmic framework for the problem 'sum of products' can be translated to obtain an algorithm for the problem 'maximum of sums'.

```
Input: $\boldsymbol{y}^{(t)} \quad \forall\, t \in \{1, ..., T\}$
Output: $\left(x_1^{(t)^*}, x_2^{(t)^*}\right) \quad \forall\, t \in \{1, ..., T\}$
Initialization: Compute likelihoods $p\left(\boldsymbol{y}^{(t)}|x_1^{(t)}, x_2^{(t)}\right) \quad \forall\, t \in \{1, ..., T\}$
$\gamma^{(1)}\left(x_1^{(1)}, x_2^{(1)}\right) \leftarrow \ln p\left(x_1^{(1)}\right) + \ln p\left(x_2^{(1)}\right) + \ln p\left(\boldsymbol{y}^{(1)}|x_1^{(1)}, x_2^{(1)}\right)$
Forward recursion:
for $\forall t \in \{2, ..., T\}$ do
    $\gamma_1^{(t)}\left(x_1^{(t)}, x_2^{(t-1)}\right) \leftarrow \max_{x_1^{(t-1)}}\left[\ln p\left(x_1^{(t)}|x_1^{(t-1)}\right) + \gamma^{(t-1)}\left(x_1^{(t-1)}, x_2^{(t-1)}\right)\right]$
    $\beta_1^{(t)}\left(x_1^{(t)}, x_2^{(t-1)}\right) \leftarrow \arg\max_{x_1^{(t-1)}}\left[\ln p\left(x_1^{(t)}|x_1^{(t-1)}\right) + \gamma^{(t-1)}\left(x_1^{(t-1)}, x_2^{(t-1)}\right)\right]$
    $\gamma_2^{(t)}\left(x_1^{(t)}, x_2^{(t)}\right) \leftarrow \max_{x_2^{(t-1)}}\left[\ln p\left(x_2^{(t)}|x_2^{(t-1)}\right) + \gamma_1^{(t)}\left(x_1^{(t)}, x_2^{(t-1)}\right)\right]$
    $\beta_2^{(t)}\left(x_1^{(t)}, x_2^{(t)}\right) \leftarrow \arg\max_{x_2^{(t-1)}}\left[\ln p\left(x_2^{(t)}|x_2^{(t-1)}\right) + \gamma_1^{(t)}\left(x_1^{(t)}, x_2^{(t-1)}\right)\right]$
    $\gamma^{(t)}\left(x_1^{(t)}, x_2^{(t)}\right) \leftarrow \gamma_2^{(t)}\left(x_1^{(t)}, x_2^{(t)}\right) + \ln p\left(\boldsymbol{y}^{(t)}|x_1^{(t)}, x_2^{(t)}\right)$
end for
$\left(x_1^{(T)^*}, x_2^{(T)^*}\right) \leftarrow \arg\max_{x_1^{(T)}, x_2^{(T)}}\left[\gamma^{(T)}\left(x_1^{(T)}, x_2^{(T)}\right)\right]$
Backtracking:
for $\forall t \in \{T, ..., 2\}$ do
    $x_2^{(t-1)^*} \leftarrow \beta_2^{(t)}\left(x_1^{(t)^*}, x_2^{(t)^*}\right)$
    $x_1^{(t-1)^*} \leftarrow \beta_1^{(t)}\left(x_1^{(t)^*}, x_2^{(t-1)^*}\right)$
end for
```

Fig. 3. The junction tree algorithm for a two-chain FHMM on a max-sum semiring. This algorithm gives the exact solution to (6). For the special case of an HMM (i.e. FHMM with a single Markov chain), this algorithm is equivalent to the well known Viterbi algorithm.

per edge. Further, each node only re-sends a message to a neighbor if it is significantly different from the previously sent message in terms of the Kullback-Leibler-divergence. For initialization, variable nodes send messages with all elements set to zero. After the last iteration, we obtain the maximum a posteriori configuration $p^*(x)$ of each variable node x as a function of its incoming messages:

$$p^*(x) = \max_{\{x^{(t)}\}\backslash x} p(\{x^{(t)}\}|\{\boldsymbol{y}^{(t)}\}) = \sum_{g \in n(x)} \mu_{g \to x}(x). \tag{10}$$

We obtain the approximate solution as the set of individual maxima, $x^* = \arg\max_x p^*(x)\ \forall x \in \{x^{(t)}\}$. Neglecting again the computation of $p(\boldsymbol{y}|x_1, x_2)$, the computational complexity of this approach is $O(TK|X|^K)$, i. e. the complexity of the max-sum algorithm is an order of magnitude lower than for the junction tree algorithm.

We propose two different scheduling strategies for max-sum message passing. First, we perform message passing on the FHMM using a complete speech mixture utterance at once. This is suitable for offline processing of recordings. Second, for online processing, we partition the FHMM into overlapping segments of time frames, $\{\mathcal{T}_1, \mathcal{T}_2, ...\}$, and perform message passing on each individual segment exclusively. This concept is illustrated in Fig. 4. Each segment consists of L time frames, and neighboring segments overlap by $L - S$ frames, i.e. $\mathcal{T}_\tau = \{(\tau - 1)S + 1, (\tau - 1)S + 2, ..., (\tau - 1)S + L\}$. In step τ, we restrict message passing to time frames $t \in \mathcal{T}_\tau$. Variable nodes in $\left\{x_1^{(t)}, x_2^{(t)} | t \in \mathcal{T}_{\tau-1} \cap \mathcal{T}_\tau\right\}$, as well as factor nodes connected to them, have already received messages in the previous step $\tau - 1$. Message passing is continued with those messages, thus enabling information flow from left to right. Similar to the concept of smoothing in e.g. Kalman filters, we wish to incorporate information from future observations at least H time frames ahead. Thus, when message passing has finished in step τ (i.e. each node has sent a maximum of 15 messages per edge), the maximum probability configuration of all variable nodes up to time frame $(\tau - 1)S + L - H$ is evaluated, where H is the lower bound on the smoothing lag. Throughout the experiments, we set parameters to $L = 10$, $S = 4$ and $H = 2$.

V. Experimental Results

TABLE I

LABELS OF SPEAKERS AND FILENAMES USED FOR TESTING ON GRID DATABASE.

FE1	speaker 18	’lwixzs’	’sbil4a’	’prah4s’
FE2	speaker 20	’lwwy2a’	’sbil2a’	’prbu5p’
MA1	speaker 1	’pbbv6n’	’sbwozn’	’prwkzp’
MA2	speaker 2	’lwwm2a’	’sgai7p’	’priv3n’

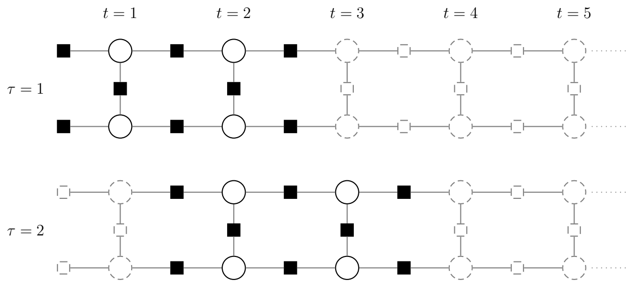

Fig. 4. For online scheduling, message passing is performed on consecutive segments $\{..., \mathcal{T}_\tau, \mathcal{T}_{\tau+1}, ...\}$. In this example, each segment has $L = 2$ time frames, and consecutive segments are shifted by $S = 1$ time frames. Factor and variable nodes with black solid lines are involved in message passing on segment $\mathcal{T}_1 = \{1, 2\}$ and $\mathcal{T}_2 = \{2, 3\}$, respectively. Dashed nodes remain inactive. When all nodes have sent a maximum of 15 messages in step $\tau = 1$, message passing is continued on segment $\mathcal{T}_2$. All nodes depending on time frames in $\mathcal{T}_1 \cap \mathcal{T}_2 = \{2\}$ continue with messages received in step $\tau = 1$. With a supposed smoothing lag $H = 1$ (see text for details), we evaluate after step $\tau = 1$ the maximum probability configuration of variables at $t = 1$.

