applied sciences

Article

Automatic Transcription of Polyphonic Vocal Music [†]

Andrew McLeod [1,*,‡], Rodrigo Schramm [2,‡], Mark Steedman [1] and Emmanouil Benetos [3]

[1] School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, UK; steedman@inf.ed.ac.uk
[2] Departamento de Música, Universidade Federal do Rio Grande do Sul, Porto Alegre 90020, Brazil; rschramm@ufrgs.br
[3] School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, UK; emmanouil.benetos@qmul.ac.uk
[*] Correspondence: A.McLeod-5@sms.ed.ac.uk; Tel.: +44-131-650-1000

[†] This paper is an extended version of our paper published in R. Schramm, A. McLeod, M. Steedman, and E. Benetos. Multi-pitch detection and voice assignment for a cappella recordings of multiple singers. In 18th International Society for Music Information Retrieval Conference (ISMIR), pp. 552–559, 2017.
[‡] These authors contributed equally to this work.

Academic Editor: Meinard Müller
Received: 31 October 2017; Accepted: 4 December 2017; Published: 11 December 2017

Abstract: This paper presents a method for automatic music transcription applied to audio recordings of a cappella performances with multiple singers. We propose a system for multi-pitch detection and voice assignment that integrates an acoustic and a music language model. The acoustic model performs spectrogram decomposition, extending probabilistic latent component analysis (PLCA) using a six-dimensional dictionary with pre-extracted log-spectral templates. The music language model performs voice separation and assignment using hidden Markov models that apply musicological assumptions. By integrating the two models, the system is able to detect multiple concurrent pitches in polyphonic vocal music and assign each detected pitch to a specific voice type such as soprano, alto, tenor or bass (SATB). We compare our system against multiple baselines, achieving state-of-the-art results for both multi-pitch detection and voice assignment on a dataset of Bach chorales and another of barbershop quartets. We also present an additional evaluation of our system using varied pitch tolerance levels to investigate its performance at 20-cent pitch resolution.

Keywords: automatic music transcription; multi-pitch detection; voice assignment; music signal analysis; music language models; polyphonic vocal music; music information retrieval

1. Introduction

Automatic music transcription (AMT) is one of the fundamental problems of music information retrieval and is defined as the process of converting an acoustic music signal into some form of music notation [1]. A core problem of AMT is multi-pitch detection, the detection of multiple concurrent pitches from an audio recording. While much work has gone into the field of multi-pitch detection in recent years, it has frequently been constrained to instrumental music, most often piano recordings due to a wealth of available data. Vocal music has been less often studied, likely due to the complexity and variety of sounds that can be produced by a singer.

Spectrogram factorization methods have been used extensively in the last decade for multi-pitch detection [1]. These approaches decompose an input time-frequency representation (such as a spectrogram) into a linear combination of non-negative factors, often consisting of spectral atoms and note activations. The most successful of these spectrogram factorization methods have been based on non-negative matrix factorisation (NMF) [2] or probabilistic latent component analysis (PLCA) [3].

While these spectrogram factorisation methods have shown promise for AMT, their parameter estimation can suffer from local optima, a problem that has motivated a variety of approaches that incorporate additional knowledge in an attempt to achieve more meaningful decompositions. Vincent et al. [4] used an adaptive spectral decomposition for multi-pitch detection assuming that the input signal can be decomposed as a sum of narrowband spectra. Kameoka et al. [5] exploited structural regularities in the spectrograms during the NMF process, adding constraints and regularization to reduce the degrees of freedom of their model. These constraints are based on time-varying basis spectra (e.g., using sound states: "attack", "decay", "sustain" and "release") and have since been included in other probabilistic models [6,7]. Fuentes et al. [8] introduced the concept of brakes, slowing the convergence rate of any model parameter known to be properly initialized. Other approaches [7,9,10] avoid undesirable parameter convergence using pre-learning steps, where spectral atoms of specific instruments are extracted in a supervised manner. Using the constant-Q transform (CQT) [11] as the input time-frequency representation, some approaches developed techniques using shift-invariant models over log-frequency [6,10,12], allowing for the creation of a compact set of dictionary templates that can support tuning deviations and frequency modulations. Shift-invariant models are also used in several recent approaches for automatic music transcription [6,13,14]. O'Hanlon et al. [15] propose stepwise and gradient-based methods for non-negative group sparse decompositions, exploring the use of subspace modelling of note spectra. This group sparse NMF approach is used to tune a generic harmonic subspace dictionary, improving automatic music transcription results based on NMF. However, despite promising results of template-based techniques [7,9,10], the considerable variation in the spectral shape of pitches produced by different sources can still affect generalization performance.

Recent research on multi-pitch detection has also focused on deep learning approaches: in [16,17], feedforward, recurrent and convolutional neural networks were evaluated towards the problem of automatic piano transcription. While the aforementioned approaches focus on the task of polyphonic piano transcription due to the presence of sufficiently large piano-specific datasets, the recently released MusicNet dataset [18] provides a large corpus for multi-instrument music suitable for training deep learning methods for the task of polyphonic music transcription. Convolutional neural networks were also used in [19] for learning salience representations for fundamental frequency estimation in polyphonic audio recordings.

Multi-pitch detection of vocal music represents a significant step up in difficulty as the variety of sounds produced by a single singer can be both unique and wide-ranging. The timbre of two singers' voices can differ greatly, and even for a single singer, different vowel sounds produce extremely varied overtone patterns. For vocal music, Bohak and Marolt [20] propose a method for transcribing folk music containing both instruments and vocals, which takes advantage of melodic repetitions present in that type of music using a musicological model for note-based transcription. A less explored type of music is a cappella; in particular, vocal quartets constitute a traditional form of Western music, typically dividing a piece into multiple vocal parts such as soprano, alto, tenor and bass (SATB). In [21], an acoustic model based on spectrogram factorisation was proposed for multi-pitch detection of such vocal quartets.

A small group of methods has attempted to go beyond multi-pitch detection, towards instrument assignment (also called timbre tracking) [9,22,23], where systems detect multiple pitches and assign each pitch to a specific source that produced it. Bay et al. [22] tracked individual instruments in polyphonic instrumental music using a spectrogram factorisation approach with continuity constraints controlled by a hidden Markov model (HMM). To the authors' knowledge, no methods have yet been proposed to perform both multi-pitch detection and instrument/voice assignment on polyphonic vocal music.

An emerging area of automatic music transcription attempts to combine acoustic models (those based on audio information only) with music language models, which model sequences of notes and other music cues based on knowledge from music theory or from constraints automatically derived from symbolic music data. This is in direct analogy to automatic speech recognition systems, which typically combine an acoustic model with a spoken language model. Ryynanen and Klapuri [24], for example, combined acoustic and music language models for polyphonic music transcription, where the musicological model estimates the probability of a detected note sequence. Another example of such an integrated system is the work by Sigtia et al. [16], which combined neural network-based acoustic and music language models for multi-pitch detection in piano music. The system used various types of neural networks for the acoustic component (feedforward, recurrent, convolutional) along with a recurrent neural network acting as a language model for modelling the correlations between pitch combinations over time.

Combining instrument assignment with this idea of using a music language model, it is natural to look towards the field of voice separation [25], which involves the separation of pitches into streams of notes, called voices, and is mainly addressed in the context of symbolic music processing. It is important to note that voice separation, while similar to our task of voice assignment, is indeed a distinct task. Specifically, while both involve an initial step of separating the incoming notes into voices, voice assignment involves a further step of labelling each of those voices as a specific part or instrument, in our case soprano, alto, tenor or bass.

Most symbolic voice separation approaches are based on voice leading rules, which have been investigated and described from a cognitive perspective in a few different works [26–28]. Among these rules, three main principles emerge: (1) large melodic intervals between consecutive notes in a single voice should be avoided; (2) two voices should not, in general, cross in pitch; and (3) the stream of notes within a single voice should be relatively continuous, without long gaps of silence, ensuring temporal continuity.

There are many different definitions of what precisely constitutes a voice, both perceptually and musically, discussed more fully in [25]; however, for our purposes, a voice is quite simply defined as the notes sung by a single vocalist. Therefore, our interest in voice separation models lies with those that separate notes into strictly monophonic voices (i.e., those that do not allow for concurrent notes), rather than polyphonic voices as in [29]. We would also like our chosen model to be designed to be run in a mostly unsupervised fashion, rather than being designed for use with human interaction (as in [30]), and for it not to require background information about the piece, such as time signature or metrical information (as in [31]). While many voice separation models remain that meet our criteria [32–36], the one described in [37] is the most promising for our use because it both (1) achieves state-of-the-art performance and (2) can be applied directly to live performance.

In this work, we present a system able to perform multi-pitch detection of polyphonic a cappella vocal music, as well as assign each detected pitch to a particular voice (soprano, alto, tenor or bass), where the number of voices is known a priori. Our approach uses an acoustic model for multi-pitch detection based on probabilistic latent component analysis (PLCA), which is modified from the model proposed in [21], and an HMM-based music language model for voice assignment based on the model of [37]. Compared to our previous work [38], this model contains a new dynamic dictionary voice type assignment step (described in Section 2.3), which accounts for its increased performance. Although previous work has integrated musicological information for note event modelling [16,20,24], to the authors' knowledge, this is the first attempt to incorporate an acoustic model with a music language model for the task of voice or instrument assignment from audio, as well as the first attempt to propose a system for voice assignment in polyphonic a cappella music. The approach described in this paper focuses on recordings of singing performances by vocal quartets without instrumental accompaniment; to that end, we use two datasets containing a capella recordings of Bach chorales and barbershop quartets. The proposed system is evaluated both in terms of multi-pitch detection and voice assignment, where it reaches an F-measure of over 70% and 50% for the two respective tasks.

The remainder of this paper is organised as follows. In Section 2, we describe the proposed approach, consisting of the acoustic model, the music language model and model integration. In Section 3, we report on experimental results using two datasets comprising recordings of vocal quartets. Section 4 closes with conclusions and perspectives for future work.

