Source–Filter-Based Single-Channel Speech Separation Using Pitch Information

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

***Abstract*—In this paper, we investigate the source–filter-based approach for single-channel speech separation. We incorporate source-driven aspects by multi-pitch estimation in the model-driven method. For multi-pitch estimation, the factorial HMM is utilized. For modeling the vocal tract filters either vector quantization (VQ) or non-negative matrix factorization are considered. For both methods, the final combination of the source and filter model results in an utterance dependent model that finally enables speaker independent source separation. The contributions of the paper are the multi-pitch tracker, the gain estimation for the VQ based method which accounts for different mixing levels, and a fast approximation for the likelihood computation. Additionally, a linear relationship between pitch tracking performance and speech separation performance is shown.**

***Index Terms*—Single-channel speech separation (SCSS), multi-pitch estimation, source–filter representation.**

I. Introduction

THE aim of source separation is to divide an instantaneous linear mixture $x = s_1 + s_2$ of two signals into its underlying source signals s_1 and s_2. For single-channel speech separation (SCSS) two sound sources are mixed into a single channel. This is in general an ill-posed problem and cannot be solved without further knowledge about the sources or their interrelationship. SCSS can be mainly divided into the area of implicit models also known as computational auditory scene analysis (CASA) and explicit models known as underdetermined blind source separation methods [1].

Implicit models try to mimic the remarkable ability of the human auditory system to recover individual sound components in adverse environments. Here, the mixture is a scene to be organized and particular extracted components are merged to form output streams of individual sources. The CASA systems in [2] and [3] are the most important representatives. Both systems are heavily based on harmonicity as cue for separation. Wang *et al.* [4], [5] suggested the use of the ideal binary mask as computational goal for auditory scene analysis. The ideal binary mask uses the mixture maximization (*mixmax*) approach [6], i.e., the element-wise maximum operator applied on a time–frequency representation, i.e., the spectrogram, to separate the two signals.

In contrast, explicit models incorporate prior knowledge such that the individual source characteristics are learned in a generative manner during a training phase. This speaker-dependent model is used as source prior knowledge and is applied for separation without considering the interfering component. The two most prominent explicit models are the factorial-max vector quantization [7] (VQ) and the factorial-max hidden Markov model [8] which also integrates time dependencies. In both models, the most likely states at every time instance are selected in the *mixmax* sense, conditioned on the observed mixture. Another method capable of identifying components with temporal structure in a time–frequency representation is non-negative matrix factorization (NMF) [9], [10]. Here, the mixture is decomposed into a bases and a weight matrix. The weight matrix specifies the contribution of each basis to model the observation. The layered factorial HMM in [11] is currently the best performing method on the Pascal Speech Separation Challenge [12]. To model speaker characteristics, an acoustic model which is driven by a grammar model is used. Only for the grammar model a first-order Markov process is employed. However, in this work speech recognition of mixed signals is the main task.

In this paper, we use the source- and model-driven approach as already proposed by Radfar *et al.* [13]. They suggest to also consider the speech signal characteristics and use them as an additional cue for separation. Using this as a basis, the signal can be decomposed into a fine spectral structure related to the excitation signal and a coarse spectral structure representing the vocal tract information. The source-driven part extracts the fundamental frequency (f_0) or its perceived counter part, the pitch information of each speaker using a multi-pitch tracking method. Afterwards, the estimated pitch of each speaker is used to synthesize an artificial excitation signal representing the fine spectral structure. Utilizing this excitation signal, the vocal-tract filters (VTFs) are estimated based on a probabilistic model-driven approach. This decomposition results in a speaker independent (SI) system in contrast to most other methods, e.g., [7], [10], and [11]. For both, the multi-pitch estimation algorithm and the VTF estimation method we used the same time–frequency representation, i.e., the spectrogram.

An approach for robust multi-pitch tracking has been proposed in [14]. It is based on the unitary model of pitch perception

Manuscript received August 31, 2009; revised December 17, 2009; accepted February 25, 2010. Date of publication April 05, 2010; date of current version October 27, 2010. This research was carried out in the context of COAST-ROBUST, a joint project of Graz University of Technology, Nuance Communications International, and Sail Labs Technology. This work was supported by the Austrian KNet Program, in part by the ZID Zentrum for Innovation and Technology, Vienna, in part by the Steirische Wirtschaftsfoerderungsgesellschaft mbh, in part by the Land Steiermark, and in part by the Austria Science Fund under Project P19737-N15. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sharon Gannot.

The authors are with the Signal Processing and Speech Communication Laboratory, Graz University of Technology, Graz 8010, Austria (e-mail: michael.stark@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASL.2010.2047419

[15], upon which several improvements are introduced to yield a probabilistic representation of the periodicities in the signal. Semi-continuous pitch trajectories are then obtained by tracking these likelihoods using an 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. Recently, it was shown that factorial HMMs (FHMMs) [16] provide a natural framework to track the pitch of multiple speakers [17], [18].

In this paper, we go a step further and use Gaussian mixture models (GMMs) to model the spectrogram features of the speech mixture. For this purpose, we require *supervised* data, i.e., the pitch-pairs for the corresponding speech mixture spectrograms to learn the GMMs. These data are generated from single speaker recordings applying the *RAPT* pitch tracking method [19]. Learning the GMMs is combined with the minimum description length (MDL) [20] criterion to find the optimal number of Gaussian components for modeling the spectrograms belonging to a specific pitch-pair. This approach significantly outperforms two methods based on correlogram features. We report these results in [21].

For the coarse spectral structure, a speaker independent VTF model is trained. Therefore, we compare two statistical methods, one is based on VQ and the other one on NMF. Additionally, we propose a new gain estimation method for the VQ model addressing the problem of different mixing levels. Furthermore, we propose a computational efficient method for the likelihood estimation. This method is in spirit similar to beam search [22] used for efficient decoding in HMMs.

To evaluate these methods, we assess performance in various ways using the Grid Corpus [12]. First, results are reported for every single building block, i.e., the multi-pitch tracking algorithm, the gain estimation, and the likelihood approximation. Second, separation performance on the SCSS task is assessed extracting pitch information in a speaker-dependent (SD), gender-dependent (GD), and finally speaker-independent (SI) way. Moreover, we perform separation using reference single pitch trajectories taken from *RAPT*. Third, we present performance results using just the excitation signals for speech separation.

The remainder of this paper is structured as follows. In Section II, we introduce the general model for the source–filter-based SCSS. Section III presents the multi-pitch tracking algorithm. The proposed VTF models are characterized in Section IV. The experimental setup and results are discussed in Section V. Finally, we conclude and give future perspectives in Section VI.

II. Source- and Model-Driven Approach

In the source–filter model, the speech signal is composed of an excitation signal that is shaped by the vocal tract acting as a filter process. Hence, a speech segment s_i is the convolution of the excitation e_i with the VTF h_i which are further multiplied by a gain factor g_i in the time domain as

$$s_i = g_i\,(e_i \star h_i) \tag{1}$$

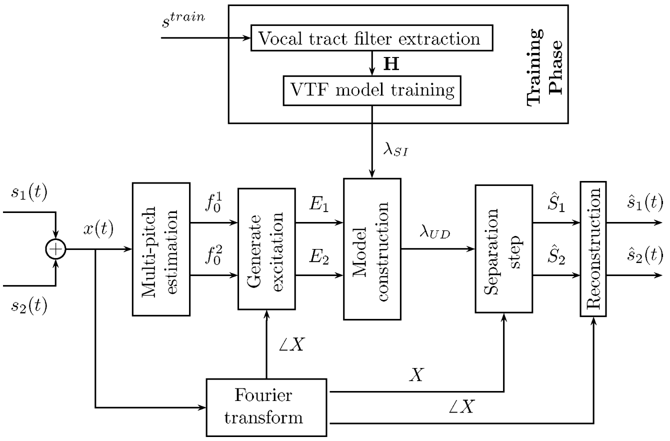


Fig. 1. Blockdiagram of the separation system.

where the speaker index is given as $i \in \{1,2\}$. The convolution results in a multiplicative relation in the frequency domain as

$$\breve{S}_i = g_i\,\breve{E}_i\,\breve{H}_i.$$

Generally, we denote signals in time domain in lower case, e.g., s_i and x, signals in the magnitude spectral domain by uppercase characters with a half-pipe, e.g., $\breve{S}_i$ and $\breve{X}$, and signals in the log-magnitude spectrum in uppercase only as $S_i = \log \breve{S}_i$ and $X = \log \breve{X}$ throughout the paper.

The overall SCSS system is shown in Fig. 1 and consists of the following building blocks: a multi-pitch tracking unit followed by the excitation generation unit is representing the source-driven part. In this paper, we compare SD, GD, and SI multi-pitch tracking performance and employ them for speech separation. Once the pitch trajectories of each speaker are estimated, i.e., f_0^1 and f_0^2, they are further utilized to create the excitation signals E_1 and E_2. Details about the source-driven part are described in Section III. VTFs, known as spectral envelopes, are extracted from SI training data s^{train} and are used to train SI models λ_{SI}, either a $\lambda_{\mathrm{SI}}^{\mathrm{VQ}}$ or a $\lambda_{\mathrm{SI}}^{\mathrm{NMF}}$ model (see Section IV). The combination of the excitation signal E_i and the VTF model which is carried out in the model combination block of Fig. 1, results in an utterance dependent (UD) model λ_{UD}, i.e., the VTFs in combination with the excitation are modeling a particular utterance. Thus, the harmonic excitation signal acts as discriminative feature and introduces utterance dependency which enables speech separation. The UD model is further used to separate the speech mixture in the separation step.

For performance analysis we can estimate the component signals in two ways.

- The most likely speech bases of each component speech signal are used to find the respective binary masks (BMs). Afterwards, the BM is used to filter the speech mixture X in order to get an estimate of the component signal $\hat{S}_i$.
- The estimated speech bases are directly used for synthesis of the component speech signals $\hat{S}_i$.

In the reconstruction block of Fig. 1, the separated speech signals are synthesized by first applying the inverse Fourier transform on each speech segment using the phase of the mixed speech signal $\angle X$. For speech signal reconstruction the overlap–add method is used.