TABLE II

LABELS OF FEMALE AND MALE SPEAKERS USED FOR TRAINING GENDER DEPENDENT AND SPEAKER INDEPENDENT MODELS ON GRID DATABASE.

	speaker									
FE	4	7	8	11	15	16	21	22	23	24
MA	3	5	6	9	10	12	13	14	17	19

We evaluate the performance of the proposed MIXMAX and linear interaction model, abbreviated as MM and LI, respectively. Both models are combined with one of the three presented tracking methods (max-sum for batch processing, max-sum for online processing, and method based on junction tree algorithm, abbreviated as BA, ON, and JT, respectively), giving a total of six variants. We compare the performance of the proposed methods to the correlogram based method [1], which we call COR-HMM. This method achieves a high accuracy for speech mixtures in difficult signal conditions, and can be applied ad-hoc to a given speech mixture, i.e. no training is required. However, being agnostic to speaker specific information, it does not facilitate a proper assignment of the estimated pitch values to their corresponding speakers. In contrast to this, the proposed methods can incorporate speaker specific information, which helps to identify the correct speaker assignment. Hence, the resulting pitch trajectories are suitable for their use in SCSS [12]. A simple SCSS experiment based on the estimated pitch trajectories is shown in section VI. To allow a proper comparison of our methods to COR-HMM, we use an error measure that is invariant to correct speaker assignment. On the other hand, to evaluate the performance of the proposed methods in terms of successful speaker assignment, we propose a slightly modified error measure. We give details on both error measures below.

A. Data

For experimental comparisons, we used two different databases:

1) The Mocha-TIMIT database [23] consists of 460 English utterances from both a male and a female speaker, sampled at 16kHz. In addition, laryngograph signals are available for all recordings, from which the reference pitch $f_0[t]$ was acquired using the RAPT method [2] together with manual removal of erroneous pitch estimates in nonaudible regions.[6] The speaker dependent GMMs were trained on 400 sentences each, while 60 test instances were obtained by mixing the remaining male and female utterances at 0dB.
2) Two male and two female speakers (abbreviated as MA1, MA2 and FE1, FE2, respectively), were selected from the GRID database [24], and 500 English sentences were selected per speaker. For each speaker, 497 sentences were used to train speaker dependent GMMs, while the remaining three sentences were used for testing, as shown in Table I. Test mixtures were created for each speaker pair, including same-gender mixtures, resulting in a total of 54 test mixtures (9 mixtures for each of the 6 speaker pairs). In addition to speaker dependent (SD) GMMs, gender dependent (GD) GMMs were trained using speakers listed in Table II, where again 497 utterances per speaker were used. Moreover, one set of speaker independent (SI) GMMs was trained using all speakers in Table II. As no laryngograph signals are available for this database, the reference pitch trajectories were obtained directly from the single speech utterances using the RAPT method.

B. Experimental Setup

The features $\boldsymbol{y}^{(t)}$ or $\tilde{\boldsymbol{y}}^{(t)}$ of the proposed methods are based on the log-spectrogram or magnitude spectrogram of the speech mixture, respectively. Given an input signal at sampling rate

[6] An implementation of the RAPT algorithm is provided by the Entropic speech processing system (ESPS) “get_f0” method.

$f_s = 16$kHz, we compute the spectrogram via the 1024 point FFT, using a Hamming window of length 32ms and step size of 10ms. Next, we obtain each observation vector $\tilde{\boldsymbol{y}}^{(t)} \in \mathbb{R}^{64}$ by taking the magnitude of spectral bins 2-65, which corresponds to a frequency range up to 1000Hz.[7] Likewise, we obtain $\boldsymbol{y}^{(t)} = \log \tilde{\boldsymbol{y}}^{(t)}$.

Both transition matrices of the FHMM are obtained by counting the transitions of the reference pitch values from single speaker recordings in the training set. Additionally, we apply Laplace smoothing on both transition matrices.[8] Prior distributions $p(x_k^{(1)})$ are obtained likewise. Again, these priors and transitions are obtained in a SD, GD and SI fashion using the training data as proposed for the corresponding GMMs. Remarkably, we observed during the experiments that performance results are consistently better if the transitions matrices remain unnormalized, i.e. when using $p(x_k^{(t)}, x_k^{(t-1)})$ instead of $p(x_k^{(t)}|x_k^{(t-1)})$. Similar effects on performance results are observed with $p(x_k^{(t)}|x_k^{(t-1)})^{\alpha}$, where the additional tuning parameter $\alpha \approx 2$. Throughout the experiments, however, we use unnormalized transitions. Speaker dependent GMMs are trained on both databases, as described in section III-A and III-B. Moreover, gender dependent and speaker independent experiments are performed on the GRID database. For training the GMMs with MDL, we restricted the maximal number of components per GMM to 20.

C. Performance Measure

For every test instance, each method estimates two pitch trajectories, $\tilde{f}_0^{(1)}[t]$ and $\tilde{f}_0^{(2)}[t]$. To compare the performance of the proposed methods to COR-HMM, we use the error measure proposed in [1]: E_{ij} denotes the percentage of time frames where i pitch points are misclassified as j pitch points, i.e. E_{12} means the percentage of frames with two pitch values estimated whereas only one pitch point is present. For each of the two reference pitch trajectories, $f_0^1[t]$ and $f_0^2[t]$, the corresponding pitch frequency deviation is defined as

$$\Delta f^{(i)}[t] = \min_k \frac{\left| \tilde{f}_0^{(k)}[t] - f_0^{(i)}[t] \right|}{f_0^{(i)}[t]},$$

i.e. at each time instance, the closest of the two estimated pitch points is assigned to a reference pitch trajectory. The gross detection error rate E_{Gross} is the percentage of time frames where

[7] This covers the most relevant frequency range, while keeping the model complexity low.

[8] Laplace smoothing amounts to the initialization of each element of the transition matrix with count one, i.e. adding the prior information that each transition was observed at least once. This smoothes the transition probabilities.

the frequency deviation $\Delta f^{(i)}[t]$ is larger than 20% for one or both references $f_0^{(i)}$. The fine detection error $E_{Fine}^{(i)}$ is the average frequency deviation in percent at time frames where $\Delta f^{(i)}[t]$ is smaller than 20%. The overall error, E_{Total}, is defined as the sum of all error terms:[9] $E_{Total} = E_{01} + E_{02} + E_{10} + E_{12} + E_{20} + E_{21} + E_{Gross} + E_{Fine}$, where $E_{Fine} = E_{Fine}^{(1)} + E_{Fine}^{(2)}$.

To evaluate the performance in terms of successful speaker assignment, we propose a slightly modified error measure. First, each of the two estimated pitch trajectories is assigned to a ground truth trajectory, $f_0^{(1)}[t]$ or $f_0^{(2)}[t]$. From the two possible assignments, $(\tilde{f}_0^{(1)} \to f_0^{(1)}, \tilde{f}_0^{(2)} \to f_0^{(2)})$ or $(\tilde{f}_0^{(1)} \to f_0^{(2)}, \tilde{f}_0^{(2)} \to f_0^{(1)})$, the one is chosen for which the overall quadratic error is smallest. Note that this assignment is not done for each individual time frame, but for the global pitch trajectory. Next, we define the *speaker assigned pitch frequency deviation* as

$$\bar{\Delta} f^{(i)}[t] = \frac{\left|\tilde{f}_0^{(i)}[t] - f_0^{(i)}[t]\right|}{f_0^{(i)}[t]},$$

where $f_0^{(i)}[t]$ denotes the reference chosen for $\tilde{f}_0^{(i)}[t]$. For each reference trajectory, we define the corresponding permutation error $\bar{E}_{Perm}^{i}[t]$ to be one at time frames where the voicing decision for both estimates is correct, but the pitch frequency deviation exceeds 20%, and $\tilde{f}_0^{i}[t]$ is within the 20% error bound of the other reference pitch. This indicates a permutation of pitch estimates due to incorrect speaker assignment. The overall permutation error rate $\bar{E}_{Perm}$ is the percentage of time frames where either $\bar{E}_{Perm}^{1}[t]$ or $\bar{E}_{Perm}^{2}[t]$ is one. Next, we define for each reference trajectory the corresponding gross error $\bar{E}_{Gross}^{i}[t]$ to be one at time frames where the voicing decision is correct, but the pitch frequency deviation exceeds 20% and no permutation error was detected. This indicates inaccurate pitch measurements independent of permutation errors. Again, the overall gross error rate $\bar{E}_{Gross}$ is the percentage of time frames where either $\bar{E}_{Gross}^{(1)}[t]$ or $\bar{E}_{Gross}^{(2)}[t]$ is one. This slightly different definition of the gross error rate ensures that voicing errors or permutation errors do not account for an additional increase in the gross error rate. The fine detection error $\bar{E}_{Fine}^{(i)}$ is the average speaker assigned frequency deviation in percent at time frames where $\bar{\Delta} f^{(i)}[t]$ is smaller than 20%. Finally, the overall error, $\bar{E}_{Total}$, is the sum of all error terms: $\bar{E}_{Total} = E_{01} + E_{02} + E_{10} + E_{12} + E_{20} + E_{21} + \bar{E}_{Gross} + \bar{E}_{Fine} + \bar{E}_{Perm}$, where $\bar{E}_{Fine} = \bar{E}_{Fine}^{(1)} + \bar{E}_{Fine}^{(2)}$.