2. Proposed Method

In this section, we present a system for multi-pitch detection and voice assignment applied to audio recordings of polyphonic vocal music (where the number of voices is known a priori) that integrates an acoustic model with a music language model. First, we describe the acoustic model, a spectrogram factorization process based on probabilistic latent component analysis (PLCA). Then, we present the music language model, an HMM-based voice assignment model. Finally, a joint model is proposed for the integration of these two components. Figure 1 illustrates the proposed system pipeline.

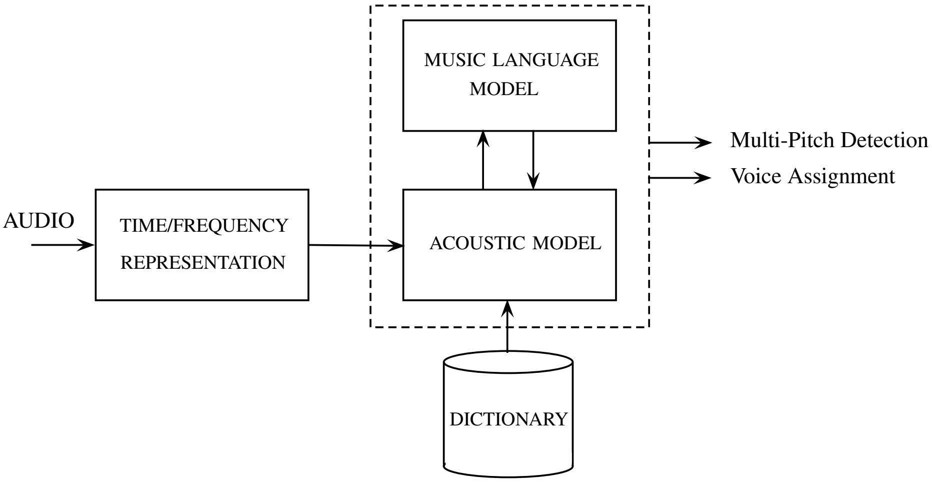


Figure 1. Proposed system diagram.

2.1. Acoustic Model

The acoustic model is a variant of the spectrogram factorisation-based model proposed in [21]. The model's primary goal is to explore the factorization of an input log-frequency spectrogram into components that have a close connection with singing characteristics such as voice type and the vocalization of different vowel sounds. We formulate the model dictionary templates into a six-dimensional tensor, representing log-frequency index, singer source, pitch, tuning deviation with 20 cent resolution, vowel type and voice type. Similarly to [9], the singer source and vowel type parameters constrain the search space into a mixture-of-subspaces, clustering a large variety of singers into a small number of categories. In this model, the voice type parameter corresponds to the vocal part (SATB), where each vocal part is linked to a distinct set of singers (the singer source). For details on the dictionary construction, see Section 2.1.2. As time-frequency representation, we use a normalised variable-Q transform (VQT) spectrogram [39] with a hop size of 20 ms and 20-cent frequency resolution. For convenience, we have chosen a pitch resolution that produces an integer number of bins per semitone (five in this case) and is also close to the range of just noticeable differences in musical intervals [40]. The input VQT spectrogram is denoted as $X_{\omega,t} \in \mathbb{R}^{\Omega \times T}$, where ω denotes log-frequency and t time. In the model, $X_{\omega,t}$ is approximated by a bivariate probability distribution $P(\omega, t)$, which is in turn decomposed as:

$$P(\omega, t) = P(t) \sum_{s,p,f,o,v} P(\omega|s, p, f, o, v) P_t(s|p) P_t(f|p) P_t(o|p) P(v) P_t(p|v) \qquad (1)$$

where $P(t)$ is the spectrogram energy (known quantity) and $P(\omega|s, p, f, o, v)$ is the fixed pre-extracted spectral template dictionary. The variable s denotes the singer index (out of the collection of singer subjects used to construct the input dictionary); $p \in \{21, \ldots, 108\}$ denotes pitch in Musical Instrument Digital Interface (MIDI) scale; f denotes tuning deviation from 12-tone equal temperament in 20-cent resolution ($f \in \{1, \ldots, 5\}$, with $f = 3$ denoting ideal tuning); o denotes the vowel type; and v denotes the voice type (e.g., soprano, alto, tenor, bass). The contribution of specific singer subjects from the training dictionary is modelled by $P_t(s|p)$, i.e., the singer contribution per pitch over time. $P_t(f|p)$ is the tuning deviation per pitch over time, and finally, $P_t(o|p)$ is the time-varying vowel contribution per pitch . (Although $P_t(o|p)$ is not explicitly used in this proposed approach, it is kept to ensure consistency with the Real World Computing (RWC) audio dataset [41] structure (see Section 2.1.2).) Unlike in [21] (which uses $P_t(v|p)$), this model decomposes the probabilities of pitch and voice type as $P(v)P_t(p|v)$. That is, $P(v)$ can be viewed as a mixture weight that denotes the overall contribution of each voice type to the whole input recording, and $P_t(p|v)$ denotes the pitch activation for a specific voice type (e.g., SATB) over time.

The factorization can be achieved by the expectation-maximization (EM) algorithm [42], where the unknown model parameters $P_t(s|p)$, $P_t(f|p)$, $P_t(o|p)$, $P_t(p|v)$ and $P(v)$ are iteratively estimated. In the expectation step, we compute the posterior as:

$$P_t(s,p,f,o,v|\omega) = \frac{P(\omega|s,p,f,o,v)P_t(s|p)P_t(f|p)P_t(o|p)P(v)P_t(p|v)}{\sum_{s,p,f,o,v} P(\omega|s,p,f,o,v)P_t(s|p)P_t(f|p)P_t(o|p)P(v)P_t(p|v)} \tag{2}$$

In the maximization step, each unknown model parameter is then updated using the posterior from Equation (2):

$$P_t(s|p) \propto \sum_{f,o,v,\omega} P_t(s,p,f,o,v|\omega)X_{\omega,t} \tag{3}$$

$$P_t(f|p) \propto \sum_{s,o,v,\omega} P_t(s,p,f,o,v|\omega)X_{\omega,t} \tag{4}$$

$$P_t(o|p) \propto \sum_{s,f,v,\omega} P_t(s,p,f,o,v|\omega)X_{\omega,t} \tag{5}$$

$$P_t(p|v) \propto \sum_{s,f,o,\omega} P_t(s,p,f,o,v|\omega)X_{\omega,t} \tag{6}$$

$$P(v) \propto \sum_{s,f,o,p,\omega,t} P_t(s,p,f,o,p|\omega)X_{\omega,t} \tag{7}$$

The model parameters are randomly initialised, and the EM algorithm iterates over Equations (2)–(7). In our experiments, we use 30 iterations, as this ensures that the model will converge; in practice, the model converges after about 18 iterations. In order to promote temporal continuity, we apply a median filter to the $P_t(p|v)$ estimate across time, before its normalisation at each EM iteration, using a filter span of 240 ms, a duration of approximately half of one beat in Allegro tempo.

2.1.1. Acoustic Model Output

The output of the acoustic model is a semitone-scale pitch activity tensor for each voice type and a pitch shifting tensor, given by $P(p,v,t) = P(t)P(v)P_t(p|v)$ and $P(f,p,v,t) = P(t)P(v)P_t(p|v)P_t(f|p)$, respectively. By stacking together slices of $P(f,p,v,t)$ for all values of p, we can create a 20-cent resolution time-pitch representation for each voice type v:

$$P(f',v,t) = P\left(f' \pmod 5 + 1, \left\lfloor \frac{f'}{5} \right\rfloor + 21, v, t\right) \tag{8}$$

where $f' \in \{0,...,439\}$ denotes pitch in 20-cent resolution. The voice-specific 20-cent resolution pitch activation output is given by $P(f',v,t)$, and the overall multi-pitch activations without voice assignment are given by $P(f',t) = \sum_v P(f',v,t)$. The 20-cent resolution multi-pitch activations $P(f',t)$ are converted into multi-pitch detections, represented by a binary matrix $\boldsymbol{B}(f',t)$, through a binarisation process with a fixed threshold L_{th}. Specifically, pitch activations whose values are greater than L_{th} are set to one in matrix $\boldsymbol{B}$, while all others are set to zero.

This binarised matrix $\boldsymbol{B}(f', t)$ is then post-processed in order to obtain more accurate pitch activations. In this step, we scan each time frame of the matrix $\boldsymbol{B}$, replacing the pitch candidates by the position of spectrogram peaks detected from $X_{\omega,t}$ and that are validated by a minimum pitch distance rule:

$$(\Delta_{peaks}(\boldsymbol{X_t}, \boldsymbol{B}(f', t)) < T_1) \vee (\Delta_{peaks}(\boldsymbol{X_t}, \boldsymbol{B}(f', t-1)) < T_2), \tag{9}$$

where $\boldsymbol{B}(f', t)$ represents each binarised pitch activation at time frame t. The function Δ_{peaks} in (9) indicates the minimum pitch distance between the selected list of peak candidates in $\boldsymbol{X_t}$ and each pitch candidate $\boldsymbol{B}(f', t)$ and $\boldsymbol{B}(f', t-1)$, respectively. In our experiments, we use $T_1 = 1$ and $T_2 = 3$, based on density distributions of $|\Delta_{peaks}|$, which were estimated from measurements in our datasets using the pitch ground truth. The use of the previous frame $(t-1)$ helps to keep the temporal continuity when a pitch candidate is eventually removed by the L_{th} threshold.