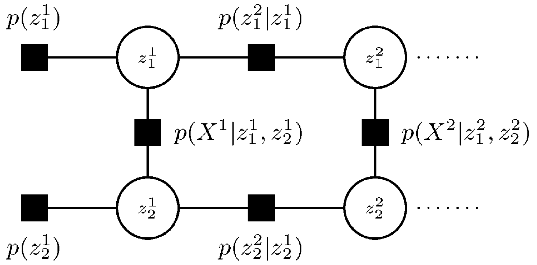


Fig. 2. Factorial HMM shown as a factor graph [23]. Factor nodes are shown as shaded rectangles together with their functional description. Hidden variable nodes are shown as circles. Observed variables X^t are absorbed into factor nodes.

In this paper, we use different models to represent certain speaker spaces. The SI space is characterized by one universal model valid for all speakers and phonemes they can articulate. The GD model is trained to represent the distribution unique for each gender, male or female. Further, the SD model describes the space of each individual speaker. A subset of the SD space is the utterance dependent space, i.e., an individual model per utterance. Hence, the SI space can be decomposed according to: UD $\subseteq$ SD $\subseteq$ GD $\subseteq$ SI.

III. Multi-Pitch Tracking Using FHMM

We use an FHMM for tracking the pitch trajectories of both speakers. The FHMM represented as factor graph [23] is shown in Fig. 2. The hidden state random variables are denoted by z_i^t, where $i \in \{1, 2\}$ indicates the Markov chain related to the speaker index and t the time index from 1 to T. Similarly, the observed random variables, i.e., the log magnitude spectrum, are denoted by X^t at time t. Each z_i^t represents a discrete random variable related to the pitch of speaker i at t, while X^t is continuous. For simplicity, all hidden variables are assumed to have cardinality $|z|$. 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 $p(z_i^t \,|\, z_i^{t-1})$. The dependency of the observed variables X^t on the hidden variables of the same time frame is defined by the observation probability $p(X^t \,|\, z_1^t, z_2^t)$. Finally, the prior distribution of the hidden variables in every chain is denoted by $p(z_i^1)$. Denoting the whole sequence of variables, i.e., $\{z^t\} = \bigcup_{t=1}^{T} \{z_1^t, z_2^t\}$ and $\{X^t\} = \bigcup_{t=1}^{T} \{X^t\}$, the joint distribution of all variables is given by

$$\begin{aligned} p(\{z^t\}, \{X^t\}) &\\ = p(\{X^t\} \,|\, \{z^t\}) p(\{z^t\}) &\\ = \prod_{i=1}^{2} \left[p\left(z_i^1\right) \prod_{t=2}^{T} p\left(z_i^t \middle| z_i^{t-1}\right) \right] \prod_{t=1}^{T} p\left(X^t \middle| z_1^t, z_2^t\right). \end{aligned}$$

The number of possible hidden states, i.e., per time frame is $|z|^2$. As pointed out in [16], 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 $|z|^2$ states would allow any $|z|^2 \times |z|^2$ transition matrix between two hidden states, the FHMM is restricted to two $|z| \times |z|$ transition matrices.

A. FHMM Parameters

The state-conditional observation likelihoods $p(X^t \,|\, z_1^t, z_2^t)$ are modeled with a GMM using $M \geq 1$ components according to

$$p\left(X^t \,|\, \boldsymbol{\Theta}_{z_1,z_2}\right) = \sum_{m=1}^{M} \alpha_{z_1,z_2}^m \mathcal{N}\left(X^t; \boldsymbol{\Theta}_{z_1,z_2}^m\right).$$

To obtain $X^t \in \mathbb{R}^{64}$, we first apply the zero padded 1024 point FFT on a Hamming windowed signal segment x^t of length 32 ms. Next, we take the log magnitude of spectral bins 2–65, which corresponds to a frequency range up to 1 kHz. This covers the most relevant frequency range of resolved harmonics while keeping the model complexity low. α_{z_1,z_2}^m corresponds to the weight of each component $m = 1, \ldots, M$. These weights are constrained to be positive $\alpha_{z_1,z_2}^m \geq 0$ and $\sum_{m=1}^{M} \alpha^m = 1$. The parameters $\boldsymbol{\Theta}_{z_1,z_2} = \{\alpha_{z_1,z_2}^m, \boldsymbol{\Theta}_{z_1,z_2}^m\}_{m=1}^{M}$ can be learned by the EM algorithm [24], where $\boldsymbol{\Theta}_{z_1,z_2}^m = \{\boldsymbol{\mu}_{z_1,z_2}^m, \boldsymbol{\Sigma}_{z_1,z_2}^m\}$.

Each hidden variable has $|z| = 200$ states, where state value "1" refers to "no pitch," and state values "2'–'200" correspond to different pitch frequencies ranging from less than 1 ms to 12.5 ms, i.e., 80 Hz-∼1 kHz. Note that segments of silence and unvoiced speech are modeled by $z = 1$. For learning the GMM we need *supervised* data, i.e., the pitch-pairs for the corresponding speech mixture spectrograms. These data are composed from single speaker recordings using the *RAPT* pitch extraction [19]. Hence, with both pitch trajectories for the mixed utterances at hand, we can easily learn a GMM $p(X^t \,|\, \boldsymbol{\Theta}_{z_1,z_2})$ for each pitch-pair (z_1, z_2). Accordingly, we have to determine 200×200 GMMs. Unfortunately, data might be rarely available for some pitch-pairs, whereas, there is plenty of data for, e.g., $(z_1 = 1, z_2 = 1)$. For this reason, we use MDL to determine the number of components of the GMM automatically. The MDL criterion [20] is

$$MDL = -\log p\left(\mathcal{X}_{z_1,z_2} \,|\, \boldsymbol{\Theta}_{z_1,z_2}\right) + \frac{M(L+1)}{2} \log N_{z_1,z_2}$$

where L is the number of parameters per component (for GMMs with diagonal covariance matrix $L = 2d$ where $d = 64$ in our case), in $\mathcal{X}_{z_1,z_2}$ all spectrogram samples belonging to (z_1, z_2) are collected, and N_{z_1,z_2} denotes the size of $\mathcal{X}_{z_1,z_2}$, i.e., $N_{z_1,z_2} = |\mathcal{X}_{z_1,z_2}|$. This equation has the intuitive interpretation that the log-likelihood $-\log p(\mathcal{X}_{z_1,z_2} \,|\, \boldsymbol{\Theta}_{z_1,z_2})$ is the code length of the *encoded* data. The term $(M(L+1))/(2) \log N_{z_1,z_2}$ models the optimal code length for all parameters $\boldsymbol{\Theta}_{z_1,z_2}$. In case of $N_{z_1,z_2} = 1$ for a particular (z_1, z_2) we use a single Gaussian with $\boldsymbol{\mu}_{z_1,z_2}^m = \mathcal{X}_{z_1,z_2}$ and $\boldsymbol{\Sigma}_{z_1,z_2}^m$ is set to a small $\sigma_{\min}\mathbf{I}$, where $\mathbf{I}$ is the identity matrix. For $N_{z_1,z_2} > 1$, we train GMMs with M ranging from 1 to 15, and take the GMM whose corresponding MDL criterion is minimal. If there is no training sample available for the pitch-pair (z_1, z_2), i.e., $N_{z_1,z_2} = 0$, we set $\boldsymbol{\mu}_{z_1,z_2}^m = 0$ and $\boldsymbol{\Sigma}_{z_1,z_2}^m = \mathbf{I}$. Prior to pitch tracking all spectrogram samples are normalized to zero mean and unit variance. Finally, we multiply the pitch likelihood $p(X^t \,|\, z_1^t, z_2^t)$ with the pitch-pair prior $p(z_1^t, z_2^t)$, since this slightly improved the performance in our experiments.

Both transition matrices of the FHMM $p(z_i^t \,|\, z_i^{t-1})$ are obtained by counting and normalizing the transitions of the pitch values from single speaker recordings. Additionally, we apply Laplace smoothing[1] on both transitions $p(z_i^t \,|\, z_i^{t-1})$. The prior distributions $p(z_i^1)$ are obtained in a similar manner.

B. Tracking

The task of tracking involves searching the sequence of hidden states $\{z^t\}^*$ that maximizes the conditional distribution $p(\{z^t\} \,|\, \{X^t\})$:

$$\{z^t\}^* = \arg\max_{\{z^t\}} p(\{z^t\} \,|\, \{X^t\}). \tag{2}$$

For HMMs, the exact solution to this problem is found by the Viterbi algorithm. For FHMMs, an exact solution can be found using the junction tree algorithm [25]; however, this approach gets intractable with increasing number of hidden Markov chains and $|z|$. Algorithms for approximate and exact solutions on FHMMs are derived in [16]. Approximate inference algorithms are often derived from the framework of variational inference. The sum–product algorithm [23] can be derived under a similar setting of variational principles [26], although more intuitive derivations exist for graphs without loops. When applied on a graph with loops, as is the case for a FHMM, the solutions are in general not guaranteed to converge and can only approximate the optimal solution.

In this paper, we explored the max-sum algorithm (a variant of sum-product algorithm) as well as the junction tree algorithm. We apply both variants on the *loopy* FHMM graph to obtain a solution for (2). In contrast to the junction tree algorithm, the max-sum algorithm can only approximate (2). In [18], experimental results suggested that the obtained solutions sufficiently approximate the exact solution, while computational complexity is much lower. Indeed, the time complexity of the max-sum algorithm applied to a FHMM is $\mathcal{O}(TK|z|^K)$, where K is the number of Markov chains. In contrast, the time complexity of the junction tree algorithm is $\mathcal{O}(TK|z|^{K+1})$.

In the sequel, we give a short overview of the used max-sum message passing algorithm. For a detailed discussion, we refer the interested reader to [23], [25], and [26]. Further, details on the junction tree algorithm are given in [16]. The max-sum algorithm is based on passing messages between nodes of a graph. Among various types of graphs, factor graphs [23] have become popular to depict the mechanisms of message passing. Fig. 2 shows a FHMM as factor graph, where the functional dependency of each variable node, for brevity called z, is made explicit by "factor nodes," shown as shaded rectangles, i.e., each rectangle denotes a function $f(\{\hat{z}\})$ of its adjacent (i.e., neighboring) variable nodes $\{\hat{z}\}$.