[9] Note that E_{Total}, as proposed in [1], can be larger than 100%.

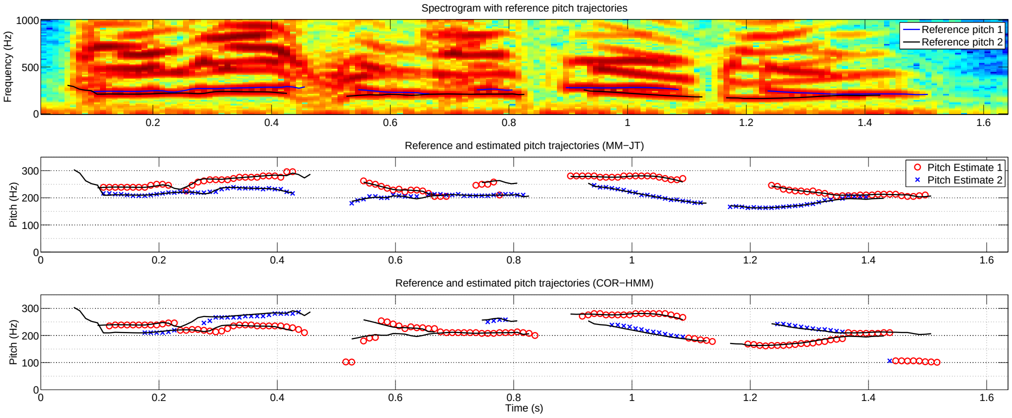

Fig. 5. Tracking results of speaker dependent MM-JT and COR-HMM on GRID test mixture on two female speakers ('lwixzs' and 'lwwy2a'). Top panel: Spectrogram of speech mixture, together with both reference pitch trajectories. Middle panel: Estimated pitch trajectories using MM-JT, together with reference pitch trajectories (black solid lines). Bottom panel: Estimated pitch trajectories using COR-HMM, together with reference pitch trajectories (black solid lines).

D. Results on GRID Database

Table III summarizes the performance of the proposed MM-JT method on the GRID database in terms of E_{Total} for all three training scenarios (SD, GD and SI). For comparison, Table IV shows E_{Total} of COR-HMM on GRID. This compares the accuracy of the proposed MM-JT method to the baseline algorithm COR-HMM, independent of correct speaker assignment. Both tables show that the main contributors to E_{Total} are E_{21} and E_{Gross}. The overall accuracy of MM-JT degrades when using GD or SI instead of SD models, however E_{Total} is still lower than for COR-HMM.

TABLE III
RESULTS FOR MM-JT ON GRID DATABASE. PERFORMANCE IS MEASURED IN TERMS OF E_{Total}.

		E_{01}	E_{02}	E_{10}	E_{12}	E_{20}	E_{21}	E_{Gross}	E_{Fine}	E_{Total}
SD	Mean	0.94	0.01	6.33	2.40	1.72	11.83	18.11	3.27	**44.63**
	Std	1.36	0.08	3.19	1.93	2.56	5.92	7.17	1.22	15.77
GD	Mean	1.71	0.07	5.61	2.94	2.11	15.75	22.01	3.62	**53.81**
	Std	1.70	0.33	3.79	3.22	2.81	7.78	9.22	1.36	17.92
SI	Mean	2.43	0.06	5.44	3.45	1.96	15.29	21.25	3.64	**53.50**
	Std	2.06	0.18	3.56	2.62	2.50	7.36	8.69	1.34	16.94

TABLE IV

RESULTS FOR COR-HMM ON GRID DATABASE. PERFORMANCE IS MEASURED IN TERMS OF E_{Total}.

	E_{01}	E_{02}	E_{10}	E_{12}	E_{20}	E_{21}	E_{Gross}	E_{Fine}	E_{Total}
Mean	1.00	0.08	8.46	0.87	2.56	19.97	27.10	2.81	**62.85**
Std	1.49	0.20	3.62	1.23	3.11	7.70	7.85	1.53	14.73

TABLE V

RESULTS FOR MM-JT ON GRID DATABASE. PERFORMANCE IS MEASURED IN TERMS OF $\bar{E}_{Total}$.

		E_{01}	E_{02}	E_{10}	E_{12}	E_{20}	E_{21}	$\bar{E}_{Gross}$	$\bar{E}_{Fine}$	$\bar{E}_{Perm}$	$\bar{E}_{Total}$
SD	Mean	0.94	0.01	6.33	2.40	1.72	11.83	0.91	2.86	0.51	**27.52**
	Std	1.36	0.08	3.19	1.93	2.56	5.92	1.40	0.80	1.29	10.56
GD	Mean	1.71	0.07	5.61	2.94	2.11	15.75	2.10	3.93	3.96	**38.17**
	Std	1.70	0.33	3.79	3.22	2.81	7.78	1.87	2.24	4.38	12.33
SI	Mean	2.43	0.06	5.44	3.45	1.96	15.29	2.44	4.08	10.95	**46.07**
	Std	2.06	0.18	3.56	2.62	2.50	7.36	2.13	2.32	7.89	13.68

TABLE VI

RESULTS FOR COR-HMM ON GRID DATABASE. PERFORMANCE IS MEASURED IN TERMS OF $\bar{E}_{Total}$.

	E_{01}	E_{02}	E_{10}	E_{12}	E_{20}	E_{21}	$\bar{E}_{Gross}$	$\bar{E}_{Fine}$	$\bar{E}_{Perm}$	$\bar{E}_{Total}$
Mean	1.00	0.08	8.46	0.87	2.56	19.97	1.32	3.30	16.28	**53.83**
Std	1.49	0.20	3.62	1.23	3.11	7.70	1.80	2.79	10.12	12.99

To demonstrate the capability of MM-JT in correctly assigning pitch trajectories to their corresponding speakers, we compare its performance to COR-HMM on GRID using the proposed error measure $\bar{E}_{Total}$ in Table V and VI. Using SD models, we achieve approximately half the value of $\bar{E}_{Total}$ in comparison to COR-HMM. While both methods achieve similar outcomes for $\bar{E}_{Gross}$ and $\bar{E}_{Fine}$, major differences arise in $\bar{E}_{Perm}$ and E_{21}. GD or SI models cause a drop in performance compared to the speaker dependent case. However, we still outperform COR-HMM. Here, mostly E_{21} and $\bar{E}_{Perm}$ are the main contributors to $\bar{E}_{Total}$. Using GD models, we observe a large increase in $\bar{E}_{Perm}$ for same-gender scenarios, while for different-gender scenarios, $\bar{E}_{Perm}$ is significantly lower than for COR-HMM. This indicates the beneficial influence of SD