2.1.2. Dictionary Extraction

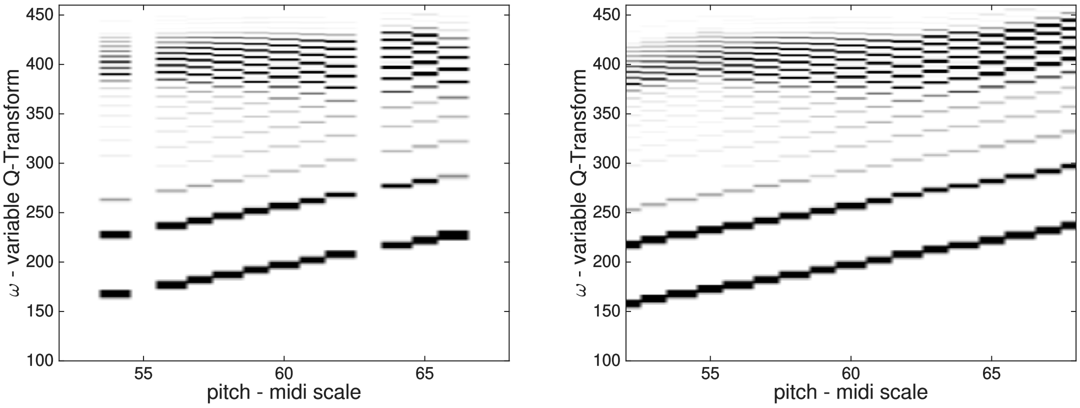
Dictionary $P(\omega|s, p, f, o, v)$ with spectral templates from multiple singers is built based on English pure vowels (monophthongs), such as those used in the solfège system of learning music: Do, Re, Mi, Fa, Sol, La, Ti and Do. The dictionaries use spectral templates extracted from solo singing recordings in the Musical Instrument Sound subset of the Real World Computing (RWC) database (RWC-MDB-I-2001 Nos. 45–49) [41]. The recordings contain sequences of notes following a chromatic scale, where the range of notes varies according to the tessitura of distinct vocal parts. Each singer sings a scale in five distinct English vowels (/a/, /æ/, /i/, /ɒ/, /u/). In total, we have used 15 distinct singers: 9 male and 6 female, consisting of 3 human subjects for each voice type (bass, baritone, tenor, alto, soprano).

Although the aim of this work is the transcription of vocal quartets, we keep the spectral templates from all five voice types in the dictionary because we do not know in advance the voice types present in each audio recording. This decision allows the dictionary to cover a wider variety of vocal timbres during the spectral decomposition, although not all of the resulting voice assignment probabilities will be used during its integration with the music language model for a single song. Rather, our model dynamically aligns one of the dictionary's voice types to each vocal part in a song. This dynamic dictionary alignment is based on the music language model's voice assignments and is discussed further in Section 2.3.

The fundamental frequency (f_0) sequence from each monophonic recording is estimated using the Probabilistic YIN (PYIN) algorithm [43]. Afterwards, the time-frequency representation is extracted using the VQT, with 60 bins per octave. A spectral template is extracted for each frame, regarding the singer source, vowel type and voice type. In order to incorporate multiple estimates from a common pitch, the set of estimates that fall inside the same pitch bin are replaced by its metrically-trimmed mean, discarding 20% of the samples as possible outliers. The use of the metrically-trimmed mean aims to reduce the influence of possible pitch inaccuracies obtained from the automatic application of the PYIN algorithm. However, there is no guarantee that the final estimate will be free of eventual outliers. The set of spectral templates is then pre-shifted across log-frequency in order to support tuning deviations for ±20 and ±40 cent and are stored into a six-dimensional tensor matrix $P(\omega|s, p, f, o, v)$. Due to the available data from the chromatic scales, the resulting dictionary $P(\omega|s, p, f, o, v)$ has some pitch templates missing, as shown in Figure 2a.

To address the aforementioned issue, we have investigated alternative ways to fill out the missing templates in the dictionary, including spectrum estimation by replication [14,44], linear and nonlinear interpolation and a generative process based on Gaussian mixture models (inspired by [45,46]). Following experimentation, we have chosen the replication approach, where existing templates belonging to the same dictionary are used to fill in the missing parts of the pitch scale, as this has been shown to achieve the best performance [47]. In this approach, the spectral shape of a given pitch p_n is repeated (with the appropriate log-frequency shift) over all subsequent pitches

$p \in [p_{n+1}, p_{m-1}]$ until another template is found (the pitch template p_m). Figure 2b illustrates the resulting dictionary templates of one singer example (vowel /a/) from our audio dataset, following the above replication process.



(**a**) (**b**)

Figure 2. Example from an /a/ vowel utterance (one singer) templates: (**a**) original templates from the variable-Q transform (VQT) spectrogram; (**b**) revised dictionary templates following replication.

2.2. Music Language Model

The music language model attempts to assign each detected pitch to a single voice based on musicological constraints. It is a variant of the HMM-based voice separation approach proposed in [37], where the main change is to the emission function (here it is probabilistic, while in the previous work, it was deterministic). The model separates sequential sets of multi-pitch activations into monophonic voices (of type SATB) based on three principles: (1) consecutive notes within a voice tend to occur on similar pitches; (2) there are minimal temporal gaps between them; and (3) voices are unlikely to cross.

The observed data for the HMM are notes generated from the acoustic model's binarised 20-cent resolution multi-pitch activations $\boldsymbol{B}(f', t)$, where each activation generates a note n with pitch $\text{Pitch}(n) = \lfloor \frac{f'}{5} \rfloor$, onset time $\text{On}(n) = t$ and offset time $\text{Off}(n) = t + 1$. Duplicates are discarded in the case where two 20-cent resolution detections refer to the same semitone pitch. O_t represents this set of observed notes at frame t.

2.2.1. State Space

In the HMM, a state S_t at frame t contains a list of M monophonic voices V_i, $1 \leq i \leq M$. M is set via a parameter, and in this work, we use $M = 4$. In the initial state S_0, all of the voices are empty, and at each frame, each voice may be assigned a single note (or no note). Thus, each voice contains the entire history of the notes, which have been assigned to it from Frame 1 to t. This is necessary because the note history is used in the calculation of the transition probabilities (Section 2.2.2); however, it causes the theoretical state space of our model to blow up exponentially. Therefore, instead of using precomputed transition and emission probabilities, we must use transition and emission probability functions, presented in the following sections.

Conceptually, it is helpful to think of each state as simply a list of M voices. Thus, each state transition is calculated based on the voices in the previous state (though some of the probability calculations require knowledge of individual notes).

2.2.2. Transition Function

A state S_{t-1} has a transition to state S_t if and only if each voice $V_i \in S_{t-1}$ can either be transformed into the corresponding $V_i \in S_t$ by assigning to it a single note with onset time t, or if it is identical to the corresponding $V_i \in S_t$.

This transition from S_{t-1} to S_t can be represented by the variable T_{S_{t-1},N_t,W_t}, where S_{t-1} is the original state, N_t is a list of every note with onset time t assigned to a voice in S_t and W_t is a list of integers, each representing the voice assignment index for the corresponding note N_t. Specifically, N_t and W_t are of equal length, and the i-th integer in W_t represents the index of the voice to which the i-th note in N_t is assigned in S_t. Notice that here, N_t only contains those observed notes that are assigned to a voice in S_t, rather than all observed notes.

The HMM transition probability $P(S_t|S_{t-1})$ is defined as $P(T_{S_{t-1},N_t,W_t})$:

$$P(T_{S_{t-1},N_t,W_t}) = \Psi(W_t) \prod_{i=1}^{|N_t|} C(S_{t-1}, n_i, w_i) P(V_{w_i}, n_i) \tag{10}$$

The first term in the above product is a function representing the voice assignment probability and is defined as follows:

$$\Psi(W) = \prod_{j=1}^{M} \begin{cases} P_v & j \in W \\ 1 - P_v & j \notin W \end{cases} \tag{11}$$

Here, the parameter P_v is the probability that a given voice contains a note in a frame.

$C(S_{t-1}, n, w)$ is a penalty function used to minimize the voice crossings, which are rare, though they do sometimes occur. It returns by default one, but its output is multiplied by a parameter P_{cross}—representing the probability of a voice being out of pitch order with an adjacent voice—for each of the following cases that applies:

1. $w > 1$ and $\text{Pitch}(V_{w-1}) > \text{Pitch}(n)$
2. $w < M$ and $\text{Pitch}(V_{w+1}) < \text{Pitch}(n)$

These cases in fact provide the definition for precisely what constitutes two voices being "out of pitch order". For example, if the soprano voice contains a note at a lower pitch than the alto voice in a given frame, the soprano voice is said to be out of pitch order. Cases 1 and 2 apply when a note is out of pitch order with the preceding or succeeding voice in the state, respectively. $\text{Pitch}(V)$ represents the pitch of a voice and is calculated as a weighted sum of the pitches of its most recent l (a parameter) notes, where each note's weight is twice the weight of the previous note. Here, n_i refers to the i-th note assigned to voice V.

$$\text{Pitch}(V) = \frac{\sum\limits_{i=0}^{\min(l,|V|)} (2^i \text{Pitch}(n_{|V|-i}))}{\sum\limits_{i=0}^{\min(l,|V|)} 2^i} \tag{12}$$

$P(V, n)$ represents the probability of a note n being assigned to a voice V and is the product of a pitch score and a gap score.

$$P(V, n) = \text{pitch}(V, n)\ \text{gap}(V, n) \tag{13}$$

The pitch score, used to minimise melodic jumps within a voice, is computed as shown in Equation (14), where $\mathcal{N}(\mu, \sigma, x)$ represents a normal distribution with mean μ and standard deviation σ evaluated at x, and σ_p is a parameter. The gap score is used to prefer temporal continuity within a voice and is computed using Equation (15), where $\text{Off}(V)$ is the offset time of the most recent note in V, and σ_g and g_{min} are parameters. Both Δ_p and Δ_g return one if V is empty.

$$\text{pitch}(V, n) = \mathcal{N}(\text{Pitch}(V), \sigma_p, \text{Pitch}(n)) \tag{14}$$

$$\text{gap}(V, n) = \max\left(\ln(-\frac{\text{On}(n) - \text{Off}(V)}{\sigma_g} + 1) + 1, g_{min}\right) \tag{15}$$

2.2.3. Emission Function

A state S_t emits a set of notes with onset at time t, with the constraint that a state containing a voice with a note at onset time t must emit that note. The probability of a state S_t emitting the note set O_t is shown in Equation (16), using the voice posterior $P_t(v|p)$ from the acoustic model.

$$P(O_t|S_t) = \prod_{n \in O_t} \begin{cases} P_t(v = i|p = \rho(n)) & n \in V_i \in S_t \\ 1 & \text{otherwise} \end{cases} \tag{16}$$

Notice that a state is not penalised for emitting notes not assigned to any of its voices. This allows the model to better handle false positives from the multi-pitch detection. For example, if the acoustic model detects more than M pitches, the state is allowed to emit the corresponding notes without penalty. We do, however, penalise a state for not assigning a voice any note during a frame, but this is handled by $\Psi(W)$ from Equation (11).