For the max-sum algorithm, each node sends to every neighbor a vector valued message μ, which is itself a function of the messages it received, (as well as $f(\{\hat{z}\})$, for the case of a factor node). A message from variable node z to factor node f is

$$\mu_{z\to f}(z) = \sum_{g\in n(z)\setminus f} \mu_{g\to z}(z) \tag{3}$$

while a message from factor f to variable z is

$$\mu_{f\to z}(z) = \max_{\{\hat{z}\}\setminus z}\left(\ln f(\{\hat{z}\}) + \sum_{y\in\{\hat{z}\}\setminus z} \mu_{y\to f}(y)\right). \tag{4}$$

Here, $n(x)$ denotes the set of neighbor nodes of x. We normalize each message and restrict each node to send a maximum of 15 messages per link. 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. After the last iteration, we obtain the maximum *a posteriori* configuration $p^*(z)$ of each variable node $z \in \{z^t\}$ as a function of its incoming messages according to

$$p^*(z) = \max_{\{z^t\}\setminus z} p(\{z^t\} \,|\, \{X^t\}) = \sum_{g\in n(z)} \mu_{g\to z}(z). \tag{5}$$

Although the set of maxima $z^* = \arg\max_z p^*(z)\ \forall z \in \{z^t\}$ does not necessarily yield the global maximum $\{z^t\}^*$, as multiple global maxima might be present, a backtracking stage may lead to inconsistencies due to the loops in the factor graph. For this reason, we simply set the global maximum $\{z^t\}^*$ to the set of individual maxima z^*.

C. Excitation Synthesis

Once the pitch tracks are estimated for each speaker, the harmonic part of the excitation signal is modeled as

$$e_i\left(t_f, \omega_0^i(t), \angle\,\check{X}(u)\right) = \sum_{u=1}^{U(\omega_0^i, f_{\max})} \sin\left(u\,\omega_0^i(t)\,t_f + \angle\check{X}(u)\right) \tag{6}$$

where U denotes the number of harmonics up to a specified highest frequency $f_{\max}$ set to 4 kHz, $\omega_0(t)$ is f_0 in radians of a particular time frame, $\angle\,\check{X}$ is the phase of the mixed signal, and $t_f = [1, \ldots, T_f]$ is the time index in a time frame. ω_0^i at t is the sampling frequency divided by z_i^t, i.e., $\omega_0^i = (2\pi f_s)/(z_i^t)$ for all $2 \le z_i^t \le 200$. For unvoiced and silent signals, i.e., $z_i^t = 1$, a Gaussian random signal is used as excitation and for voiced signals, a Gaussian random signal filtered by a high-pass with cutoff frequency at $f_{\max}$ is added to (6). This equation is similar to the harmonic plus noise model [27] but without amplitude weighting of the harmonics. In our case, this weighting is provided by the VTF estimation algorithm.

IV. Vocal Tract Filter Models

In this section, we propose two different statistical VTF models for speech separation. The first method is based on the maximum-likelihood (ML) estimation of the VQ codewords. Here, the mixture maximization (*mixmax*) approach [6] is used as combination operator to represent the speech mixture X. Moreover, we discuss a gain estimation method to make our separation approach suitable for different mixing levels. Finally, we restrict the search space of the VQ to the most promising codewords by applying beam search. Second, we use NMF to model the VTFs.

[1] 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 smooths the transition probabilities.

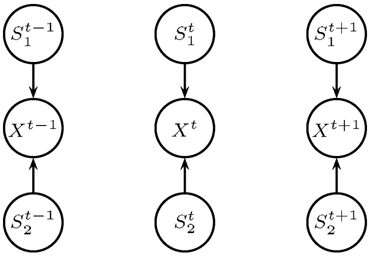


Fig. 3. Factorial-max VQ where X^t is represented by two state variables S_1^t and S_2^t.

A. Maximum-Likelihood-Based Source Separation

The speech mixture is approximated in the log magnitude–frequency domain by the conditional probability distribution model as

$$p(X \mid S_1, S_2) = \mathcal{N}(X; \max(S_1, S_2), \Sigma) \quad (7)$$

where X, S_1, and S_2 are the speech mixture and the respective underlying speech signals and $\mathcal{N}$ is the normal distribution. For the separation of the speech signals, we rely on the sparse nature of speech in its high-resolution time–frequency representation. Therefore, the *mixmax* operator is employed to combine the hidden variables S_1 and S_2 in order to represent the observation X. Hence, the log-spectrum of a mixed signal X can be approximated by the element-wise maximum of the log-spectrum of the component signals S_1 and S_2: $X \approx \max[S_1, S_2]$. This leads to the notion of the binary mask (BM). A "1" in the BM assigns the corresponding time–frequency cell to speaker one, whereas the cell is allocated to speaker two in case of a "0." The binary masks of two speakers are complementary, i.e., $\text{BM}_1 = \overline{\text{BM}}_2$. The observation model in (7), where two hidden variables explain an observation is called a factorial-max VQ model illustrated in Fig. 3.

Now, given the speaker dependent models and assuming that we have access to the state sequence chosen by the latent variables q_1 and q_2 associated with each speaker, the joint distribution of the observation and the underlying source signals for a particular instant of time is given as

$$p(X, S_1, S_2 \mid q_1, q_2) = p(X \mid S_1, S_2)\, p(S_1 \mid q_1)\, p(S_2 \mid q_2) \quad (8)$$

where q_i is associated with a particular basis S_i of speaker i. Finally, we require the posterior distribution for X given the unobserved hidden variables to model the dependency between the speech mixture X and the current states $q_1 \in Q$ and $q_2 \in Q$. This is achieved by marginalization over the underlying signal components as

$$p(X \mid q_1, q_2) = \int_{S_1} \int_{S_2} p(X \mid S_1, S_2)\, p(S_1 \mid q_1)\, p(S_2 \mid q_2)\, dS_1 dS_2. \quad (9)$$

The aim of source separation is to compute the observation likelihood $p(X \mid q_1, q_2)$ conditioned on the given state sequences. To form an estimate of the component signals the maximum *a posteriori* (MAP) estimate has to be computed. The MAP can be found by Bayes theorem using (9) as

$$p(q_1, q_2 \mid X) = \frac{p(X \mid q_1, q_2)\, p(q_1)\, p(q_2)}{p(X)} \quad (10)$$

where $p(q_1)$ and $p(q_2)$ are assumed to be independent prior distributions. Thus, the most likely states can be found by

$$\{q_1^\star, q_2^\star\} = \arg\max_{q_1, q_2}[p(q_1, q_2 \mid X)]. \quad (11)$$

Assuming uniform priors $p(q_1)$ and $p(q_2)$ and neglecting $p(X)$, $p(q_1, q_2 \mid X) \propto p(X \mid q_1, q_2)$ and we can further write $\{q_1^\star, q_2^\star\} = \arg\max_{q_1, q_2}[p(X \mid q_1, q_2)]$.

Besides, we assume that the density function conditioned on the states is deterministic and has the following property:

$$p(S_i \mid q_j) = \begin{cases} 1, & \text{if} \quad i = j \\ 0, & \text{otherwise.} \end{cases} \quad (12)$$

Introducing this assumption in (9) results in the following relation: $p(X \mid q_1, q_2) = p(X \mid S_1, S_2)$, where we represent S_i by the basis $\mu_{q_i}^s$ which has been drawn by the latent variable q_i. This relation directly enables the application of the VQ bases in (7) to separate the speech signals.

In the sequel, we introduce the source–filter-based approach for the above framework. To estimate the SI-VTF densities $p(H \mid q)$, the training data is split into Q mutually nonintersecting clusters. These densities are assumed to be Gaussian with a spherical covariance matrix. Hence, we use the k-means algorithm [28] to determine the cluster centers μ_q^h. Thus, the SI VTF model is composed of: $\lambda_{\text{SI}}^{\text{VQ}} = \bigcup_{q \in Q} \mu_q^h$. Each codebook entry of the VQ can be thought of representing a prototype VTF. Using this knowledge we can formulate an utterance dependent VQ by incorporating E_i and g_{q_i}, the gain factor depending on the basis q_i of speaker i as

$$\mu_{q_i}^s = \mu_q^h + \ln g_{q_i} + E_i \quad (13)$$

where E_i is provided by pitch extraction and the gain factor g_{q_i} is in detail introduced in Section IV-A1. Thus, the UD model $\lambda_{\text{UD}}^{\text{VQ}}$ can be found by

$$\lambda_{\text{UD}_i}^{\text{VQ}} = \bigcup_{q_i \in Q} \mu_{q_i}^s. \quad (14)$$

The UD models $\lambda_{\text{UD}_i}^{\text{VQ}}$ are employed for separation. Hence, the most likely bases $\mu_{q_1}^s$ and $\mu_{q_2}^s$ representing X are selected and used to find the BM or the synthesized signals.

To extract the vocal tract filters we use the SEEVOC method described in [29], where the gain information is implicitly included. Hence, for an equal mixing level of the two speech signals the gain factor can be excluded from the model in (13) or set to $g_{q_i} = 1$. For different mixing levels, however, the model does not match anymore and has to be adjusted.

Since VQ is prone to model the same VTF shapes at different gain levels with separate bases $\mu_{q_i}^h$, training data are mean normalized with the advantage of reducing model complexity and increasing robustness in learning the model. This results in a loss of the gain information which has to be recovered.

Recently, Kristjansson *et al.* [11] proposed to estimate the mixing level measured in the target-to-masker ratio (TMR) as defined in (21) for the whole speech utterance *a priori*. The speaker dependent models are then globally adjusted by the estimated TMR before separation. Therefore, the whole utterance must be available in advance. Hence, this method cannot be applied for online separation. Furthermore, in [11] a TMR is selected out of a discrete finite set, which also seems to be impractical. In contrast to their work, we propose to estimate the gain associated to each speaker for every speech segment separately. The gain estimation has the benefit to be applicable for online processing without restriction to a fixed discrete set.