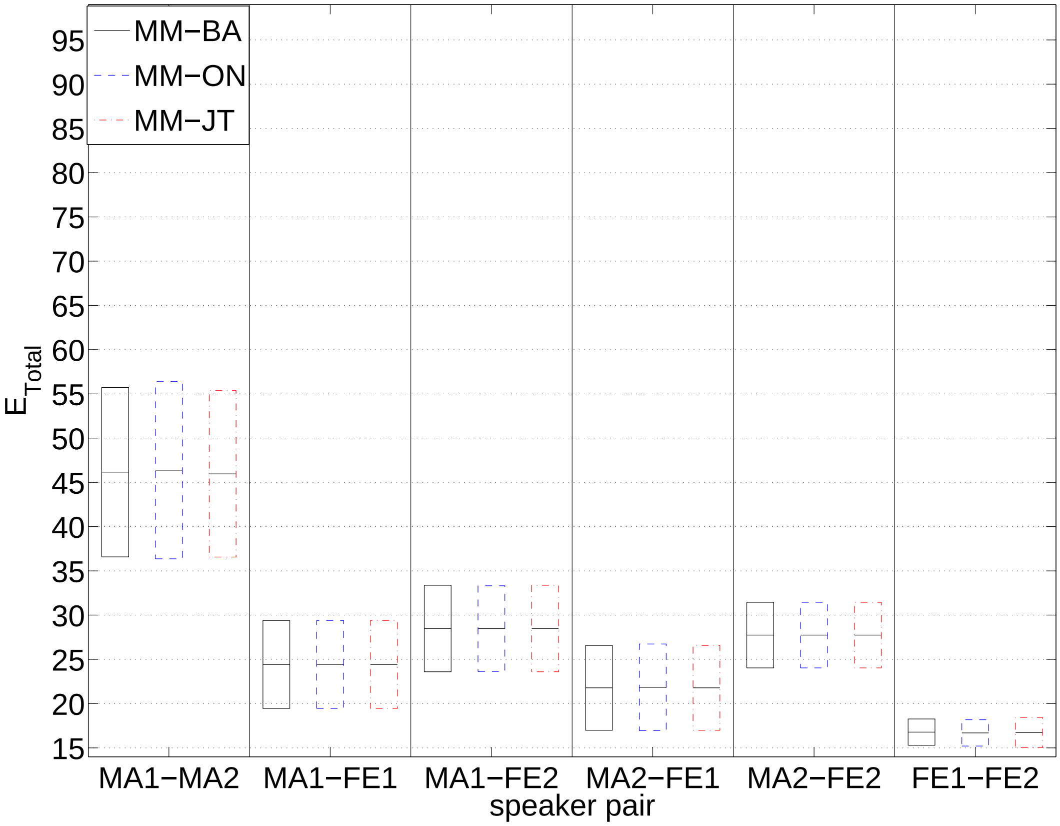

Fig. 6. $\bar{E}_{Total}$ of MIXMAX approach using speaker dependent models and different tracking algorithms on GRID. Each box depicts the mean and standard deviation of a method over 9 test mixtures of a given speaker pair.

or GD models on correct speaker assignment. Fig. 5 depicts the tracking result of MM-JT (using SD models) and COR-HMM on a test mixture of two female speakers. This demonstrates the excellent speaker assignment of our method, provided that prior knowledge of speaker characteristics is available.

All variants of the proposed method are compared in terms of $\bar{E}_{Total}$ only. Fig. 6 compares the performance of various tracking methods, i.e. MM-BA, MM-ON and MM-JT, using speaker dependent models. The performance of all three trackers is essentially equivalent. The situation is somewhat different for gender dependent models shown in Fig. 7. For the same-gender scenario (MA1-MA2, FE1-FE2), the parameters of the FHMM are the same in each Markov chain. Moreover, the observation likelihood is symmetric in x_1 and x_2, i.e. $p(\boldsymbol{y}|x_1, x_2) = p(\boldsymbol{y}|x_2, x_1)$.

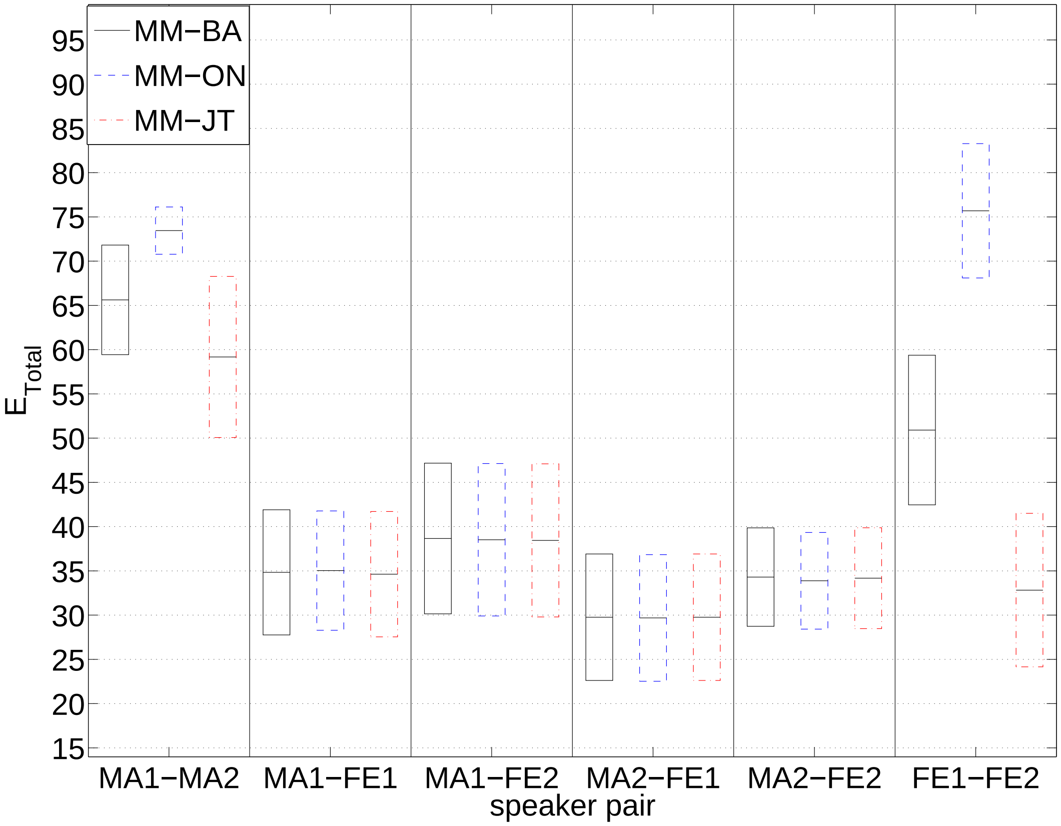

Fig. 7. $\bar{E}_{Total}$ of MIXMAX approach using gender dependent models and different tracking algorithms on GRID. Each box depicts the mean and standard deviation of a method over 9 test mixtures of a given speaker pair.

In that case, we observe that both variants of the loopy max-sum algorithm work significantly worse than the junction-tree algorithm. Fig. 8 compares the performance for speaker independent models. In this case, MM-BA and MM-ON perform worse than MM-JT for all speaker pairs.

To indicate the computation time of the methods involved, measurements were performed on a 2.4 GHz dual core machine with 8 Gb main memory. All algorithms were implemented and tested in Matlab. For computation of the MIXMAX likelihoods in (4), a Matlab-MEX implementation was used. As shown in Table VII, the computational requirements of the MIXMAX likelihoods depend on the particular set of GMMs involved. A comparison with (4) reveals that the computational complexity is mostly determined by the term $C = \sum_{x_1=1}^{|X|} \sum_{x_2=1}^{|X|} M_{1,x_1} M_{2,x_2}$, which depends on the actual set of GMMs involved. Table VII shows the average computation