2.2.4. Inference

To find the most likely final state given our observed note sets, we use the Viterbi algorithm [48] with beam search with beam size b. That is, after each iteration, we save only the $b = 50$ most likely states given the observed data to that point, in order to handle the complexity of the HMM. A simple two-voice example of the HMM being run discriminatively can be found in Figures 3 and 4.

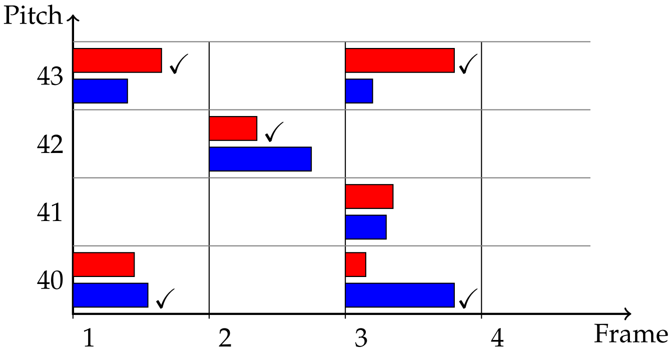


Figure 3. An example of an input to the music language model given a simple song with only two voices. Here, for each detected pitch, there are two bars, representing the relative value of $P_t(p|v)$ for each voice at that frame. (The upper voice is shown in red and the lower voice is shown in blue.) The ground truth voice assignment for each detected note is given by a check mark next to the bar representing the correct voice. Notice that there is a false positive pitch detection at Pitch 41 at Frame 3.

Figure 3 shows example input pitch detections, where empty grid cells represent pitches that have not passed the PLCA's post-processing binarisation step, and the bars in the other cells represent relative values of $P_t(p|v)$ for each colour-coded voice. (The upper voice is shown in red and the lower voice is shown in blue.) There is a check mark next to the bar representing the ground-truth voice assignment for each detected pitch. Notice that there is no check mark in the cell representing Pitch 41 at Frame 3, indicating a false positive pitch detection.

Figure 4 shows the HMM decoding process of the input from Figure 3, using a beam size of two and two voices. Notes are represented as "[pitch, frame]" and are colour-coded based on their ground truth voice assignment. (Notes belonging to the upper voice are shown in red and notes belonging to the lower voice are shown in blue.) Again, notice false positive pitch detection [41, 3]. In this figure, the emission sets O_t are shown on the bottom, and the boxes below each O_t node list the emitted notes in decreasing pitch order. Meanwhile, the voices contained by a state at each time step are listed in the boxes above each S_t node, where voices are listed in decreasing pitch order and are separated by braces. The most likely state hypothesis at each time step is on the bottom row, and each state box (except for

S_0) has an incoming arrow indicating which prior state hypothesis was used to transition into that state. Those state hypotheses with an entirely correct voice assignment are represented by a thick border.

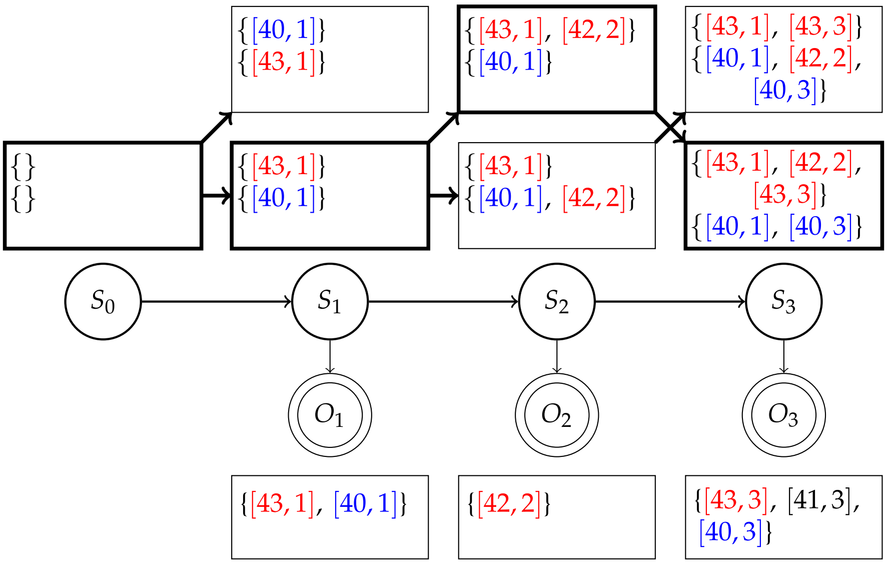


Figure 4. An example of the music language model being run on the detected pitches from Figure 3 with a beam size of two and two voices. Notes are represented as "[pitch,frame]" and are colour-coded based on their ground truth voice assignment. (Notes belonging to the upper voice are shown in red and notes belonging to the lower voice are shown in blue.) The observed note sets are listed beneath each O_t. Notice the false positive pitch detection $[41, 3]$ in O_3. The two most likely state hypotheses at each step are listed in the large rectangles above each state S_t, where the voices are notated with braces. The most likely state hypothesis at each step appears on the bottom row, and each state has an incoming arrow indicating which prior state hypothesis was used to transition into that state. Those state hypotheses with an entirely correct voice assignment are represented by a thick border.

Initially, S_0 contains two empty voices. Next, O_1 is seen, and the most likely voice assignment is also the correct one, assigning the pitches to the voices in decreasing pitch order. The second hypothesis for S_1 is very unlikely: the two voices are out of pitch order with each other, and its values of $P_t(p|v)$ are lower than the correct assignments. Thus, once O_2 is seen at Frame 2, that hypothesis drops out, and both hypothesis S_2 states transition from the most likely S_1 state. However, due to noisy $P_t(p|v)$ estimates from the PLCA, the most likely S_2 contains an incorrect assignment for the note $[42, 2]$, while the second S_2 hypothesis is correct. In S_3, however, these hypotheses flip back, resulting in the correct overall voice assignment for this example input. Notice that the false positive pitch detection $[41, 3]$ is not assigned to any hypothesis state since its values of $P_t(p|v)$ are relatively small. Meanwhile, the $P_t(p|v)$ estimates from the PLCA for the other two pitches are quite good and allow the HMM to correct itself (assuming good parameter settings), judging that the voice $\{[43, 1], [42, 2], [43, 3]\}$ in the higher voice is more likely than the voice $\{[40, 1], [42, 2], [40, 3]\}$ in the lower voice, even given the noisy $P_t(p|v)$ estimates for the note $[42, 2]$.

2.3. Model Integration

In this section, we describe the integration of the acoustic model and the music language model into a single system that jointly performs multi-pitch detection and voice assignment from audio. The pitch activations $P_t(p|v)$ for each voice type from the PLCA dictionary (bass, tenor, baritone, alto and soprano) are quite noisy, resulting in very low accuracy for voice assignment, as can be seen from our results (Table 1, row Schramm and Benetos [21]). However, we have found that a good prior distribution for $P_t(p|v)$ can drive the spectrogram factorisation towards a more meaningful voice assignment. This prior is given by the music language model, and its integration into the system pipeline is performed in two stages.

Table 1. Voice assignment results, where standard deviations are shown in parentheses. The post-processing refinement step described in Section 2.1.1 was also run on the output of all cited methods. For those that do not output any voice assignment information (Klapuri [49], Salamon and Gomez [50], Vincent et al. [4] and Pertusa and Iñesta [51]), the music language model was run once on its output with default settings and $M = 4$. VOCAL4-MP represents our proposed method with the acoustic model only. For VOCAL4-MP and [21], voice assignments are derived from each model's probabilistic voice assignment estimates ($P_t(v|p)$ for [21] and $P_t(p|v)$ for VOCAL4-MP). VOCAL4-VA refers to our fully-integrated model.

Model	Bach Chorales				
	F_{va}	F_s	F_a	F_t	F_b
Klapuri [49]	28.12 (4.38)	24.23 (10.28)	22.98 (11.85)	29.35 (12.43)	35.92 (10.97)
Salamon and Gomez [50]	24.83 (5.31)	30.03 (12.63)	25.24 (10.92)	21.09 (9.91)	22.95 (9.30)
Vincent et al. [4]	18.30 (4.87)	13.43 (7.03)	15.52 (6.50)	17.14 (6.77)	27.10 (8.44)
Pertusa and Iñesta [51]	44.05 (4.60)	40.18 (11.28)	43.34 (7.38)	41.54 (7.02)	50.56 (6.16)
Schramm and Benetos [21]	20.31 (3.40)	20.42 (5.36)	21.27 (4.75)	14.49 (1.37)	25.05 (2.12)
VOCAL4-MP	21.84 (9.37)	12.99 (11.23)	10.27 (10.13)	22.72 (6.72)	41.37 (9.41)
VOCAL4-VA	**56.49 (10.48)**	**52.37 (12.92)**	**49.13 (11.22)**	**53.10 (11.71)**	**71.38 (6.06)**

Model	Barbershop Quartets				
	F_{va}	F_s	F_a	F_t	F_b
Klapuri [49]	20.90 (5.79)	2.53 (4.82)	29.02 (13.25)	7.94 (7.48)	44.09 (14.26)
Salamon and Gomez [50]	20.38 (6.61)	11.14 (10.27)	35.14 (14.04)	8.44 (8.22)	26.81 (13.69)
Vincent et al. [4]	19.13 (8.52)	10.20 (8.25)	17.97 (9.03)	15.93 (8.85)	32.41 (12.41)
Pertusa and Iñesta [51]	37.19 (8.62)	30.68 (13.94)	**36.15 (11.70)**	29.15 (13.90)	52.78 (10.37)
Schramm and Benetos [21]	23.98 (4.34)	24.45 (6.36)	31.61 (6.79)	13.55 (2.18)	26.34 (2.03)
VOCAL4-MP	18.35 (7.56)	2.40 (5.54)	10.56 (13.92)	16.61 (7.31)	43.85 (3.46)
VOCAL4-VA	**49.06 (14.65)**	**41.78 (18.78)**	34.62 (16.29)	**35.59 (16.93)**	**84.25 (6.58)**

Since multi-pitch detections from the acoustic model are the input for the music language model, spurious detections can result in errors during the voice separation process. Therefore, in the first stage, we run the EM algorithm using only the acoustic model from Section 2.1 for 15 iterations to allow for convergence to stable multi-pitch detections. Next, the system runs for 15 more EM iterations, this time also using the music language model from Section 2.2. During each EM iteration in this second stage, the acoustic model is run first, and then, the language model is run on the resulting multi-pitch detections. To integrate the two models, we apply a fusion mechanism inspired by the one used in [52] to improve the acoustic model's pitch activations based on the resulting voice assignments.