1) Gain Estimation: Since the MAP approach can not account for bias mismatches and the gain has been removed before VTF training, we have to derive the gain for UD models. This gain estimation is also suitable for SD models. In general the gain normalized speech segment $\bar{S}$ and the speech segment with gain S have the following relation in the log-domain:

$$S = a + \bar{S} \rightarrow a = S - \bar{S} \tag{15}$$

where a is the gain vector containing the same value for each vector entry, i.e., $a = \text{const.} \equiv g \cdot \mathbf{E}$, where $\mathbf{E}$ is a vector with all components equal to one. Using the introduced model $\lambda_{\text{UD}_i}^{\text{VQ}}$, the normalized speech bases do not match exactly anymore, resulting in a gain vector containing different values. In order to tackle this problem, we have to estimate the gain for each speech segment. The Gaussian probability density model for one speech segment S_i conditioned on $\mu_{q_i}^s$ is given as

$$p\left(S_i \left| \mu_{q_i}^s \right.\right) = \frac{\exp\left(-\frac{\left(S_i - \mu_{q_i}^s\right)^2}{2\sigma^2}\right)}{\sqrt{2\pi}\,\sigma} \tag{16}$$

where $\mu_{q_i}^s$ is given in (13). This probability basically measures the similarity between the speech segment S_i and $\mu_{q_i}^s$. The probability density function for the mixture of (7) is given as

$$p\left(X \left| \mu_{q_1}^s, \mu_{q_2}^s \right.\right) = \frac{\exp\left(-\frac{\left(X - \max\left(\mu_{q_1}^s,\, \mu_{q_2}^s\right)\right)^2}{2\sigma^2}\right)}{\sqrt{2\pi}\,\sigma} \tag{17}$$

where we represent S_i by $\mu_{q_i}^s$. The task is to find the gain factors g_i such as to maximize the likelihood of observing X. We discovered that we can estimate each gain factor independently. In order to estimate the gain of a speaker's speech bases given the observed speech mixture X we adapt (15) to $a = X - (E_i + \mu_{q_i}^h)$ and perform quantile filtering [30] on the gain vector a. In contrast to the quantile filtering as defined in [30], where the filtering is performed over time we define the quantile filtering over frequency. Therefore, the gain vector is first sorted in ascending order

$$a(\rho_0) \leq a(\rho_1) \leq \cdots \leq a(\rho_D). \tag{18}$$

The estimate for the gain is obtained by taking the rth-quantile as $g = a(\lfloor rD \rceil)$, where $0 \leq r \leq 1$ and $\lfloor \cdot \rceil$ indicates the element-wise rounding operator. Taking the value $r = 0$ corresponds to the minimum in a and $r = 0.5$ to the median. For noise estimation the median is considered to be a robust estimator and therefore has been taken in our case. Fig. 4(a) and (b) compares the gain estimates (black solid line) to the true gains (blue dashed lines) for two female component speakers given the mixture over time. The amplitudes are normalized to the range between zero and one. For evaluation, the gain estimates are found given the normalized speech segment $\bar{S}_i$ of the respective speaker and the speech mixture X as observation.

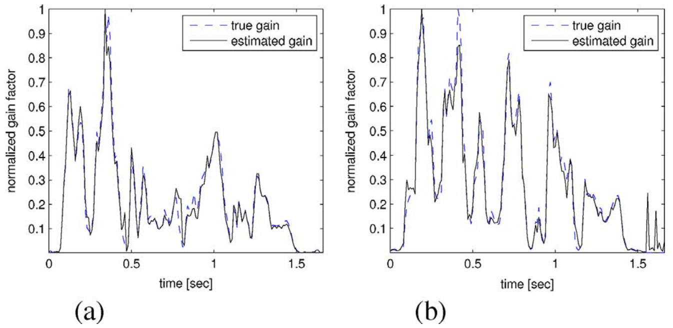


Fig. 4. Gain adjustment method. (a) and (b) show the true (blue dashed line) and estimated (black solid line) normalized gain for the component signals FE1 and FE2 observing just their mixture signal. (a) Speaker FE1. (b) Speaker FE2.

2) Efficient Likelihood Estimation: An efficient way to speed up the MAP estimation of (11) is to adapt beam search.

To make the beam search applicable for VQ we utilize the continuity property of speech, i.e., the energy in each frequency band changes slowly over time. We extracted the spectrum X^t with a time overlap of 50%—hence, at least half of the information contained in the current mixture X^t is also contained in X^{t+1}. Bearing in mind the above assumption, we can formulate the beam search for VQ. Therefore, we specify N, the number of surviving bases, i.e., the beam width. Furthermore, at step $t = 1$ using (11) we compute as initialization the full expectation and get $\{q_1^\star, q_2^\star\}$ the most likely state for each speaker model.

Given the most probable states, the most similar states at the next time step are selected for q_1 and q_2, computing the posterior as

$$p\left(q_1 \left| X^{t+1}, q_2^{t,\star} \right.\right) = \mathcal{N}\left(X^{t+1}, \max\left(\mu_{q_1}, \mu_{q_2}^{t,\star}\right), \Sigma\right) \tag{19}$$

where μ_{q_i} is the state mean of the random variable q representing speaker i. Here, we compute the likelihood of the first model being in state q_1 conditioned on the observation and the most likely state $q_2^\star$ of the second model and vice versa for the second model. Subsequently, we sort the likelihoods $p(q_1 \,|\, X^{t+1}, q_2^{t,\star})$ and $p(q_2 \,|\, X^{t+1}, q_1^{t,\star})$ in ascending order and specify a reduced set of $Q_i^{\prec}$ states containing the N best matching bases for each speaker used at $t + 1$ as

$$\hat{p}^{t+1}(S_i \,|\, q_i) = p(S_i \,|\, q_i^{\prec}), \quad i \in \{1, 2\} \tag{20}$$

where $Q_i^{\prec} \subseteq Q_i$ and $q_i^{\prec} \in Q_i^{\prec}$. This equation shows that (12) becomes dependent on time and that we only have ones where $q_i^{\prec} \in Q_i^{\prec}$. Hence, for time step $t + 1$ we only allow the N most likely states determined at time step t. Using the beam search procedure the computational complexity can be reduced from $O(T\,Q^2)$ to $O(Q^2 + (T - 1)\,N^2)$.

TABLE I
LABEL OF FEMALE AND MALE SPEAKERS USED FOR TRAINING SPEAKER INDEPENDENT MODELS

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

B. Separation Using Non-Negative Matrix Factorization

Furthermore, we have investigated NMF [9], [31] for VTF modeling. NMF approximates a non-negative matrix $V^{D\times T}$ by the product of two non-negative matrices $W^{D\times R}$ and $A^{R\times T}$, where D is the number of frequency bins and R is the approximation level, i.e., the number of bases. In our case, the VTF training data in the magnitude frequency domain corresponds to $V = \breve{S}_{\text{train}}$. The SI bases W are estimated and collected in $\lambda_{\text{SI}}^{\text{NMF}}$. The decomposition of V, in W and A is based on minimizing the Kullback–Leibler distance [9]. While during training the bases W are estimated, in the separation phase the weights A are of interest. These weights specify the contribution of each basis for the approximation of the speech mixture $\breve{X}$. Typically, in the separation step a union of all UD bases is constructed by combining them as $W_{\text{UD}} = W_{\text{UD}_1} \cup W_{\text{UD}_2}$. The UD bases W_{UD_i} can be constructed from W using the excitation $\breve{E}_i$ as

$$W_{\text{UD}_i} = W \cdot \breve{E}_i.$$

During separation, we fix the bases W_{UD} and estimate the weights A best approximating the mixed signal $\breve{X}$. Further, the reconstruction is done by first splitting the bases matrix W_{UD} as well as the estimated weight matrix A into the parts belonging to the corresponding sources. Finally, the reconstruction of the signals is given as

$$\hat{S}_i = W\ \breve{E}_i\ A_i$$

where $\hat{S}_i$ is the respective estimated spectrum of speaker i.

V. EXPERIMENTS

To evaluate the proposed separation algorithms, the Grid corpus recently provided by Cooke *et al.* [12] for the SCSS task has been selected. For both, source separation and multi-pitch tracking we assess performance using the true pitch tracks. The pitch tracks can only be extracted for the training corpus since for the test data only the speech mixtures are available. For this reason we use data from the training corpus for training and testing. For reference single pitch extraction we use *RAPT* [19], i.e., this is considered as ground truth. The sampling frequency was resampled to 16 kHz. For spectrogram calculation the signal was cut into segments of 32 ms with time shifts of 10 ms.

We use the spectral envelope estimation vocoder (SEEVOC) method described in [29] to extract the VTFs. For training SI models for both, pitch tracking and VTF modeling, we use ten male (MA) and ten female (FE) speakers each producing a maximum of 2 minutes of speech. The labels of the speakers are shown in Table I.

Two randomly selected male and female speakers, each uttering three sentences as shown in Table II were used for testing. For simplicity, we will call these speakers FE1, FE2, MA1, and MA2 in the sequel.

TABLE II
LABELS OF SPEAKERS AND FILE NAMES USED FOR TESTING

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

To evaluate the speech separation performance the target-to-masker ratio (TMR) has been used. To avoid synthesis distortions affecting the quality assessment, the TMR has been measured by comparing the magnitude spectrograms of the true source and the separated signal as

$$\text{TMR}_i = \frac{\sum_{d,t} \breve{S}_i^2(d,t)}{\sum_{d,t}(\breve{S}_i(d,t) - \hat{S}_i(d,t))^2} \tag{21}$$

where $d = [1,\ldots,D]$ is the frequency bin index and $\breve{S}_i$ and $\hat{S}_i$ are the source and separated signal spectra of the considered speaker i. All possible combinations between target speakers and their interfering speakers are evaluated, resulting in altogether 54 speech mixtures. Hence, 108 separated component signals are used for evaluation. For testing all files are mixed at equal level of 0-dB TMR. Audio examples of the mixtures and the separated files are available online.[2]

A. Multi-Pitch Tracking Results

In [21], we compared the performance of the proposed multipitch tracker to the well known approach in [14], and experimentally showed its superior performance on the Mocha-TIMIT database. In the following, we omit any comparisons to other algorithms, and report the performance of our approach on the Cooke database only.