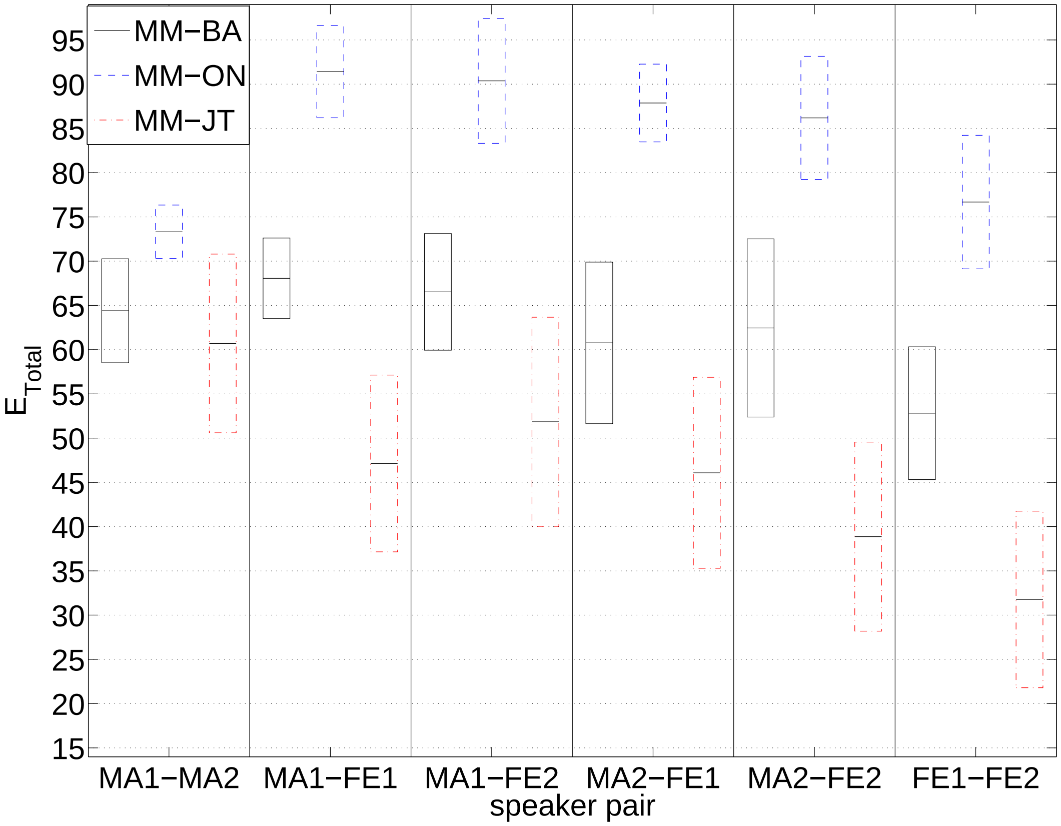

Fig. 8. $\bar{E}_{Total}$ of MIXMAX approach using speaker independent models and different tracking algorithms on GRID. Each box depicts the mean and standard deviation of a method over 9 test mixtures of a given speaker pair.

time for the MIXMAX likelihoods for all three training scenarios, together with the average value of C. Table VIII indicates the computation time for the three different tracking algorithms. In this setting with two speakers, the time performance of JT is comparable to BA. Note however that the computational complexity of JT is an order of magnitude larger than for BA or ON, while the complexity of ON and BA differs only by a constant factor. Thus, for tracking more than two speakers, the computation time for JT is expected to be much higher than for BA or ON.

TABLE VII

AVERAGE COMPUTATION TIME OF MIXMAX LIKELIHOODS IN (4), SHOWN IN SECONDS PER ANALYSIS FRAME. FOR EACH TRAINING SCENARIO, THE AVERAGE OF FACTOR C IS GIVEN.

	C	Time [s]
SD	9.45e5	0.44
GD	6.60e6	2.71
SI	1.09e7	4.38

TABLE VIII

AVERAGE COMPUTATION TIME OF TRACKING ALGORITHMS, SHOWN IN SECONDS PER ANALYSIS FRAME. FOR EACH METHOD, THE MEAN AND STANDARD DEVIATION IS GIVEN.

	JT	BA	ON
Mean	0.15	0.11	0.42
Std	0.01	0.002	0.03

E. Results on Mocha-TIMIT Database

On the Mocha-TIMIT database, each example of the test set was mixed with white Gaussian noise at different SNR conditions, ranging from 40dB down to 0dB in 10dB steps. For each SNR condition, we evaluate the performance of the proposed methods, where the parameters remained optimized for clean speech. Fig. 9 shows E_{Total} for COR-HMM, MM-JT and LI-JT for all noise conditions. Likewise, Fig. 10 shows $\bar{E}_{Total}$ for the same setup. LI-JT and MM-JT have an equivalent performance over a range of SNR conditions down to 20dB, and both significantly outperform COR-HMM. Both proposed interaction models show decreasing performance for lower SNR conditions. At 0dB SNR, MM-JT still outperforms the baseline algorithm, while LI-JT is less robust to noise and performs worse than COR-HMM.

VI. Application to Single-Channel Source Separation

We demonstrate the performance of the proposed multipitch tracking algorithm when applied to the problem of SCSS. Based on the estimated pitch trajectories, we generate a binary mask for each speaker, and recover an estimate of the single speaker utterance by masking the mixed

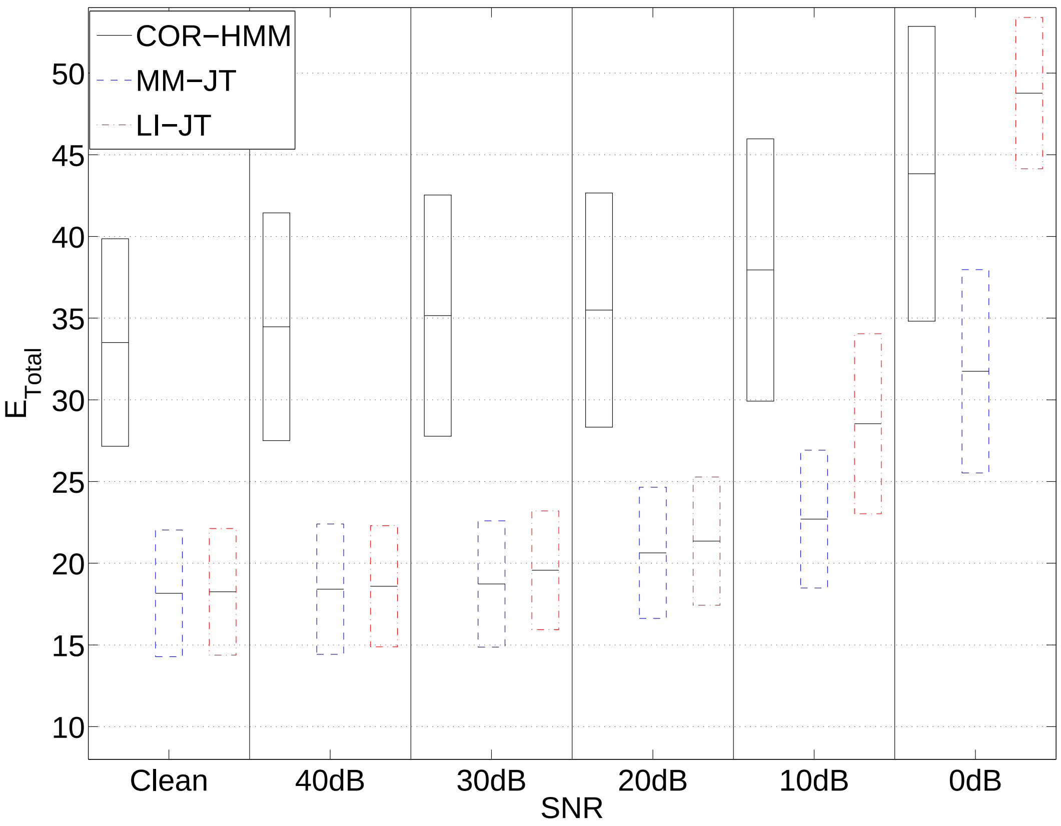

Fig. 9. E_{Total} of COR-HMM, speaker dependent MM-JT, and speaker dependent LI-JT approach on Mocha-TIMIT database with background noise at different SNRs. Each box depicts the mean and standard deviation of a method over 60 male-female mixtures.

speech spectrogram $Y[t, u]$, where u denotes the frequency index. For more advanced SCSS methods, we refer the interested reader to [12], [31], [45].