The output of the language model is introduced into the acoustic model as a prior to $P_t(p|v)$. During the acoustic model's EM updates, Equation (6) is modified as:

$$P_t^{new}(p|v) = \alpha P_t(p|v) + (1-\alpha)\phi_t(p|v), \tag{17}$$

where α is a weight parameter controlling the effect of the acoustic and language model and ϕ is a hyperparameter defined as:

$$\phi_t(p|v) \propto P_t^a(p|v) P_t(p|v). \tag{18}$$

$P_t^a(p|v)$ is calculated from the most probable final HMM state $S_{t_{max}}$ using the pitch score $\Delta_p(V, n)$ from the HMM transition function of Equation (14). For V, we use the voice $V_v \in S_{t_{max}}$ as it was at frame $t-1$, and for n, we use a note at pitch p. The probability values are then normalised over all pitches per voice. The pitch score returns a value of one when the V is an empty voice (thus becoming a uniform distribution over all pitches). The hyperparameter of Equation (18) acts as a soft mask, reweighing the pitch contribution of each voice based on detected pitches from the previous iteration.

Performance depends on a proper alignment between the voice types present in each song and the voice types present in the PLCA dictionary. Therefore, we dynamically assign one of the five voice types present in the dictionary (see Section 2.1.2) to each of the voices extracted by the music language model. During the first integrated EM iteration, the acoustic model's voice probabilities $P_t(p|v)$ are set to a uniform distribution upon input to the music language model. Additionally, we cannot be sure which voice types are present in a given song, so we run the language model with $M = 5$. Here, the acoustic model's detections contain many overtones, and we do not want to simply use $M = 4$, because many of the overtones are actually assigned a slightly greater probability than the correct notes by the acoustic model. Rather, the overtones tend to be higher in pitch than the correct notes and, thus, are almost exclusively assigned to the fifth voice by the HMM. These decisions combined allow the music language model to drive the acoustic model towards the correct decomposition without being influenced by the acoustic model's initially noisy voice type probabilities.

After this initial HMM iteration, we make the dynamic dictionary voice type assignments using the following equation:

$$\text{VoiceType}(V_i) = \arg\max_{v} \sum_{p,t} P_t(p|v) P_t^a(p|V_i), \tag{19}$$

such that each voice V_i from the HMM is assigned the voice type v from the dictionary that gives the greatest correlation between the (initial, non-uniform) PLCA voice probabilities $P_t(p|v)$ and the HMM voice priors $P_t^a(p|V_i)$. This alignment procedure begins with the HMM's lowest voice and performs a greedy search, such that for each subsequent voice, the arg max only searches over those dictionary voice types not already assigned to a lower HMM voice. This dynamic dictionary voice type assignment allows the model to decide which voice types are present in a given song at runtime. For all subsequent iterations, this voice type assignment is saved and used during integration. Additionally, the HMM is now run with $M = 4$, and the voice type assignment is used to ensure that the PLCA output $P_t(p|v)$ estimates correspond to the correct voice indices in the HMM. This dynamic dictionary type alignment is a novel feature of the proposed model compared to our previous work [38].

We also place certain constraints on the HMM during its first iteration. Specifically, where O_t is the set notes observed at frame t: (1) if $|O_t| \leq M$, each note in O_t must be assigned to a voice in S_t; and (2) if $|O_t| > M$, the voices in S_t must contain exactly the M most likely pitch activations from O_t, according to the $P(p,t)$ from the acoustic model, where ties are broken such that lower pitches are considered more likely (since overtones are the most likely false positives).

The final output of the integrated system is a list of the detected pitches at each time frame that are assigned to a voice in the most probable final HMM state $S_{t_{max}}$, along with the voice assignment for each after the full 30 EM iterations. Figure 6 shows an example output of the integrated system and is discussed more in depth in Section 3.4.

3. Evaluation

3.1. Datasets

We evaluate the proposed model on two datasets of a capella recordings: one of 26 Bach chorales and another of 22 barbershop quartets, in total 104 minutes. (Original recordings are available at http://www.pgmusic.com/bachchorales.htm and http://www.pgmusic.com/barbershopquartet.htm respectively.) These are the same datasets used in [21], allowing for a direct comparison between it and the acoustic model proposed in Section 2.1. Each file is in wave format with a sample rate of 22.05 kHz and 16 bits per sample. Each recording has four distinct vocal parts (SATB), with one part per channel. The recordings from the barbershop dataset each contain four male voices, while the Bach chorale recordings each contain a mixture of two male and two female voices.

A frame-based pitch ground truth for each vocal part was extracted using a monophonic pitch tracking algorithm [43] on each individual monophonic track with default settings. Experiments are conducted using the mix down of each audio file with polyphonic content, not the individual tracks.

3.2. Evaluation Metrics

We evaluate the proposed system on both multi-pitch detection and voice assignment using the frame-based precision, recall and F-measure as defined in the Music Information Retrieval Evaluation eXchange (MIREX) multiple-F0 estimation evaluations [53], with a frame hop size of 20 ms.

The F-measure obtained by the multi-pitch detection is denoted as F_{mp}, and for this, we combine the individual voice ground truths into a single ground truth for each recording. For voice assignment, we simply use the individual voice ground truths and define voice-specific F-measures of F_s, F_a, F_t and F_b for each respective SATB vocal part. We also define an overall voice assignment F-measure F_{va} for a given recording as the arithmetic mean of its four voice-specific F-measures.

3.3. Training

To train the acoustic model, we use recordings from the RWC dataset [41] to generate the six-dimensional dictionary of log-spectral templates specified in Section 2.1, following the procedure described in Section 2.1.2.

For all parameters in the music language model, we use the values reported in [37] that were used for voice separation in the fugues, except that we double the value of σ_p to eight to better handle noise from the acoustic model. We also introduce two new parameters to the system: the voice crossing probability P_{cross} and the voice assignment probability P_v. We use MIDI files of 50 Bach chorales, available at http://kern.ccarh.org/ (none of which appear in the test set), splitting the notes into 20-ms frames, and measure the proportion of frames in which a voice was out of pitch order with another voice and the proportion of frames in which each voice contains a note. This results in values of $P_{cross} = 0.006$ and $P_v = 0.99$, which we use for testing.

To train the model integration weight α, we use a grid search on the range $[0.1, 0.9]$ with a step size of 0.1, maximising F_{va} for each of our datasets. Similarly, the value of the threshold L_{th} that is used for the binarisation of the multi-pitch activations in Section 2.1.1 is based on a grid search on the range $[0.0, 0.1]$ with a step size of 0.01, again maximising F_{va} for each dataset. To avoid overfitting, we employ cross-validation, using the parameter settings that maximise the chorales' F_{va} when evaluating the barbershop quartets, and vice versa; nonetheless, the resulting parameter settings are the same for both datasets: $\alpha = 0.1$ and $L_{th} = 0.01$.

3.4. Results

We use five baseline methods for evaluation: Vincent et al. [4], which uses an adaptive spectral decomposition based on NMF; Pertusa and Iñesta [51], which selects candidates among spectral peaks, validating candidates through additional audio descriptors; Schramm and Benetos [21], a PLCA model for multi-pitch detection from multi-singers, similar to the acoustic model of our proposed system, although it also includes a binary classifier to estimate the final pitch detections from the pitch activations; as well as two multi-pitch detection methods from the Essentia library [54]: Klapuri [49], which sums the amplitudes of harmonic partials to detect pitch presence; and Salamon and Gomez [50], which uses melodic pitch contour information to model pitch detections. For all five of these methods, we also run the post-processing refinement step described in Section 2.1.1 on their output.

We evaluate the above systems against two versions of our proposed model: VOCAL4-MP, using only the acoustic model described in Section 2.1; and VOCAL4-VA, using the fully-integrated model.

From the multi-pitch detection results in Table 2, it can be seen that our integrated model VOCAL4-MP achieves the highest F_{mp} on both datasets. In fact, VOCAL4-VA outperforms VOCAL4-MP substantially, indicating that the music language model is indeed able to drive the acoustic model to a more meaningful factorisation.

Table 2. Multi-pitch detection results, where standard deviations are shown in parentheses. The post-processing refinement step described in Section 2.1.1 was also run on the output of all cited methods. VOCAL4-MP represents our proposed method with the acoustic model only, while VOCAL4-VA refers to our fully-integrated model.

Model	Bach Chorales	Barbershop Quartets
Klapuri [49]	54.62 (3.00)	48.24 (4.50)
Salamon and Gomez [50]	49.52 (5.18)	45.22 (6.94)
Vincent et al. [4]	53.58 (6.27)	51.04 (8.52)
Pertusa and Iñesta [51]	67.19 (3.82)	63.85 (6.69)
Schramm and Benetos [21]	71.03 (3.33)	70.84 (6.17)
VOCAL4-MP	63.05 (3.12)	59.09 (5.07)
VOCAL4-VA	**71.76 (3.51)**	**75.70 (6.18)**

For voice assignment, using each baseline method above that does not output any voice assignment information (Klapuri [49], Salamon and Gomez [50], Vincent et al. [4] and Pertusa and Iñesta [51]), we run our music language model once on its output with default settings and $M = 4$, after the post-processing refinement step. Meanwhile, for Schramm and Benetos [21], as well as VOCAL4-MP, the voice assignments are derived from each model's probabilistic voice assignment estimates ($P_t(v|p)$ for [21] and $P_t(p|v)$ for VOCAL4-MP).