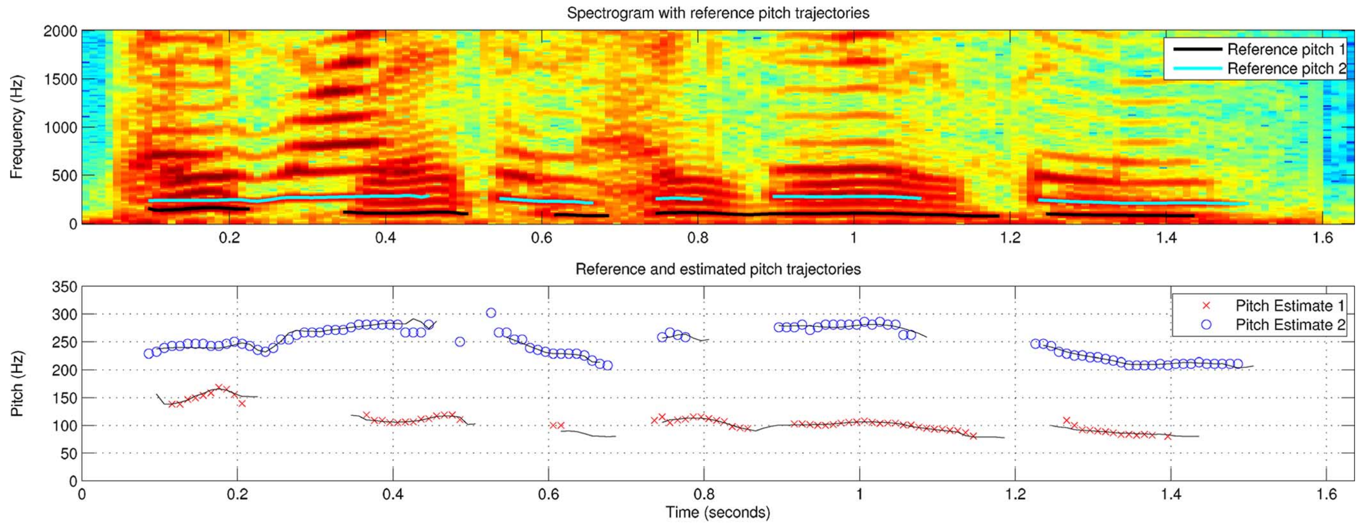
For every test mixture, the method estimates two pitch trajectories, $f_0^1[t]$ and $f_0^2[t]$. For performance evaluation, each of the two estimated pitch trajectories needs to be assigned to its ground truth trajectory, $\tilde{f}_0^1[t]$ or $\tilde{f}_0^2[t]$. From the two possible assignments, $(f_0^1 \rightarrow \tilde{f}_0^1, f_0^2 \rightarrow \tilde{f}_0^2)$ or $(f_0^1 \rightarrow \tilde{f}_0^2, f_0^2 \rightarrow \tilde{f}_0^1)$, the one is chosen for which the overall quadratic error is smaller. Note that this assignment is not done for each individual time frame, but for the global pitch trajectory.

To evaluate the resulting estimates, we use an error measure similarly to [14], however slightly modified to additionally measure the performance in terms of successful speaker assignment. E_{ij} denotes the percentage of time frames where i pitch points are misclassified as j pitch points, e.g., E_{12} means the percentage of frames with two pitch values estimated whereas only one pitch is present. The pitch frequency deviation is defined as

$$\Delta f_0^i[t] = \frac{\left| f_0^i[t] - \tilde{f}_0^i[t] \right|}{\tilde{f}_0^i[t]} \tag{22}$$

, where $\tilde{f}_0^i[t]$ denotes the reference chosen for $f_0^i[t]$. For each reference trajectory, we define the corresponding permutation

[2] https://www.spsc.tugraz.at/people/michael-stark/SCSS



E_{01}	E_{02}	E_{10}	E_{12}	E_{20}	E_{21}	E_{Gross}	E_{Fine}	E_{Total}
3.0675	0	7.3620	1.2270	0	15.9509	0	3.5844	31.1918

Fig. 5. Trajectories found by the proposed multi-pitch tracker, applied on speaker MA1 (“prwkzp”) and speaker FE1 (“lwixzs”) speaking simultaneously. The overall accuracy is high, yet some parts of the trajectory of speaker 1 cannot be tracked successfully. This leads to a high contribution of E_{21} and E_{10} to the overall error. The corresponding error measures on this test instance are shown in the Table at the bottom.

TABLE III
PERFORMANCE OF FHMM-BASED MULTI-PITCH TRACKING FOR SPEAKER DEPENDENT (SD) TRAINING. MEAN AND STANDARD DEVIATION (STD) OVER THE NINE TEST INSTANCES OF EACH SPEAKER PAIR ARE SHOWN

		E_{01}	E_{02}	E_{10}	E_{12}	E_{20}	E_{21}	E_{Gross}	E_{Fine}	E_{Perm}	E_{Total}
MA1-MA2	Mean	1.97	0.00	6.26	0.84	3.48	25.99	0.73	5.18	2.80	**47.24**
	Std	1.83	0.00	2.63	1.23	4.05	4.63	0.79	0.91	3.75	5.11
MA1-FE1	Mean	1.65	0.00	4.17	1.10	0.49	15.11	0.35	3.61	0.00	**26.48**
	Std	1.23	0.00	2.19	0.85	0.52	5.16	0.33	0.17	0.00	5.70
MA1-FE2	Mean	0.67	0.00	6.43	0.61	0.88	19.82	1.64	3.09	0.13	**33.27**
	Std	0.54	0.00	2.09	0.75	0.69	5.41	1.67	0.22	0.38	5.87
MA2-FE1	Mean	2.23	0.13	4.52	1.11	0.46	12.91	0.07	3.54	0.00	**24.97**
	Std	1.55	0.38	3.10	0.90	0.56	3.97	0.20	0.51	0.00	4.04
MA2-FE2	Mean	1.56	0.00	4.66	1.04	1.41	19.89	0.88	3.37	0.00	**32.82**
	Std	1.71	0.00	2.05	1.15	1.27	7.68	1.27	0.36	0.00	6.23
FE1-FE2	Mean	1.29	0.00	5.46	0.96	1.06	15.02	0.46	5.19	0.44	**29.88**
	Std	1.07	0.00	2.29	1.10	0.76	5.25	0.49	0.36	0.88	4.87

error $E^i_{\text{Perm}}[t]$ to be one at time frames where the voicing decision for both estimates is correct, but the pitch frequency deviation exceeds 20%, and $f^i_0[t]$ 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 E_{Perm} is the percentage of time frames where either $E^1_{\text{Perm}}[t]$ or $E^2_{\text{Perm}}[t]$ is one. Next, we define for each reference trajectory the corresponding gross error $E^i_{\text{Gross}}[t]$ 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. The overall gross error rate E_{Gross} is the percentage of time frames where either $E^1_{\text{Gross}}[t]$ or $E^2_{\text{Gross}}[t]$ is one. Finally, the fine detection error E^i_{Fine} is the average frequency deviation in percent at time frames where $\Delta f^i_0[t]$ is smaller than 20%. The overall error E_{Total} is defined as the sum of all error terms

$$E_{\text{Total}} = E_{01} + E_{02} + E_{10} + E_{12} + E_{20} + E_{21} + E_{\text{Gross}} + E_{\text{Fine}} + E_{\text{Perm}} \quad (23)$$

where $E_{\text{Fine}} = E^1_{\text{Fine}} + E^2_{\text{Fine}}$. For our SD models, we train each transition matrix used in the FHMM on reference pitch data from the corresponding speaker. Moreover, the GMM-based observation model is trained on mixtures of the two corresponding speakers. Similar to [18], experimental results for our SD models suggested that both tracking algorithms we studied—the junction tree algorithm and the max-sum algorithm—obtain solutions with equivalent E_{Total}. For this reason, we use the max-sum algorithm for tracking with SD models, as it is more efficient in terms of computational complexity. Table III shows the resulting error measure on the test set. To illustrate the performance and its corresponding error measure, we show an exemplary tracking result for the SD model in Fig. 5.

The GD observation models are trained on 3.3 hours of speech mixtures comprising ten male–male, male–female, or female–female speakers, respectively. The GD transition matrices are trained on reference pitch data of either male or female speakers. In contrast to the SD model case, we observed that the max-sum algorithm performs worse than the junction tree algorithm for GD models applied to same gender mixtures.

TABLE IV
PERFORMANCE OF FHMM-BASED MULTI-PITCH TRACKING FOR GENDER DEPENDENT (GD) TRAINING. MEAN AND STANDARD DEVIATION (STD) OVER THE NINE TEST INSTANCES OF EACH SPEAKER PAIR ARE SHOWN

		E_{01}	E_{02}	E_{10}	E_{12}	E_{20}	E_{21}	E_{Gross}	E_{Fine}	E_{Perm}	E_{Total}
MA1-MA2	Mean	3.89	0.00	6.84	3.84	1.95	21.59	1.89	8.78	5.80	**54.57**
	Std	2.05	0.00	3.26	3.13	2.30	5.70	1.20	2.57	6.37	12.03
MA1-FE1	Mean	3.96	0.00	4.87	3.29	0.77	18.98	1.26	3.87	1.95	**38.95**
	Std	1.86	0.00	2.82	3.49	0.83	3.68	1.11	0.71	1.53	6.38
MA1-FE2	Mean	2.53	0.00	4.93	3.61	1.25	17.64	1.42	3.76	1.45	**36.58**
	Std	1.86	0.00	2.16	1.96	1.51	4.54	1.10	0.64	2.30	8.79
MA2-FE1	Mean	4.05	0.07	2.75	2.42	0.33	14.39	6.08	4.31	2.46	**36.86**
	Std	2.04	0.20	1.76	1.63	0.42	6.05	5.93	0.89	2.72	8.74
MA2-FE2	Mean	2.38	0.00	4.06	3.01	0.52	14.43	2.09	3.90	0.65	**31.04**
	Std	1.63	0.00	2.19	1.85	0.46	4.48	2.65	0.65	0.93	4.56
FE1-FE2	Mean	3.10	0.00	3.56	4.47	0.40	9.90	1.51	6.40	10.18	**39.51**
	Std	2.12	0.00	1.47	2.07	0.58	3.50	1.28	3.16	4.68	6.40

TABLE V
PERFORMANCE OF FHMM-BASED MULTI-PITCH TRACKING FOR SPEAKER INDEPENDENT (SI) TRAINING. MEAN AND STANDARD DEVIATION (STD) OVER THE NINE TEST INSTANCES OF EACH SPEAKER PAIR ARE SHOWN

		E_{01}	E_{02}	E_{10}	E_{12}	E_{20}	E_{21}	E_{Gross}	E_{Fine}	E_{Perm}	E_{Total}
MA1-MA2	Mean	4.33	0.13	5.39	5.35	0.97	22.09	2.56	7.53	7.93	**56.28**
	Std	1.45	0.38	2.82	3.83	1.03	3.49	1.25	0.87	6.99	11.07
MA1-FE1	Mean	4.10	0.07	4.78	4.95	0.90	17.92	2.72	4.62	17.64	**57.69**
	Std	0.90	0.21	3.20	3.87	0.71	5.58	1.32	1.17	8.63	13.96
MA1-FE2	Mean	2.92	0.00	4.86	4.97	0.78	17.11	2.39	4.13	24.69	**61.83**
	Std	1.78	0.00	2.78	3.96	1.43	4.38	1.51	0.96	16.33	17.92
MA2-FE1	Mean	3.85	0.00	2.42	4.04	0.46	15.10	6.06	3.76	20.95	**56.65**
	Std	2.17	0.00	1.47	2.62	0.76	5.11	4.01	1.34	9.62	11.15
MA2-FE2	Mean	2.89	0.00	3.75	3.68	0.70	12.82	5.20	4.17	19.03	**52.25**
	Std	2.25	0.00	1.71	2.09	0.68	4.60	2.62	1.13	9.29	11.64
FE1-FE2	Mean	4.08	0.00	2.34	2.00	0.29	12.06	1.76	7.14	8.64	**38.31**
	Std	2.20	0.00	0.84	1.95	0.66	3.57	2.23	4.35	4.91	8.10

In that case, 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., $p(\boldsymbol{y}^{(t)} \,|\, x_1, x_2) = p(\boldsymbol{y}^{(t)} \,|\, x_2, x_1)$. For this reason, we apply the junction tree algorithm for tracking with GD models. Table IV gives the performance results for this model.