Specifically, given the pitch trajectory related to speaker i, $\tilde{f}_i[t]$, we synthesise for each t the corresponding excitation signal:

$$e_i[t,n] = \sum_{u=1}^{U(\tilde{\omega}_i[t], f_{max})} \sin\left(\, u\ \tilde{\omega}_i[t]\, n \ + \ \angle Y[t,u]\ \right) + \epsilon[n], \tag{11}$$

where $n = [1, ..., T_a]$ and $T_a = 512$ is the number of samples within the synthesis frame, $\tilde{\omega}_i[t]$ is $\tilde{f}_i[t]$ in radians, $\angle Y[t, u]$ is the phase of the mixed speech spectrogram, and $U(\tilde{\omega}_i[t], f_{max})$ denotes the number of harmonics corresponding to $\tilde{\omega}_i[t]$ up to a predefined frequency $f_{max} = 4$kHz.

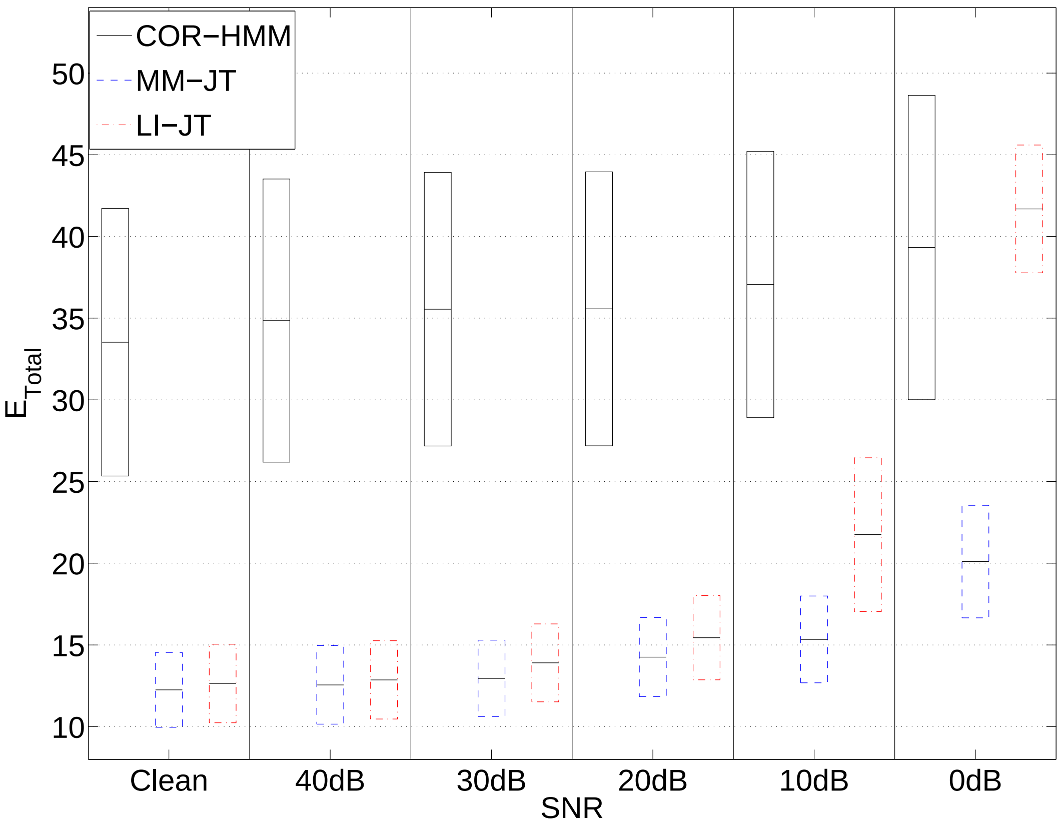

Fig. 10. $\bar{E}_{Total}$ of COR-HMM, speaker dependent MM-JT, and speaker dependent LI-JT approach on Mocha-TIMIT database with background noise at different SNRs. Each box depicts the mean and standard deviation of a method over 60 male-female mixtures.

Further, $\epsilon[n]$ is a Gaussian random signal filtered by a high-pass with cutoff f_{max}. For unvoiced frames (i.e. $\tilde{f}_i[t] = 1$), we set the excitation $e_i[n]$ to a white Gaussian noise sequence of length T_a. Next, we compute the discrete Fourier transform (DFT) on each synthesis frame

$$E_i[t, u] = \text{DFT}\left\{e_i[t, n]\right\},$$

and obtain the binary mask of speaker 1 as

$$\text{BM}_1[t, u] = \begin{cases} 1, & \text{if} \quad |E_1[t, u]| > |E_2[t, u]| \\ 0, & \text{otherwise.} \end{cases}$$

We set the binary mask of speaker 2 to the complement, i.e. $\text{BM}_2 = \overline{\text{BM}}_1$. Using the binary

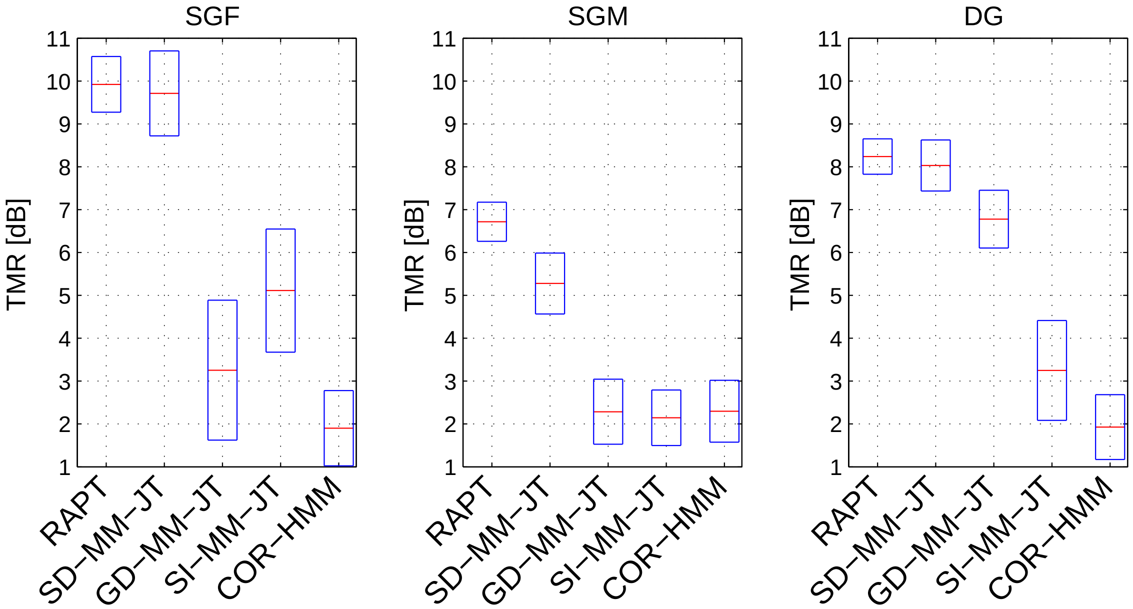

Fig. 11. Performance in terms of the target-to-masker ratio (TMR) for single-channel speech separation. Each panel shows the results using pitch trajectories obtained by five different methods: RAPT denotes the case where the reference pitch trajectories are used, SD-MM-JT, GD-MM-JT and SI-MM-JT denote the pitch trajectories obtained by the proposed MM-JT method in the speaker dependent, gender dependent and speaker independent scenario, respectively, and COR-HMM denotes the case where the pitch trajectories are obtained from the baseline method [1]. The leftmost panel summarizes the TMR for the same-gender female (SGF) scenario, while the middle and rightmost panel summarize results for the same-gender male (SGM) and different gender (DG) scenario, respectively. Each box depicts the mean and standard deviation over all test mixtures and target/masker combinations.

mask, we obtain the estimated spectrogram of speaker i as

$$\tilde{S}_i[t,u] = \mathrm{BM}_i[t,u]\ |Y[t,u]| \exp(j\angle Y[t,u]).$$

From this, we finally re-synthesize the time domain signal.

We use the commonly used target-to-masker (TMR) measure to assess the quality of the separation result:

$$\mathrm{TMR}_i = \frac{\sum_{t,u} S_i^2[t,u]}{\sum_{t,u}(S_i[t,u] - \tilde{S}_i[t,u])^2}, \tag{12}$$

where $S_i[t,u]$ is the clean speech spectrogram of speaker i.