The voice assignment results are shown in Table 1, where it is shown that VOCAL4-VA outperforms the other models, suggesting that a language model is necessary for the task. It is also clear that integrating the language model as we have (rather than simply including one as a post-processing step) leads to greatly improved performance. Specifically, notice that the difference in performance between our model and the baseline methods is much greater for voice separation than for multi-pitch detection, even though we applied our language model to those baseline methods' results as post-processing.

Also interesting to note is that our model performs significantly better on the bass voice than on the other voices. While this is also true of many of the baseline methods, for none of them is the difference as great as with our model. Overtones are a major source of errors in our model, and the bass voice avoids these since it is almost always the lowest voice.

A further investigation into our model's performance can be found in Figure 5, which shows all of the VOCAL4-VA model's F-measures, averaged across all songs in the corresponding dataset after each EM iteration. The first thing to notice is the large jump in performance at Iteration 15, when the language model is first integrated into the process. This jump is most significant for voice assignment, but is also clear for multi-pitch detection. The main source of the improvement in multi-pitch detection is that the music language model helps to eliminate many false positive pitch detections using the integrated pitch prior. In fact, the multi-pitch detection performance improves again after the 16th iteration and then remains relatively stable throughout the remaining iterations.

The voice assignment results follow a similar pattern, though without the additional jump in performance after Iteration 16. In the Bach chorales, the voice separation performance even continues to improve until the end of all 30 iterations. For the barbershop quartets, however, the performance increases until Iteration 20, before decreasing slightly until the end of the process. This slight decrease in performance over the final 10 iterations is due to the alto and soprano voices: F_b and F_t each remain stable over the final 10 iterations, while F_a and F_s each decrease. This difference is likely explained by the acoustic model not being able to properly decompose the alto and soprano voices. The barbershop quartets have no true female voices (i.e., each part is sung by a male vocalist), but the template dictionary's alto and soprano voices are sung by female vocalists; thus, the alto and soprano parts must be estimated through a rough approximation of a spectral basis combination of female voices. Such a rough approximation could be the cause of our model's difficulty in decomposing the alto and soprano voices in the barbershop quartets.

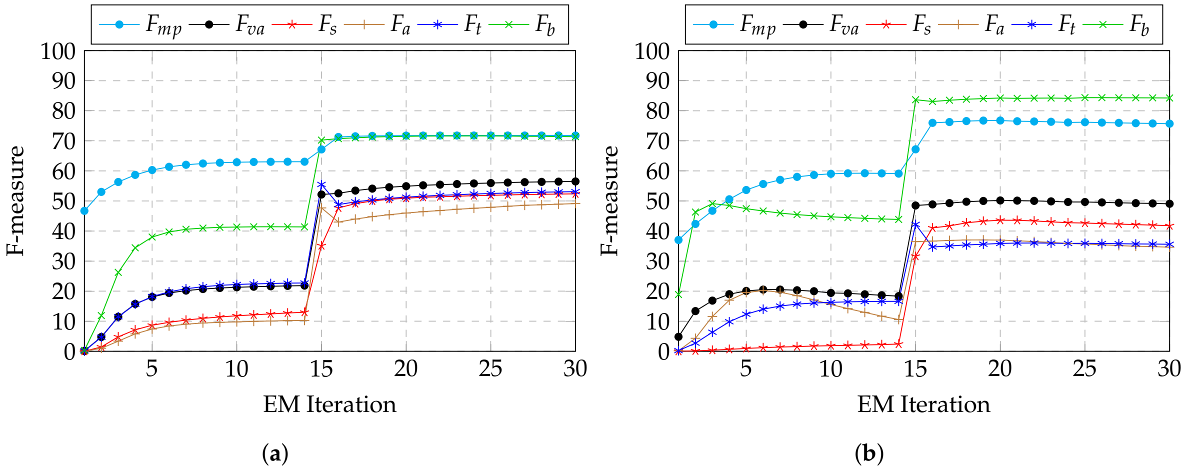


Figure 5. The VOCAL4-VA model's F-measures after each EM iteration, averaged across all songs in each dataset. (**a**) Bach chorales; (**b**) barbershop quartets.

Figure 6 illustrates the output of our proposed system, run on excerpts from both the Bach chorale (a, left) and barbershop quartet (b, right) datasets, for the joint multi-pitch detection and voice assignment tasks. Figure 6(a1),(b1) show the ground truth, using colour to denote vocal part; Figure 6(a2),(b2) show the probabilistic pitch detections from the acoustic model after the 30th EM iteration, summed over all voices ($\sum_{v=1}^{5} P_t(p|v)$), where a darker shade of gray indicates a greater probability; Figure 6(a3),(b3) present the final output of the integrated system, again using colour to denote vocal part.

As mentioned earlier, the bass voice assignment outperforms all other voice assignments in almost all cases, since false positive pitch detections from the acoustic model often correspond with overtones from lower notes that occur in the same pitch range as the correct notes from higher voices. These overtone errors are most commonly found in the soprano voice, for example at around 105 seconds in the Bach chorale excerpt and around 64.5 seconds in the barbershop quartet excerpt, where Figure 6(a2),(b2) clearly show high probabilities for these overtones. It is clear from Figure 6(a3),(b3) that such overtone errors in the soprano voice also lead to voice assignment errors in the lower voices since our system can now assign the correct soprano pitch detections to the alto voice, alto to tenor and tenor to bass.

Another common source of errors (for both multi-pitch detection and voice assignment) is vibrato. The acoustic model can have trouble detecting vibrato, and the music language model prefers voices with constant pitch over voices alternating between two pitches, leading to many off-by-one errors in pitch detection. Such errors are evident throughout the Bach chorale excerpt, particularly in the tenor voice towards the beginning where our system detects mostly constant pitches (both in the acoustic model output and the final output) while the ground truth contains some vibrato. Furthermore, at the end of both excerpts, there is vibrato present, and our system simply detects no pitches rather than the vibrato. This is most evident in the tenor voice of the Bach chorale, but is also evident in the soprano, alto and tenor voices of the barbershop quartet.

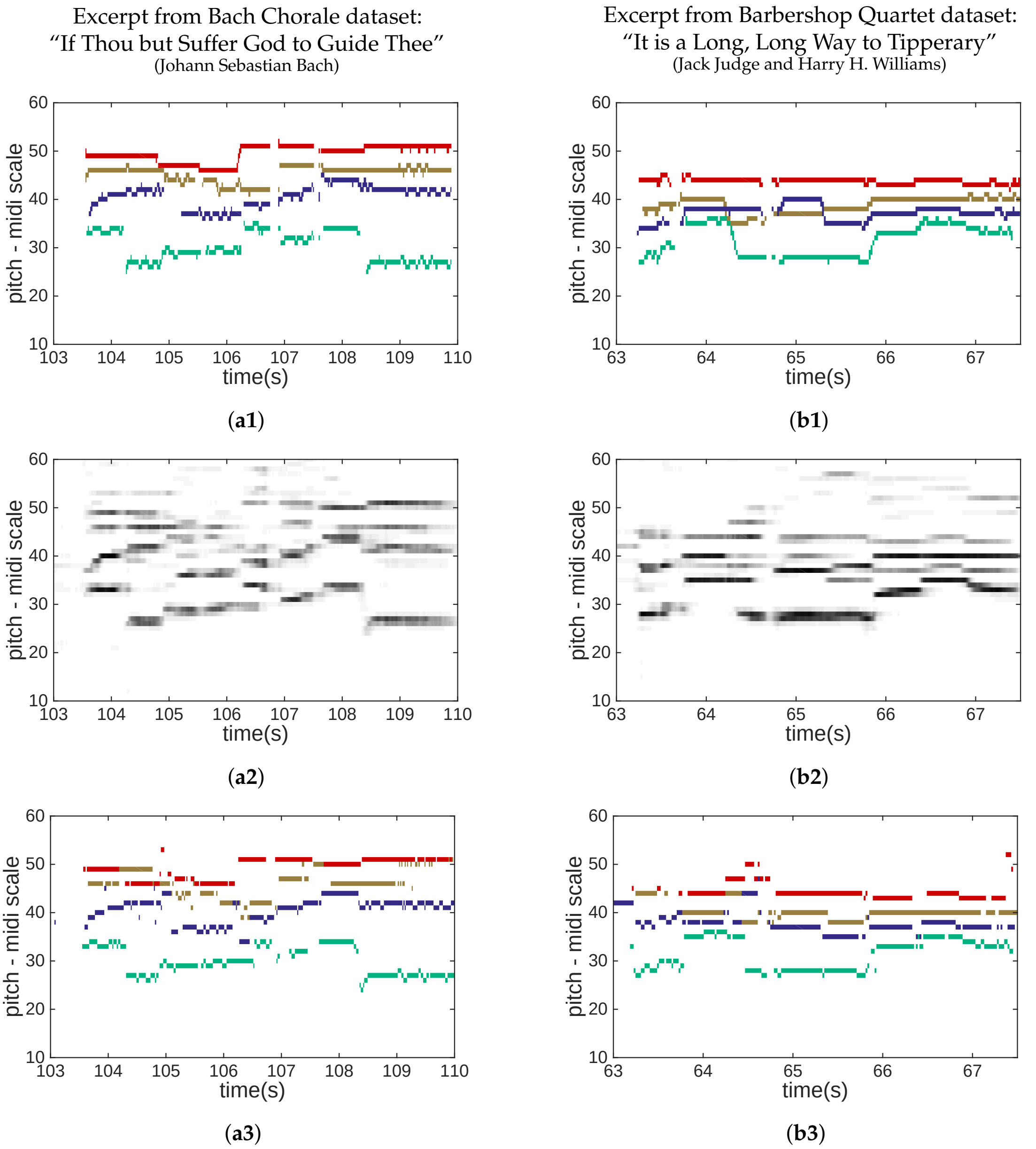


Figure 6. Example system input and output of excerpts from the Bach Chorale (**a**) (left) and barbershop quartet (**b**) (right) datasets. (**a1**,**b1**) show the ground truth, using colour to denote vocal part (red: soprano; brown: alto; blue: tenor; green: bass). (**a2**,**b2**) show the probabilistic pitch detections from the acoustic model after the 30th EM iteration, summed over all voices ($\sum_{v=1}^{5} P_t(p|v)$), where a darker shade of gray indicates a greater probability; (**a3**,**b3**) present the final output of the integrated system, again using colour to denote vocal part.