Finally, SI models are trained on 6.5 hours of speech mixtures composed of any combination of the ten male and ten female speakers. The transition matrix is trained on reference pitch data from both male and female speakers. For the same reason as for GD models, we use the junction tree algorithm for tracking with SI models. Table V shows the performance results.

The careful reader will notice that for the SI model, as well as for the male–male and female–female GD model, both Markov chains of the FHMM have the same transition matrix. In this case, the FHMM only allows symmetric solutions, i.e., identical pitch trajectories. To prevent this, we add a small amount of noise to create two slightly different transition matrices, for each Markov chain. This heuristic breaks the symmetry in the FHMM and allows individual trajectories for both speakers.

B. Gain Estimation Results

In this section, we assess performance of the gain estimation described in Section IV-A1. Therefore, two speech signals are mixed at TMR levels of 0, 3, 6, and 9 dB. Afterwards all signals are transformed to the log-magnitude domain and each component signal segment, i.e., S_1 and S_2, is normalized such that the maximum frequency component has 0 dB. For every speech segment the gain is estimated using the observed mixed signal X and the normalized log-magnitude spectrum of the speech segment $\bar{S}_i$. The signal is recovered by weighting the normalized signal segment with the estimated gain. The performance is measured utilizing the TMR for both the target (t) and the masker (m) speech signal as shown in Table VI. We observe that the gain can be recovered quite well for all three cases, namely same gender female (SGF), same gender male (SGM), and different gender (DG). Especially the TMR improvement for the 9-dB mixing case has to be emphasized. The masker-to-target ratio is -9 dB for the masker; thus, this method can increase the TMR by at least 19 dB for all cases. Without gain normalization, we measure an TMR of, e.g., 1.58 dB for the target and 2.55 dB for the masker speaker for the SGF case mixed at equal level.

TABLE VI
GAIN ESTIMATION PERFORMANCE FOR FOUR DIFFERENT MIXING LEVELS. RESULTS ARE MEASURED IN TMR TO THE ORIGINAL SPEECH FILE

TMR	SGF		SGM		DG	
[dB]	t	m	t	m	t	m
0	17.1	19.1	15.7	16.1	19.3	18.2
3	17.5	19.1	15.9	12.9	19.6	16.1
6	18.6	16.5	16	11	19.4	13.7
9	20.8	13.6	16.3	8.6	19.3	10.5

C. Efficient Likelihood Estimation Results

In order to speed up the likelihood estimation we introduced the beam search in Section IV-A2 for statistical models without memory, i.e., GMMs or VQ codebooks. This section summarizes the computational complexity of the beam search (BS) as a suboptimal search heuristic and compares results to the full

TABLE VII
COMPLEXITY COMPARISON FOR VQ USING FULL SEARCH (FS), FAST LIKELIHOOD ESTIMATION (FLE), AND BEAM SEARCH (BS)

Method	Comp. Complexity	♯ of Evaluations
FS	$\mathcal{O}(T\ Q^2)$	2.62e7
FLE	$\mathcal{O}(T\ (k^2\ + K^2))$	1.058e5
BS	$\mathcal{O}(Q^2 + (T-1)\ N^2)$	5.096e5

TABLE VIII
SEPARATION RESULTS IN TMR FOR DIFFERENT LIKELIHOOD ESTIMATION METHODS

Method		SGF	SGM	DG
FS	Mean	9.31	5.11	8.06
	Std	3.04	1.43	2.45
FLE	Mean	6.62	4.19	6.55
	Std	2.77	1.39	2.14
BS	Mean	9.49	4.41	7.51
	Std	2.64	1.39	2.03

search (FS) and the fast likelihood estimation (FLE) method as proposed in [11]. For the experiments we used VQ as statistical model to capture speaker dependent characteristics. Each speaker dependent VQ contains $Q = 512$ bases, i.e., $\mu^s_{q_i}$, trained on the log-magnitude spectrum. Hence, the training data was quantized into 512 cells. For separation we used the MAP estimate as defined in (11). For the BS method, a beam width of $N = 50$ has been selected. For the FLE method which employs a hierarchical structure, for the top layer as well as for the bottom layer $k = K = 23$ bases have been used. For convenience, we assume to have $T = 100$ speech frames which corresponds to 1 second of speech for a frame rate of 10 ms. The computational complexity for each search method is summarized in Table VII.

The complexity for both suboptimal methods can be reduced by two orders of magnitude. A comparison of the likelihood estimation methods in terms of TMR with mean and standard deviation is summarized in Table VIII.

For the given model size Q, the results of the full search are the upper bound for all three cases, i.e., SGF, SGM, and DG. Interestingly, the BS method shows a slightly higher TMR as the FS for the SGF case. We believe that this is due to the continuity assumption we employed for the complexity reduction of the BS. However, this TMR difference is not significant. Moreover, the proposed BS for all three cases has a superior performance compared to the FLE for the specified setting.

D. Speech Separation Results

All modules discussed so far are used to build the source separation algorithm. For both, $\lambda^{\text{VQ}}_{\text{SI}}$ and $\lambda^{\text{NMF}}_{\text{SI}}$, we trained models with 500 bases, respectively. The dimension of the bases corresponds to the number of frequency bins used in the spectrogram, i.e., 512. For training we used 200 iterations for NMF and 150 iterations for VQ. For VQ we perform experiments with and without gain normalized VTF models.

We conducted different experiments with focus on various parts of the system presented in the following.

1) Source separation experiments are carried out using reference $\tilde{f}^i_0$ trajectories for each speaker. The extraction is done on the single speaker utterances using *RAPT* [19], before mixing and will be called the supervised mode. This is the upper bound on performance we currently can achieve using our method.
2) SD trained models for multi-pitch f^i_0 estimation are utilized to separate the speakers. This method is already unsupervised but presumes to know the speaker identities in advance to select the adequate SD models.
3) A GD multi-pitch tracker has been explored to separate the speech mixture.
4) No prior knowledge is assumed anymore and speaker independent models for both the f_0 estimation and the VTF estimation are employed for separation.

Note, for all four experiments the same SI VTF model is used. For each of the four different pitch extraction methods enumerated above we compared four separation approaches namely, *Exci*, *NMF*, *GE-ML*, and *ML*, explained in the following:

- *Exci*: Here, we only use the excitation signals created from the f_0 trajectories by (6) for separation. Therefore, binary mask signals are derived based on the excitation signals and the speech signals are finally recovered by filtering the speech mixture with the respective BMs.
- *NMF*: We apply NMF for modeling the VTF. Utterance dependent bases are found by the combination of the SI learned VTF bases with the excitation signal.
- *GE-ML*: The VQ approach is used to separate the speech mixture. The speaker dependent model is formed by $\lambda^{\text{VQ}}_{\text{UD}}$ using gain estimation. Here, the data have been gain normalized to train the SI VTF model.
- *ML*: The VQ approach without gain estimation is used to separate the speech mixtures. Here, the gain information has not been removed from the data during training of the SI VTF model $\lambda^{\text{VQ}}_{\text{SI}}$. For separation the gain factor has been set to $g = 1$ in (13).

We report results for both, the estimated component signals $\hat{s}_i$ extracted by applying the respective BM on the speech mixture and the synthesis from the estimated speech bases. The synthesized signals naturally have a lower quality compared to the signals directly extracted using the BMs. Nevertheless, the results are rather instructive. A preliminary listening test indicated a subjectively better intelligibility of the synthesized signals compared to the BM signals for some utterances.

In all figures, the achieved mean value is depicted with a red horizontal line. The methods are identified by the label on the x-axis. Moreover, the standard deviation of the TMR is indicated by the blue box surrounding the red line. All experiments are split into three classes: SGF, SGM, and DG class.

First, performance of the supervised method using the f_0 extracted by *RAPT* [19] on the single speech utterances are presented. The results for synthesized signals are depicted in Fig. 6. Those signals are used to estimate the BM for each speaker. Further, the BMs are applied to the speech mixture in order to recover the signals. The BM results are shown in Fig. 7.

The performance of *Exci* emphasizes the importance of the fine spectral structure of speech which is a major cue for speech separation. This is well known from CASA [1]. Additionally incorporating the VTF models for separation improves the results in most of the cases. For the ML based method without gain estimation (GE) the results are getting slightly worse. Surprisingly, for the SGM case the usage of the VTF information does not im-

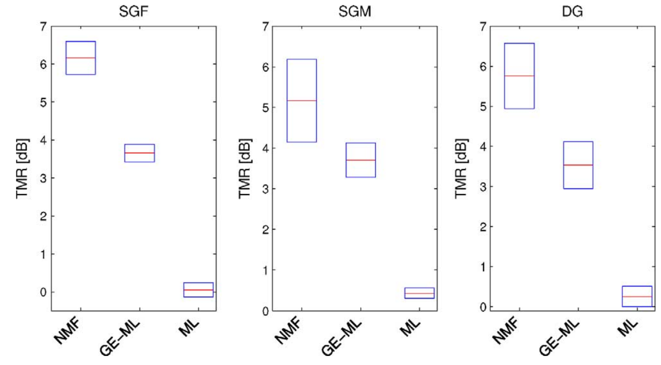


Fig. 6. Mean and standard deviation of the TMR for the synthesized signals using pitch trajectories extracted by *RAPT* [19].