We experimentally evaluate the separation performance on the GRID database in terms of the TMR, using the same experimental setup as introduced in section V. We compare the results obtained by 4 different pitch extraction methods: First, we use the reference pitch trajectories extracted directly from the single speech utterances using RAPT. Next, we use the pitch trajectories obtained from the proposed MM-JT method trained for the speaker dependent, gender dependent and speaker independent scenario (SD-MM-JT, GD-MM-JT and SI-MM-JT), respectively. Finally, we use the pitch trajectories from the baseline COR-HMM method [1]. We show performance results in Fig. 11. For the same-gender female (SGF) and the different-gender (DG) scenario, separation results using SD-MM-JT achieve almost the same TMR as for the reference pitch (RAPT). Speech separation is more difficult for the same-gender male (SGM) scenario, where performance drops relative to RAPT using SD-MM-JT. Results using GD-MM-JT are significantly better than COR-HMM for the DG scenario. Note that these results correlate well with $\bar{E}_{Total}$ and $\bar{E}_{Perm}$ presented in Table V.

VII. Conclusion

We have presented a method for multipitch tracking based on the MIXMAX interaction model as well as a linear interaction model. The performance of the proposed system was compared to a state-of-the-art multipitch tracking algorithm [1]. We investigated the performance of the proposed method using speaker dependent, gender dependent and speaker independent models, and evaluated the robustness of the proposed method to white noise at various SNR conditions. Moreover, we examined the performance in terms of correct speaker assignment, and proposed a new error measure for this purpose. Additionally, we compared the performance using different tracking algorithms, and proposed a loopy max-sum scheduling mechanism for online tracking. Finally, we evaluated the performance of single-channel speech separation based on the estimated pitch trajectories.

For speaker dependent models, the proposed method is able to reduce the error rate $\bar{E}_{Total}$ on average by 51% relative to the baseline method [1]. Moreover, the proposed method significantly improves the correct assignment of pitch trajectories to corresponding speakers, which is important for the task of SCSS. Using gender dependent or speaker independent models, we experience a performance drop relative to speaker dependent models. However, the resulting error rate still outperforms the baseline algorithm. The MIXMAX interaction model achieves a better robustness to additive white Gaussian noise than the linear interaction model. The performance with different tracking methods is mostly identical for the speaker dependent case and when using gender dependent models applied to the different gender scenario. In all other cases, the exact junction tree algorithm clearly outperforms the other tracking variants based on approximate inference.

The advantage of the proposed method is the possibility to integrate a-priori knowledge about speaker characteristics into the statistical model. We have shown that speaker dependent models clearly improve the correct speaker assignment of pitch trajectories, and demonstrated the resulting performance gain for SCSS. The usage of interaction models allows the modelling of all involved speakers independent of each other. Future work will investigate methods to adapt speaker models during processing, i.e. starting with speaker independent models, we will infer speaker relevant information and use this information to adapt towards speaker dependent models.

Appendix

Derivation of GMM based observation probability using the MIXMAX model

Given the density of two vector valued, independent random variables, $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, we seek to derive the density of $\boldsymbol{Y} = \max(\boldsymbol{S}_1, \boldsymbol{S}_2)$. First, it is easy to see that the cumulative distribution of $\boldsymbol{Y}$, $\Phi_{\boldsymbol{Y}}(\boldsymbol{y})$, is given as

$$\begin{aligned}\Phi_{\boldsymbol{Y}}(\boldsymbol{y}) &= p\left(S_{1,1} \leq y_1, ..., S_{1,D} \leq y_D, S_{2,1} \leq y_1, ..., S_{2,D} \leq y_D\right) \\ &= \int_{-\infty}^{y_1} \cdots \int_{-\infty}^{y_D} \int_{-\infty}^{y_1} \cdots \int_{-\infty}^{y_D} p\left(s_{1,1}, ..., s_{1,D}, s_{2,1}, ..., s_{2,D}\right) ds_{1,1} \cdots ds_{1,D}\, ds_{2,1} \cdots ds_{2,D},\end{aligned}$$

where $S_{i,d}$ is the d^{th} element of $\boldsymbol{S}_i$. Due to the independence of $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, we have that

$$\Phi_{\boldsymbol{Y}}(\boldsymbol{y}) = \Phi_{\boldsymbol{S}_1}(\boldsymbol{y})\Phi_{\boldsymbol{S}_2}(\boldsymbol{y}),$$

where $\Phi_{\boldsymbol{S}_i}(\cdot)$ denotes the cumulative distribution with respect to $\boldsymbol{S}_i$. Making the conditional dependency of $\boldsymbol{S}_i$ on pitch state x_i explicit, and using the definition of the GMM in (3), we get

$$\Phi_{\boldsymbol{S}_i}(\boldsymbol{y}|x_i) = \sum_{m=1}^{M_{i,x_i}} \alpha_{i,x_i}^{m} \prod_{d=1}^{D} \Phi\left(y_d|\theta_{i,x_i}^{m,d}\right),$$

where $\Phi(y|\theta) = \int_{-\infty}^{y} \mathcal{N}(x|\theta)dx$ is the univariate cumulative normal distribution. We now obtain the conditional density of $\boldsymbol{Y}$ by partial derivation of its cumulative distribution:

$$\begin{aligned}
p(\boldsymbol{y}|x_1,x_2) &= \frac{\partial^D}{\prod_d \partial y_d}\Phi_{\boldsymbol{Y}}(\boldsymbol{y}|x_1,x_2) = \frac{\partial^D}{\prod_d \partial y_d}\Phi_{\boldsymbol{S}_1}(\boldsymbol{y}|x_1)\Phi_{\boldsymbol{S}_2}(\boldsymbol{y}|x_2) \\
&= \frac{\partial^D}{\prod_d \partial y_d}\left(\sum_{m=1}^{M_{1,x_1}} \alpha_{1,x_1}^m \prod_{d=1}^{D} \Phi\left(y_d|\theta_{1,x_1}^{m,d}\right)\right)\left(\sum_{n=1}^{M_{2,x_2}} \alpha_{2,x_2}^n \prod_{d=1}^{D} \Phi\left(y_d|\theta_{2,x_2}^{n,d}\right)\right) \\
&= \frac{\partial^D}{\prod_d \partial y_d}\sum_{m=1}^{M_{1,x_1}}\sum_{n=1}^{M_{2,x_2}} \alpha_{1,x_1}^m \alpha_{2,x_2}^n \prod_{d=1}^{D} \Phi\left(y_d|\theta_{1,x_1}^{m,d}\right)\Phi\left(y_d|\theta_{2,x_2}^{n,d}\right) \\
&= \sum_{m=1}^{M_{1,x_1}}\sum_{n=1}^{M_{2,x_2}} \alpha_{1,x_1}^m \alpha_{2,x_2}^n \prod_{d=1}^{D} \left\{\mathcal{N}\left(y_d|\theta_{1,x_1}^{m,d}\right)\Phi\left(y_d|\theta_{2,x_2}^{n,d}\right) + \Phi\left(y_d|\theta_{1,x_1}^{m,d}\right)\mathcal{N}\left(y_d|\theta_{2,x_2}^{n,d}\right)\right\}.
\end{aligned}$$

Acknowledgment

The authors would like to thank the anonymous reviewers for their detailed comments which helped to improve the quality of this article.

References

[1] M. Wu, D. Wang, and G. Brown, "A multipitch tracking algorithm for noisy speech," *IEEE Trans. Speech and Audio Processing*, vol. 11, no. 3, pp. 229–241, May 2003.

[2] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," *Kleijn W.B. and Paliwal K.K. [Ed], Speech Coding and Synthesis, Elsevier Science*, pp. 495–518, 1995.

[3] L. Rabiner, M. Cheng, A. Rosenberg, and C. McGonegal, "A comparative performance study of several pitch detection algorithms," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. AASP-24, no. 5, pp. 399–418, Oct. 1976.

[4] R. Salami, C. Laflamme, J.-P. Adoul, and D. Massaloux, "A toll quality 8 kb/s speech codec for the personal communications system (pcs)," *IEEE Trans. Vehicular Technology*, vol. 43, no. 3, pp. 808–816, Aug 1994.

[5] C. Wang, "Prosodic modeling for improved speech recognition and understanding," Ph.D. dissertation, Massachusetts Institute of Technology, 2001.