A closer look at errors from both vibrato and overtones can be found in Figure 7, which shows pitch detections (red) and ground truth (black) for the soprano voice from an excerpt of "O Sacred Head Sore Wounded" from the Bach chorales dataset. Here, errors from overtones can be seen around 108.5 seconds, where the detected pitch 54 is the second partial from the tenor voice (not shown), which is at pitch 42 at that time. Errors from vibrato are evident around 107.75 seconds and 108.6 seconds, where the pitch detections remain at a constant pitch while the ground truth switches between adjacent pitches.

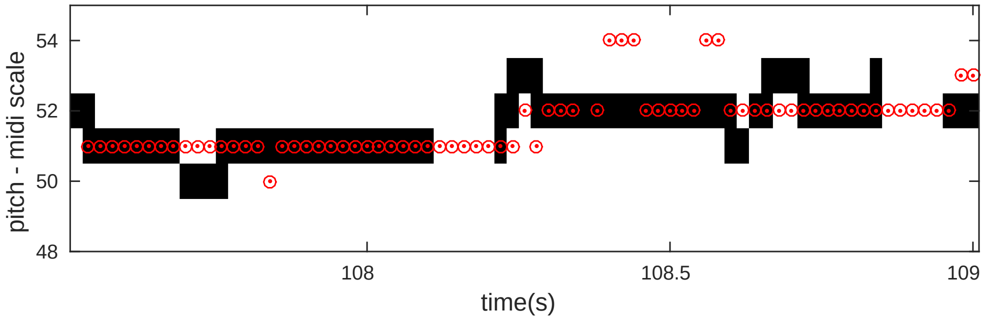


Figure 7. Pitch detections (red) and ground truth (black) for the soprano voice from an excerpt of "O Sacred Head Sore Wounded" from the Bach chorales dataset, showing errors from both vibrato and overtones (from the tenor voice, not shown).

Twenty-Cent Resolution

To further investigate our model's performance, especially on vibrato, we present its performance using 20-cent resolution instead of semitone resolution. Specifically, we divide each semitone into five 20 cent-wide frequency bins. We convert our integrated model's final semitone-based output into these bins using a post-processing step: for each detected pitch, we assign it to the 20-cent bin with the maximum $P_t(f|p)$ value from the acoustic model's final decomposition iteration.

Results are reported in terms of a cent-based pitch tolerance. A tolerance of zero cents means that a pitch detection will only be evaluated as a true positive if it is in the correct 20-cent bin. A tolerance of ±20 cents means that a pitch detection will be evaluated as a true positive if it is within one bin of the correct bin. In general, a tolerance of $\pm 20k$ cents will count any pitch detection falling within k bins of the correct bin as a true positive.

Figure 8 illustrates our model's performance using different tolerance levels. In general, our model's semitone-based F-measures lie in between its F-measures when evaluated 20-cent resolution at ±40-cent and ±60-cent tolerance. This does not sound too surprising as a tolerance of ±50 cents would approximate a semitone; however, we would have expected our model's performance with 20-cent resolution to be somewhat better than its performance with semitone resolution, as it should reduce errors associated with vibrato that crosses a semitone boundary. This lack of improvement suggests that our model's difficulty in detecting vibrato is not due simply to semitone crossings, but rather, may be a more fundamental issue of vibrato itself.

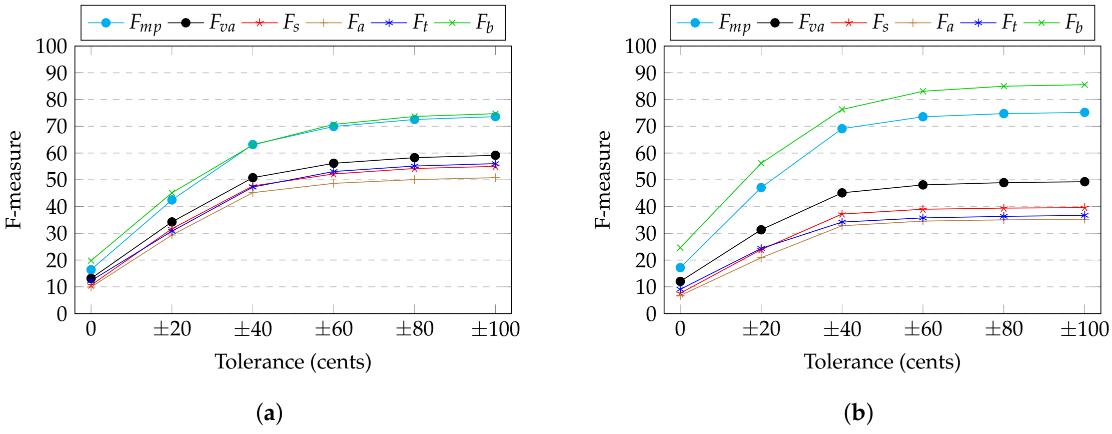


Figure 8. Our proposed model's performance on each dataset using pitch tolerance levels from zero cents up to ±100 cents. (**a**) Bach chorales; (**b**) barbershop quartets.

4. Conclusions

In this paper, we have presented a system for multi-pitch detection and voice assignment for a cappella recordings of multiple singers. It consists of two integrated components: a PLCA-based acoustic model and an HMM-based music language model. To our knowledge, ours is the first system to be designed for the task. (Supporting Material for this work is available at http://inf.ufrgs.br/~rschramm/projects/music/musingers.)

We have evaluated our system on both multi-pitch detection and voice assignment on two datasets: one of Bach chorales and another of barbershop quartets, and we achieve state-of-the-art performance on both datasets for each task. We have also shown that integrating the music language model improves multi-pitch detection performance compared to a simpler version of our system with only the acoustic model. This suggests, as has been shown in previous work, that incorporating such music language models into other acoustic music information retrieval tasks might also be of some benefit, since they can guide acoustic models using musicological principles.

For voice assignment, while our system performs well given the difficulty of the task, there is certainly room for improvement, given that the theoretical upper bound for our model is a perfect transcription if the acoustic model's $P_t(p|v)$ estimates are accurate enough. As overtones and vibrato constitute the main sources of errors in our system, reducing such errors would lead to a great improvement in the performance of our system. Thus, future work will concentrate on methods to eliminate such errors, for example by post-processing steps that examine more closely the spectral properties of detected pitches for overtone classification and the presence of vibrato. Another possible improvement could be found during the dynamic dictionary voice type assignment step. In particular, running a voice type recognition process as a preprocessing step may result in better performance.

We will also investigate the use of incorporating additional information from the acoustic model into the music language model to continue to improve performance. In particular, we currently do not use either the singer subject probabilities $P_t(s|p)$ or the vowel probabilities $P_t(o|p)$ at all, the values of which may contain useful voice separation information. Similarly, incorporating harmonic information such as chord and key information into the music language model could lead to a more informative prior for the acoustic model during integration. Additionally, learning a new dictionary for the acoustic model, for example an instrument dictionary, would allow our system to be applied to different styles of music such as instrumentals or those containing both instruments and vocals, and we intend to investigate the generality of our system in that context.

Another possible avenue for future work is the adaptation of our system to work on the note level rather than the frame level. The music language model was initially designed to do so, but the acoustic model and the integration procedure will have to be adapted as they are currently limited to working on a frame level. Such a note-based system may also eliminate the need for robust vibrato detection, as a pitch with vibrato would then correctly be classified as a single note at a single pitch. An additional benefit to adapting our system to work on the note level would be the ability to incorporate metrical or rhythmic information into the music language model.

Acknowledgments: Andrew McLeod is supported by the University of Edinburgh. Rodrigo Schramm is supported by a UK Newton Research Collaboration Programme Award (Grant No. NRCP1617/5/46). Mark Steedman is supported by EU ERC H2020 Advanced Fellowship GA 742137 SEMANTAX. Emmanouil Benetos is supported by a UK Royal Academy of Engineering Research Fellowship (Grant No. RF/128).