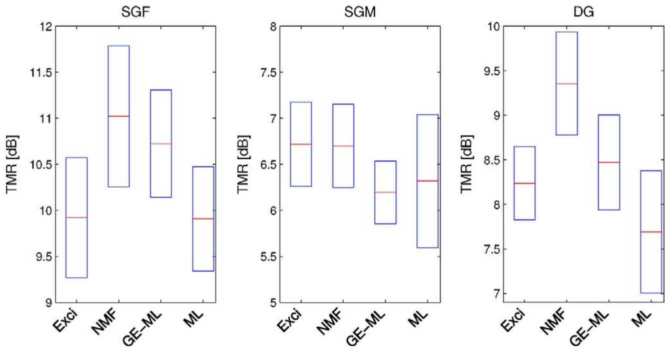


Fig. 7. Mean and standard deviation of the TMR for the BM signals using pitch trajectories extracted by *RAPT* [19].

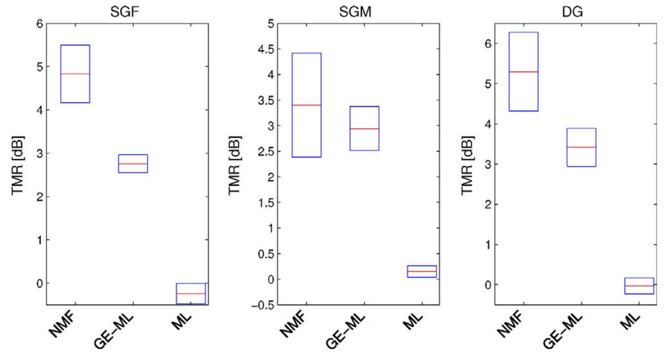


Fig. 8. Mean and standard deviation of the TMR for the synthesized signals using SD multi-pitch trajectories.

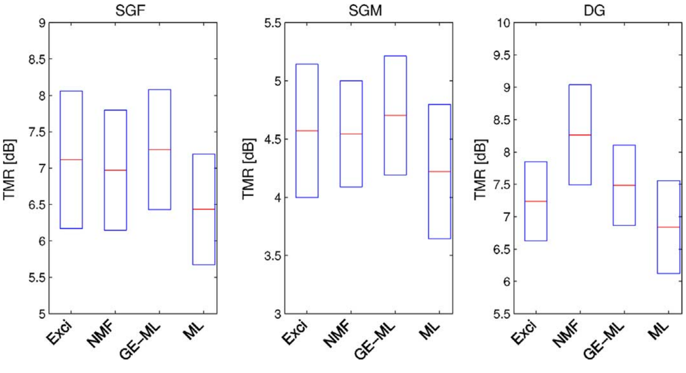


Fig. 9. Mean and standard deviation of the TMR for the BM signals using SD multi-pitch trajectories.

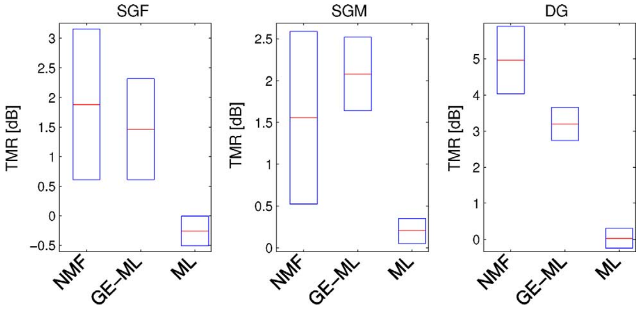


Fig. 10. Mean and standard deviation of the TMR for the synthesized signals using GD multi-pitch trajectories.

prove performance at all. We conjecture that the harmonics are rather close to each other and thus are acting as spikes which recover already the main speaker specific energy.

Next, the same separation methods are used with SD multi-pitch trajectories to create the excitation signal. In Fig. 8, the results for the synthesized signals are depicted and Fig. 9 shows results for the BM signals. As already noted in the above discussion the separation performance strongly depends on the used fundamental frequency. In our model, the f_0 information introduces at last utterance dependency. Thus, separation performance strongly correlates with the f_0 performance. Nonetheless, separation results are consistent. The GE-ML method only shows a slightly better performance compared to the excitation *Exci* signal for all cases. Moreover, for the SGM case about the same performance for all methods except the ML method can be reported using the BM. Similarities can be drawn to CASA where the separation is carried out in two steps: simultaneous grouping and sequential grouping. In our system, simultaneous grouping is executed during separation and sequential grouping is treated during multi-pitch tracking. In this respect, the sequential grouping is measured by E_{Perm}. For the SD case, Table III shows that a permutation error rarely occurs. For different gender mixtures on average only 0.03% and for same gender mixtures on average 1.62% of the speech frames are permuted.

As an intermediate step towards SI SCSS, gender-dependent multi-pitch tracking models to estimate f_0 trajectories are applied. Fig. 10 shows the results for the synthesized and Fig. 11 for the BM signals. Here, the same transitions are employed to estimate the pitch trajectories for the SG cases. Only for the DG case different transitions are taken which results in a more accurate pitch estimation and consequently in a better separation performance. Moreover, the permutation error for same and different gender mixtures occur on average in 7.99% and 1.63% of the speech frames, respectively. Both errors are coherent with the separation results.

Finally, SI extracted f_0 trajectories are employed for speech separation. This case is a fully SI SCSS method. Again Figs. 12 and 13 show the results for the synthesized and the BM extracted signals, respectively.

For the SI results, the GE-ML method performs slightly better using the synthesized signals. Nonetheless, for the found BM signals we cannot report large differences among the methods. The synthesized signals of the ML method show a rather poor performance. For different gender mixtures, E_{Perm} increases to 20.58% on average. In contrast, for same gender mixtures E_{Perm} is on average 8.28%. This is about the same E_{Perm} as

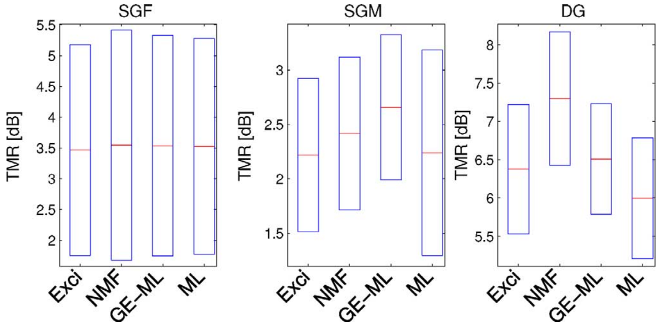


Fig. 11. Mean and standard deviation of the TMR for the BM signals using GD multi-pitch trajectories.

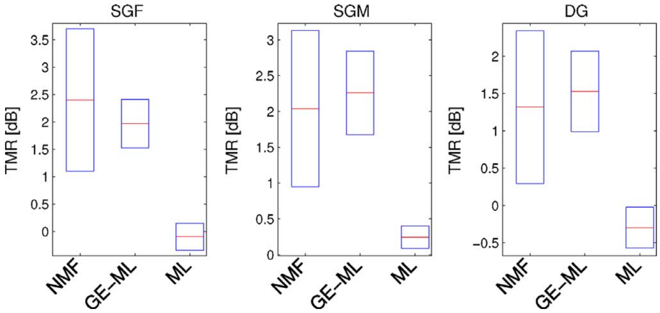


Fig. 12. Mean and standard deviation of the TMR for the synthesized signals using SI multi-pitch trajectories.

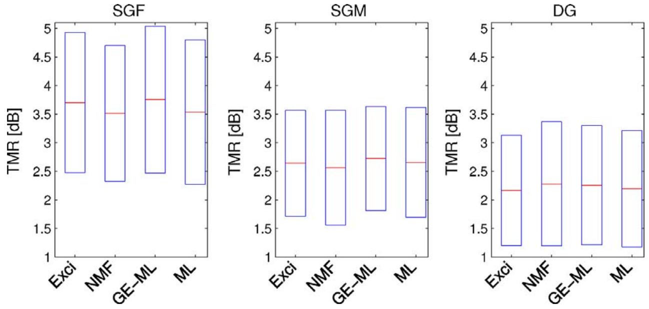
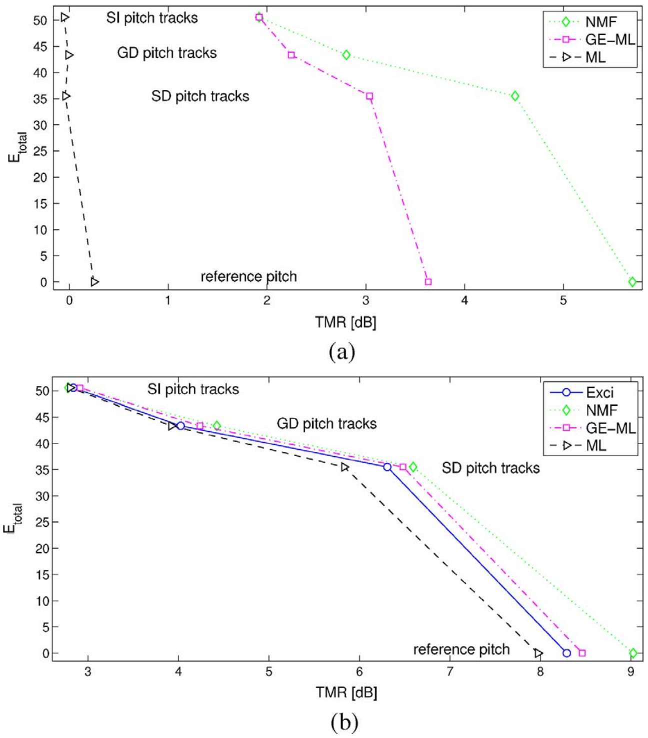


Fig. 13. Mean and standard deviation of the TMR for the BM signals using SI multi-pitch trajectories.