[6] M. Carey, E. Parris, H. Lloyd-Thomas, and S. Bennett, "Robust prosodic features for speaker identification," in *Fourth International Conference on Spoken Language, 1996. ICSLP 96.*, vol. 3, 1996, pp. 1800–1803.

[7] E. Lindemann and J. Melanson, "Noise reduction system for binaural hearing aid," US Patent 005 651 071A, 1997.

[8] C. Chen, R. Gopinath, M. Monkowski, M. Picheny, and K. Chen, "New methods in continuous mandarin speech recognition," in *Proceedings of the Eurospeech 1997*, 1997, pp. 1543–1546.

[9] S. Rosen, A. Fourcin, and B. Moore, "Voice pitch as an aid to lipreading," *Nature*, vol. 291, pp. 150–152, 1981.

[10] A. de Cheveigne and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *J. Acoust. Soc. Am.*, vol. 111, no. 4, pp. 1917–1930, 2002.

[11] D. Morgan, E. George, L. Lee, and S. Kay, "Cochannel speaker separation by harmonic enhancement and suppression," *IEEE Trans. Speech and Audio Processing*, vol. 5, no. 5, pp. 407–424, 1997.

[12] M. Stark, M. Wohlmayr, and F. Pernkopf, "Source-filter based single channel speech separation using pitch information," *IEEE Trans. Speech and Audio Processing, accepted*, 2010.

[13] Y. Shao and D. Wang, "Co-channel speaker identification using usable speech extraction based on multi-pitch tracking," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2003, pp. 205–208.

[14] A. Klapuri and M. Davy, Eds., *Signal processing methods for music transcription*. Springer, 2006.

[15] J. Le Roux, H. Kameoka, N. Ono, A. de Cheveigne, and S. Sagayama, "Single and multiple F0 contour estimation through parametric spectrogram modeling of speech in noisy environments," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1135 –1145, 2007.

[16] T. Tolonen and M. Karjalainen, "A computationally efficient multipitch analysis model," *IEEE Transactions on Speech and Audio Processing*, vol. 8, pp. 708–716, 2000.

[17] F. Sha and L. K. Saul, "Real-time pitch determination of one or more voices by nonnegative matrix factorization," *Advances in Neural Information Processing Systems*, 2005.

[18] G. Hu and D. Wang, "A tandem algorithm for pitch estimation and voiced speech segregation," *IEEE Transactions on Audio, Speech, and Language Processing*, in press, 2010.

[19] R. Meddis and L. O'Mard, "A unitary model of pitch perception," *J. Acoust. Soc. Am.*, vol. 102, no. 3, pp. 1811–1820, 1997.

[20] G. McLachlan and K. Basford, *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, 1988.

[21] A. Nadas, D. Nahamoo, and M. Picheny, "Speech recognition using noise-adaptive prototypes," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 37, no. 10, pp. 1495–1503, Oct 1989.

[22] Z. Ghahramani and M. Jordan, "Factorial hidden Markov models," *Mach. Learn.*, vol. 29, no. 2-3, pp. 245–273, 1997.

[23] A. Wrench, "A multichannel/multispeaker articulatory database for continuous speech recognition research," *Phonus*, vol. 5, pp. 3–17, 2000.

[24] M. Cooke, J. Barker, S. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *J. Acoust. Soc. Amer.*, vol. 120, pp. 2421–2424, Nov 2005.

[25] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Information Theory*, vol. 47, no. 2, pp. 498–519, Feb 2001.

[26] F. Bach and M. Jordan, "Discriminative training of hidden Markov models for multiple pitch tracking," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005, pp. 489–492.

[27] M. Wohlmayr and F. Pernkopf, "Multipich tracking using a factorial hidden Markov model," in *International Conference on Spoken Language Processing (ICSLP), Interspeech*, 2008, pp. 147–150.

[28] ——, "Finite mixture spectrogram modeling for multipitch tracking using a factorial hidden Markov model," in *International Conference on Spoken Language Processing (ICSLP), Interspeech*, 2009, pp. 1079–1082.

[29] A. P. Varga and R. K. Moore, "Simultaneous recognition of concurrent speech signals using hidden Markov model decomposition," in *European Conference on Speech Communication and Technology (Eurospeech)*, 1991, pp. 1175–1178.

[30] D. Burshtein and S. Gannot, "Speech enhancement using a mixture-maximum model," *IEEE Trans. Speech and Audio Processing*, vol. 10, no. 6, pp. 341–351, Sep 2002.

[31] S. T. Roweis, "Factorial models and refiltering for speech separation and denoising," in *Proceedings of the 8th European Conference on Speech Communication and Technology (Eurospeech)*, 2003, pp. 1009–1012.

[32] R. Rose, E. Hofstetter, and D. Reynolds, "Integrated models of signal and background with application to speaker identification in noise," *IEEE Trans. Speech and Audio Processing*, vol. 2, no. 2, pp. 245–257, Apr 1994.

[33] S. Rennie, J. Hershey, and P. Olsen, “Single-channel speech separation and recognition using loopy belief propagation,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2009, pp. 3845–3848.

[34] D. Wang, *Speech Separation by Humans and Machines*. Springer, 2004, ch. On Ideal Binary Mask As the Computational Goal of Auditory Scene Analysis, pp. 181–197.

[35] M. H. Radfar, R. M. Dansereau, and A. Sayadiyan, “Nonlinear minimum mean square error estimator for mixture-maximisation approximation,” *Electronics Letters*, vol. 42, no. 12, pp. 724–725, 2006.

[36] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. Roy. Statist. Soc.*, vol. B30, pp. 1–38, 1977.

[37] F. Pernkopf and D. Bouchaffra, “Genetic-based EM algorithm for learning Gaussian mixture models,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1344–1348, Aug 2005.

[38] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1991.

[39] R. Cowell, A. Dawid, S. Lauritzen, and D. Spiegelhalter, *Probabilistic networks and expert systems*. Springer Verlag, 1999.

[40] T. Minka, “Divergence measures and message passing,” Microsoft Research Cambridge, Tech. Rep. MSR-TR-2005-173, 2005.

[41] M. Jordan, *Learning in graphical models*. MIT Press, 1999.

[42] C. Huang and A. Darwiche, “Inference in belief networks: A procedural guide,” *Int. J. Approximate Reason.*, vol. 15, no. 3, pp. 225–263, 1996.

[43] S. Aji and R. McEliece, “The generalized distributive law,” *IEEE Trans. Information Theory*, vol. 46, no. 2, pp. 325–343, Mar 2000.

[44] Y. Weiss, “Belief propagation and revision in networks with loops,” Tech. Rep. AIM-1616, CBCL-155, 1997.

[45] P. Smaragdis, “Convolutive speech bases and their application to supervised speech separation,” *IEEE Trans. Audio Speech and Language Processing*, vol. 15, no. 1, pp. 1–12, Jan 2007.

Michael Wohlmayr graduated from Graz University of Technology (TUG) in June 2007. He conducted his Master thesis in collaboration with University of Crete, Greece. Since October 2007, he is pursuing the PhD degree at the signal processing and speech communication laboratory at the TUG. His research interests include Bayesian networks, speech and audio analysis, as well as statistical pattern recognition.

Michael Stark received his MSc (Dipl.-Ing.) degree in Electrical Engineering-Sound Engineering at Graz, University of Technology and University of Music and Performing Arts Graz, Austria, in summer 2005. In 2007 he did an internship at University of Crete, Greece. Currently he is pursuing his PhD degree at Graz, University of Technology, Austria. His research interest is in the area of speech processing with particular emphasize on source separation, speech detection, and quality assessment.

Franz Pernkopf received his MSc (Dipl. Ing.) degree in Electrical Engineering at Graz University of Technology, Austria, in summer 1999. He earned a PhD degree from the University of Leoben, Austria, in 2002. In 2002 he was awarded the Erwin Schrödinger Fellowship. He was a Research Associate in the Department of Electrical Engineering at the University of Washington, Seattle, from 2004 to 2006. Currently, he is assistant professor at the laboratory of signal processing and speech communication, Graz University of Technology, Austria. His research interests include machine learning, Bayesian networks, feature selection, finite mixture models, vision, speech, and statistical pattern recognition.