Author Contributions: All authors contributed to this work. Specifically, Rodrigo Schramm and Emmanouil Benetos designed the acoustic model. Andrew McLeod designed the music language model under supervision of Mark Steedman. Andrew McLeod, Rodrigo Schramm and Emmanouil Benetos designed the model integration. Andrew McLeod and Rodrigo Schramm performed the experiments.Andrew McLeod, Rodrigo Schramm and Emmanouil Benetos wrote the paper. All authors proofread the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Benetos, E.; Dixon, S.; Giannoulis, D.; Kirchhoff, H.; Klapuri, A. Automatic music transcription: Challenges and future directions. *J. Intell. Inf. Syst.* **2013**, *41*, 407–434.
2. Li, D.D.; Seung, H.S. Learning the parts of objects by non-negative matrix factorization. *Nature* **1999**, *401*, 788–791.
3. Shashanka, M.; Raj, B.; Smaragdis, P. Probabilistic latent variable models as nonnegative factorizations. *Comput. Intell. Neurosci.* **2008**, *2008*, 947438.
4. Vincent, E.; Bertin, N.; Badeau, R. Adaptive harmonic spectral decomposition for multiple pitch estimation. *IEEE Trans. Audio Speech Lang. Process.* **2010**, *18*, 528–537.
5. Kameoka, H.; Nakano, M.; Ochiai, K.; Imoto, Y.; Kashino, K.; Sagayama, S. Constrained and regularized variants of non-negative matrix factorization incorporating music-specific constraints. In Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing, Kyoto, Japan, 25–30 March 2012; pp. 5365–5368.
6. Benetos, E.; Dixon, S. Multiple-instrument polyphonic music transcription using a temporally constrained shift-invariant model. *J. Acoust. Soc. Am.* **2013**, *133*, 1727–1741.
7. Benetos, E.; Weyde, T. An efficient temporally-constrained probabilistic model for multiple-instrument music transcription. In Proceedings of the 16th International Society for Music Information Retrieval Conference, Malaga, Spain, 26–30 October 2015; pp. 701–707.
8. Fuentes, B.; Badeau, R.; Richard, G. Controlling the convergence rate to help parameter estimation in a PLCA-based model. In Proceedings of the 22nd European Signal Processing Conference, Lisbon, Portugal, 1–5 September 2014; pp. 626–630.
9. Grindlay, G.; Ellis, D.P.W. Transcribing multi-instrument polyphonic music with hierarchical eigeninstruments. *IEEE J. Sel. Top. Signal Process.* **2011**, *5*, 1159–1169.
10. Mysore, G.J.; Smaragdis, P. Relative pitch estimation of multiple instruments. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan, 19–24 April 2009; pp. 313–316.
11. Brown, J. Calculation of a constant Q spectral transform. *J. Acoust. Soc. Am.* **1991**, *89*, 425–434.
12. Benetos, E.; Dixon, S. A Shift-Invariant Latent Variable Model for Automatic Music Transcription. *Comput. Music J.* **2012**, *36*, 81–94.
13. Fuentes, B.; Badeau, R.; Richard, G. Blind Harmonic Adaptive Decomposition applied to supervised source separation. In Proceedings of the 2012 European Signal Processing Conference, Bucharest, Romania, 27–31 August 2012; pp. 2654–2658.
14. Benetos, E.; Badeau, R.; Weyde, T.; Richard, G. Template adaptation for improving automatic music transcription. In Proceedings of the 15th International Society for Music Information Retrieval Conference, Taipei, Taiwan, 27–31 October 2014; pp. 175–180.
15. O'Hanlon, K.; Nagano, H.; Keriven, N.; Plumbley, M.D. Non-negative group sparsity with subspace note modelling for polyphonic transcription. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2016**, *24*, 530–542.
16. Sigtia, S.; Benetos, E.; Dixon, S. An end-to-end neural network for polyphonic piano music transcription. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2016**, *24*, 927–939.
17. Kelz, R.; Dorfer, M.; Korzeniowski, F.; Böck, S.; Arzt, A.; Widmer, G. On the potential of simple framewise approaches to piano transcription. In Proceedings of the 17th International Society for Music Information Retrieval Conference, New York, NY, USA, 7–11 August 2016; pp. 475–481.
18. Thickstun, J.; Harchaoui, Z.; Kakade, S. Learning features of music from scratch. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
19. Bittner, R.M.; McFee, B.; Salamon, J.; Li, P.; Bello, J.P. Deep salience representations for F0 estimation in polyphonic music. In Proceedings of the 18th International Society for Music Information Retrieval Conference, Suzhou, China, 23–27 October 2017; pp. 63–70.
20. Bohak, C.; Marolt, M. Transcription of polyphonic vocal music with a repetitive melodic structure. *J. Audio Eng. Soc.* **2016**, *64*, 664–672.
21. Schramm, R.; Benetos, E. Automatic transcription of a cappella recordings from multiple singers. In Proceedings of the AES International Conference on Semantic Audio, Erlangen, Germany, 22–24 June 2017.

22. Bay, M.; Ehmann, A.F.; Beauchamp, J.W.; Smaragdis, P.; Downie, J.S. Second fiddle is important too: Pitch tracking individual voices in polyphonic music. In Proceedings of the 13th International Society for Music Information Retrieval Conference, Porto, Portugal, 8–12 October 2012; pp. 319–324.
23. Duan, Z.; Han, J.; Pardo, B. Multi-pitch streaming of harmonic sound mixtures. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2014**, *22*, 138–150.
24. Ryynanen, M.P.; Klapuri, A. Polyphonic music transcription using note event modeling. In Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY, USA, 16 October 2005; pp. 319–322.
25. Cambouropoulos, E. Voice and stream: Perceptual and computational modeling of voice separation. *Music Percept. Interdiscip. J.* **2008**, *26*, 75–94.
26. Huron, D. Tone and voice: A derivation of the rules of voice-leading from perceptual principles. *Music Percept.* **2001**, *19*, 1–64.
27. Tymoczko, D. Scale theory, serial theory and voice leading. *Music Anal.* **2008**, *27*, 1–49.
28. Temperley, D. A probabilistic model of melody perception. *Cogn. Sci.* **2008**, *32*, 418–444.
29. Karydis, I.; Nanopoulos, A.; Papadopoulos, A.; Cambouropoulos, E.; Manolopoulos, Y. Horizontal and vertical integration/segregation in auditory streaming: A voice separation algorithm for symbolic musical data. In Proceedings of the Proceedings 4th Sound and Music Computing Conference (SMC'2007), Lefkada, Greece, 11–13 July 2007; pp. 299–306.
30. Kilian, J.; Hoos, H. Voice separation-a local optimization approach. In Proceedings of the 3rd International Conference on Music Information Retrieval, Paris, France, 13–17 October 2002.
31. Kirlin, P.B.; Utgoff, P.E. VOISE: Learning to segregate voices in explicit and implicit polyphony. In Proceedings of the 6th International Conference on Music Information Retrieval, London, UK, 11–15 September 2005; pp. 552–557.
32. Guiomard-Kagan, N.; Giraud, M.; Groult, R.; Levé, F. Improving voice separation by better connecting contigs. In Proceedings of the 17th International Society for Music Information Retrieval Conference, New York, NY, USA, 7–11 August 2016; pp. 164–170.
33. Gray, P.; Bunescu, R. A neural greedy model for voice separation in symbolic music. In Proceedings of the 17th International Society for Music Information Retrieval Conference, New York, NY, USA, 7–11 August 2016; pp. 782–788.
34. Chew, E.; Wu, X. Separating voices in polyphonic music: A contig mapping approach. In Proceedings of the Computer Music Modeling and Retrieval, Esbjerg, Denmark, 26–29 May 2004; pp. 1–20.
35. de Valk, R.; Weyde, T. Bringing 'musicque into the tableture': Machine-learning models for polyphonic transcription of 16th-century lute tablature. *Early Music* **2015**, *43*, 563–576.
36. Duane, B.; Pardo, B. Streaming from MIDI using constraint satisfaction optimization and sequence alignment. In Proceedings of the International Computer Music Conference, Montreal, QC, Canada, 16–21 August 2009; pp. 1–8.
37. McLeod, A.; Steedman, M. HMM-Based Voice Separation of MIDI Performance. *J. New Music Res.* **2016**, *45*, 17–26.
38. Schramm, R.; McLeod, A.; Steedman, M.; Benetos, E. Multi-pitch detection and voice assignment for a cappella recordings of multiple singers. In Proceedings of the 18th International Society for Music Information Retrieval Conference, Suzhou, China, 23–28 October 2017; pp. 552–559.
39. Schörkhuber, C.; Klapuri, A.; Holighaus, N.; Dörfler, M. A matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution. In Proceedings of the AES 53rd Conference on Semantic Audio, London, UK, 26–29 January 2014.
40. Benetos, E.; Holzapfel, A. Automatic transcription of Turkish microtonal music. *J. Acoust. Soc. Am.* **2015**, *138*, 2118–2130.
41. Goto, M.; Hashiguchi, H.; Nishimura, T.; Oka, R. RWC music database: Music genre database and musical instrument sound database. In Proceedings of the 5th International Conference on Music Information Retrieval, Barcelona, Spain, 10–14 October 2004; pp. 229–230.
42. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc.* **1977**, *39*, 1–38.

43. Mauch, M.; Dixon, S. PYIN: A fundamental frequency estimator using probabilistic threshold distributions. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 659–663.
44. De Andrade Scatolini, C.; Richard, G.; Fuentes, B. Multipitch estimation using a PLCA-based model: Impact of partial user annotation. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Queensland, Australia, 19–24 April 2015; pp. 186–190.
45. Goto, M. A real-time music-scene-description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals. *Speech Commun.* **2004**, *43*, 311–329.
46. Kameoka, H.; Nishimoto, T.; Sagayama, S. A multipitch analyzer based on harmonic temporal structured clustering. *IEEE Trans. Audio Speech Lang. Process.* **2007**, *15*, 982–994.
47. Kirchhoff, H.; Dixon, S.; Klapuri, A. Missing template estimation for user-assisted music transcription. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, BC, Canada, 26–31 May 2013; pp. 26–30.
48. Viterbi, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theory* **1967**, *13*, 260–269.
49. Klapuri, A. Multiple fundamental frequency estimation by summing harmonic amplitudes. In Proceedings of the 7th International Conference on Music Information Retrieval, Victoria, BC, Canada, 8–12 October 2006.
50. Salamon, J.; Gomez, E. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Trans. Audio Speech Lang. Process.* **2012**, *20*, 1759–1770.
51. Pertusa, A.; Iñesta, J.M. Efficient methods for joint estimation of multiple fundamental frequencies in music signals. *EURASIP J. Adv. Signal Process.* **2012**, doi:10.1186/1687-6180-2012-27.
52. Giannoulis, D.; Benetos, E.; Klapuri, A.; Plumbley, M.D. Improving instrument recognition in polyphonic music through system integration. In Proceedings of the 39th International Conference on Acoustics, Speech and Signal Processing, Florence, Italy, 4–9 May 2014; pp. 5222–5226.
53. Bay, M.; Ehmann, A.F.; Downie, J.S. Evaluation of multiple-F0 estimation and tracking systems. In Proceedings of the 10th International Society for Music Information Retrieval Conference, Kobe, Japan, 26–30 October 2009, pp. 315–320.
54. Bogdanov, D.; Wack, N.; Gómez, E.; Gulati, S.; Herrera, P.; Mayor, O.; Roma, G.; Salamon, J.; Zapata, J.; Serra, X. Essentia: An Audio Analysis Library for Music Information Retrieval. In Proceedings of the 14th International Society for Music Information Retrieval Conference, Curitiba, Brazil, 4–8 November 2013.