(a)

(b)

Fig. 14. Coherence between the average TMR versus average E_{Total} for all VTF and pitch estimation methods for the reference, SD, GD, and SI pitch tracks. Results are separately plotted for (a) synthesized speech signals. (b) BM signals. (a) TMR versus E_{Total}: Synthesized signals. (b) TMR versus E_{Total}: BM signals.

for the GD models. Thus, for different gender mixtures sequential grouping is a problem which is reflected by the significant contribution of E_{Perm} to E_{Total}. This also limits the source separation performance. This issue can be mitigated by postprocessing, e.g., Shao *et al.* [32] recently proposed a clustering approach to perform sequential grouping. In summary, we can report an almost linear relation between the separation results and the multi-pitch estimation performance when moving from the supervised to the SD- and finally to SI-based pitch estimation. This is shown in Fig. 14(a) and (b) which presents the coherence between the TMR for all introduced speech separation methods and the E_{Total} of the pitch tracker. Results are separately depicted for the reference, SD, GD, and SI multi-pitch trajectories. If E_{Total} is increasing, the TMR is decreasing no matter which method is selected for separation. As already observed in Figs. 6, 8, 10, and 12 the synthesized ML signals are not suitable to make them directly audible independent of the pitch estimation method. Moreover, it can be seen from Fig. 14(b), that the NMF and GE-ML methods show almost the same performance averaged over all pitch extraction models. The ML method degrades the TMR performance compared to just using the BM extracted from the excitation (Exci) signals only. It should be noted that the phonetic content of some utterances was almost the same with only one different word in the sentence (see Table II). In a nutshell, a comparison of all proposed VTF models slightly favors the NMF.

The computational complexity of each module has been addressed in the previous sections. The overall complexity of the system is the cumulation of these complexities. The average length of the speech mixtures is 1.69 [sec]. We compare this time to the average time the system takes to separate an utterance. Therefore, we measure the average time of each system module: The multi-pitch observation likelihood computation and tracking takes on average 862 and 18 [sec], respectively. However, note that the likelihood computation amounts to the evaluation of a set of GMMs, which can be computed in parallel to a high degree. In our evaluation, only sequential computations were performed. The VTF observation likelihood calculation using the BS method takes on average 4.4 [sec]. To separate one speech file of average length 1.69 [sec] the system takes approximately 884.4 seconds. Hence, 97.5% of the processing time is currently used for the observation likelihood computation during pitch tracking. All experiments have been performed using MATLAB on an Intel CPU CORE-i7 QUAD 920 running on 2.66 GHz. However, computational costs can be further reduced by approximations [33], [34].

VI. Conclusion and Outlook

In this paper, we presented a fully probabilistic approach for source–filter-based single channel speech separation (SCSS). A multi-pitch estimation algorithm has represented the source-driven part followed by an excitation modeling method. For multi-pitch extraction we used the factorial hidden Markov model. The filter-driven part is based on a speaker independent statistical model. In particular, two models either VQ or NMF are compared for vocal tract filter (VTF) estimation. NMF slightly outperforms VQ based VTF models. Utterance dependency was achieved by the combination of the source and filter models which finally enabled speech separation. For VQ we proposed a segment-based gain estimation which accounts for arbitrary mixing levels. In contrast to the utterance-based gain estimation, the proposed method can be used for online-processing. Additionally, we introduced beam search for VQ to approximate the likelihood efficiently. We report performance for every module of the system separately on the Grid corpus [12]. Multi-pitch estimation has been performed in speaker dependent, gender dependent, and finally in a speaker independent manner. For multi-pitch tracking we introduced a permutation error measure which accounts for wrong speaker assignments of pitch estimates. We compared our SCSS results to the separation results using the reference pitch trajectories and showed the relationship between pitch tracking and source separation performance. We achieve a TMR of 7, 4.5, and 3 dB for speaker-dependent, gender-dependent, and speaker-independent models, respectively.

In the future, we aim to carry out listening tests. Moreover, we plan to unite the source and filter parts into one model. Furthermore, we aim to investigate approaches to split the symmetry of the observation likelihood of the multi-pitch tracking method in order to improve the moderate performance of speaker independent source separation.

References

[1] *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*, ser. IEEE Press, D. Wang and G. J. Brown, Eds.. New York: Wiley, 2006.

[2] G. Hu and D. Wang, “Monaural speech segregation based on pitch tracking and amplitude modulation,” *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1135–1150, Sep. 2004.

[3] D. L. Wang and G. J. Brown, “Separation of speech from interfering sounds based on oscillatory correlation,” *IEEE Trans. Neural Netw.*, vol. 10, no. 3, pp. 684–697, May 1999.

[4] G. Hu and D. Wang, “Speech segregation based on pitch tracking and amplitude modulation,” in *Proc. IEEE Workshop Applicat. Signal Process. to Audio Acoust.*, 2001, pp. 79–82.

[5] D. Wang, “On ideal binary mask as the computational goal of auditory scene analysis,” in *Speech Separation by Humans and Machines*, 1st ed. New York: Springer, Nov. 2005, p. 319.

[6] A. Nadas, D. Nahamoo, and M. A. Picheny, “Speech recognition using noise-adaptive prototypes,” *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, no. 10, pp. 1495–1503, Oct. 1989.

[7] S. T. Roweis, “Factorial models and refiltering for speech separation and denoising,” in *Proc. Eurospeech*, Sep. 2003, pp. 1009–1012.

[8] S. T. Roweis, “One microphone source separation,” *Neural Inf. Process. Syst.*, vol. 13, pp. 793–799, 2000.

[9] D. D. Lee and H. S. Seung, “Learning the parts of objects by nonnegative matrix factorization,” *Nature*, vol. 401, p. 788, 1999.

[10] P. Smaragdis and J. Brown, “Non-negative matrix factorization for polyphonic music transcription,” in *Proc. IEEE Workshop Applicat. Signal Process. to Audio Acoust.*, 2003, pp. 177–180.

[11] T. Kristjansson, J. Hershey, P. Olsen, S. Rennie, and R. Gopinath, “Super-human multi-talker speech recognition: The IBM 2006 speech separation challenge system,” in *Proc. Interspeech*, 2006, no. 1775.

[12] M. P. Cooke, J. Barker, S. P. Cunningham, and X. Shao, “An audiovisual corpus for speech perception and automatic speech recognition,” *J. Acoust. Soc. Amer.*, vol. 120, no. 5, pp. 2421–2424, 2006.

[13] M. H. Radfar, R. M. Dansereau, and A. Sayadiyan, “A maximum likelihood estimation of vocal-tract-related filter characteristics for single channel speech separation,” *J. Audio, Speech, Music Process.*, vol. 1, p. 15, 2007.

[14] M. Wu, D. Wang, and G. Brown, “A multipitch tracking algorithm for noisy speech,” *IEEE Trans. Speech Audio Process*, vol. 11, no. 3, pp. 229–241, Mar. 2003.

[15] R. Meddis and L. O’Mard, “A unitary model of pitch perception,” *J. Acoust Soc. Amer.*, vol. 102, no. 3, pp. 1811–1820, 1997.

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

[17] F. Bach and M. Jordan, “Discriminative training of hidden Markov models for multiple pitch tracking,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2005, pp. 489–492.

[18] M. Wohlmayr and F. Pernkopf, “Multipitch tracking using a factorial hidden Markov model,” in *Proc. Interspeech*, 2008.

[19] D. Talkin, “A robust algorithm for pitch tracking,” in *Speech Coding and Synthesis.*. Amsterdam, The Netherlands: Elsevier, 1995, pp. 495–518.

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

[21] M. Wohlmayr and F. Pernkopf, “Finite mixture spectrogram modeling for multipitch tracking using a factorial hidden Markov model,” in *Proc. Interspeech*, 2009.

[22] F. Jelinek, *Statistical Methods for Speech Recognition.*. Cambridge, MA: MIT Press, 1998.

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

[24] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood estimation from incomplete data via the EM algorithm,” *J. R. Statist. Soc.*, vol. B39, no. B, pp. 1–38, 1977.

[25] M. Jordan, *Learning in Graphical Models.*. Cambridge, MA: MIT Press, 1999.

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

[27] J. Laroche, Y. Stylianou, and E. Moulines, “HNS: Speech modification based on a harmonic+noise model,” in *IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 27–30, 1993, vol. 2, pp. 550–553, IEEE.

[28] P. Vary and R. Martin, *Digital Speech Transmission, Enhancement, Coding and Error Concealment.*. New York: Wiley, Mar. 2006.

[29] R. McAulay and T. Quatieri, *Speech Coding and Synthesis.*. Berlin, Germany: Elsevier, 1995, ch. 4, pp. 121–173, Sinusoidal Coding.

[30] P. C. Loizou, *Speech Enhancement: Theory and Practice*, 1st ed. Boca Raton, FL: CRC, 2007.

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

[32] Y. Shao and D. Wang, “Sequential organization of speech in computational auditory scene analysis,” *Speech Commun.*, vol. 51, no. 8, pp. 657–667, Aug. 2009.

[33] E. Bocchieri, “Vector quantization for the efficient computation of continuous density likelihoods,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1993, vol. 2, pp. 692–695.

[34] M. Stark and F. Pernkopf, “On optimizing the computational complexity for VQ-based single channel source separation,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Dallas, TX, 2010, pp. 237–240.

Michael Stark (S’07) received the M.Sc. (Dipl.-Ing.) degree in electrical engineering–sound engineering from Graz, University of Technology and University of Music and Performing Arts, Graz, Austria, in 2005. He is currently pursuing the Ph.D. degree at the Signal Processing and Speech Communication Laboratory, Graz, University of Technology.

In 2007, he did an internship at the University of Crete, Heraklion, Greece. His research interest is in the area of speech processing with particular emphasize on source separation, speech detection, and quality assessment.

Michael Wohlmayr (S'09) received the M.S. degree from the Graz University of Technology (TUG), Graz, Austria, in June 2007. He conducted his M.S. thesis in collaboration with University of Crete, Heraklion, Greece. He is currently pursuing the Ph.D. degree at the Signal Processing and Speech Communication Laboratory, TUG.

His research interests include Bayesian networks, speech and audio analysis, as well as statistical pattern recognition.

Franz Pernkopf (M'05) received the M.Sc. (Dipl.-Ing.) degree in electrical engineering from the Graz University of Technology (TUG), Graz, Austria, in summer 1999 and the Ph.D. degree from the University of Leoben, Leoben, Austria, in 2002.

He was a Research Associate in the Department of Electrical Engineering, University of Washington, Seattle, from 2004 to 2006. Currently, he is an Assistant Professor at the Signal Processing and Speech Communication Laboratory, TUG. His research interests include machine learning, Bayesian networks, feature selection, finite mixture models, vision, speech, and statistical pattern recognition.

Prof. Pernkopf was awarded the Erwin Schrödinger Fellowship in 2002.