MULTI-PITCH DETECTION AND VOICE ASSIGNMENT FOR *A CAPPELLA* RECORDINGS OF MULTIPLE SINGERS

Rodrigo Schramm[1,3*], **Andrew McLeod**[2*], **Mark Steedman**[2], **Emmanouil Benetos**[3]
[1] Department of Music, Universidade Federal do Rio Grande do Sul, Brazil
[2] School of Informatics, University of Edinburgh, UK
[3] Centre for Digital Music, Queen Mary University of London, UK
rschramm@ufrgs.br, A.McLeod-5@sms.ed.ac.uk, steedman@inf.ed.ac.uk,
emmanouil.benetos@qmul.ac.uk

ABSTRACT

This paper presents a multi-pitch detection and voice assignment method applied to audio recordings containing *a cappella* performances with multiple singers. A novel approach combining an acoustic model for multi-pitch detection and a music language model for voice separation and assignment is proposed. The acoustic model is a spectrogram factorization process based on Probabilistic Latent Component Analysis (PLCA), driven by a 6-dimensional dictionary with pre-learned spectral templates. The voice separation component is based on hidden Markov models that use musicological assumptions. By integrating the models, the system can detect multiple concurrent pitches in vocal music and assign each detected pitch to a specific voice corresponding to a voice type such as soprano, alto, tenor or bass (SATB). This work focuses on four-part compositions, and evaluations on recordings of Bach Chorales and Barbershop quartets show that our integrated approach achieves an F-measure of over 70% for frame-based multi-pitch detection and over 45% for four-voice assignment.

1. INTRODUCTION

Automatic music transcription is defined as the process of converting an acoustic music signal into some form of music notation [3]. In the past years, several signal processing and machine learning approaches have been proposed for automatic music transcription, with applications in music information retrieval, music education, computational musicology, and interactive music systems. A core problem of automatic transcription is multi-pitch detection, i.e. the detection of multiple concurrent pitches.

For multi-pitch detection, spectrogram factorization methods have been used extensively in the last decade [3]. However, despite promising results of template-based techniques [4, 11, 17], 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 [13, 22], feedforward, recurrent and convolutional neural networks were evaluated towards the problem of automatic piano transcription.

On approaches for automatic transcription of vocal music, Bohak and Marolt [5] proposed a method for transcribing folk music containing both instruments and vocals, which explores the repetitions of melodic segments 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 have attempted to go beyond multi-pitch detection, towards *instrument assignment* (also called timbre tracking) [1, 8, 11], where a system detects multiple pitches and assigns each pitch to a specific source that produced it. Bay et al. [1] tracked individual instruments in polyphonic instrumental music using a spectrogram factorisation approach with continuity constraints controlled by a hidden Markov model (HMM).

An emerging area of automatic music transcription attempts to combine *acoustic models* (i.e. 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. An example of such an integrated system is the work by Sigtia et al. [22] which combined neural network-based acoustic and music language models for multi-pitch detection in piano music.

Combining instrument assignment with this idea of using a music language model, it is natural to look at the field of voice separation, which is the separation of pitches into monophonic streams of notes, called voices, mainly addressed in the context of symbolic music pro-

* Authors 1 and 2 contributed equally to this work.

cessing [6, 14, 16]. Several voice separation approaches are based on voice leading rules, which were investigated in [12,23,24] from a cognitive perspective. Among the numerous rules pointed out by these authors, common characteristics are that large melodic intervals between consecutive notes within a single voice should be avoided and that two voices should not cross in pitch. A third principle suggested by [12] is the idea that the stream of notes should be relatively continuous within a single voice, and not have too many gaps of silence, ensuring temporal continuity.

The overarching aim of this work is to create a system able to detect multiple pitches in polyphonic vocal music and assign each detected pitch to a single voice of a specific voice type (e.g. soprano, alto, tenor, bass). Thus, the proposed method is able to perform both multi-pitch detection and voice assignment. 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 a music language model for voice assignment based on the HMM proposed in [16]. Although previous work has integrated musicological information for note event modelling [5, 19, 22], 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 70% and 45% for the two respective tasks.

2. PROPOSED METHOD

In this section, we present a system for multi-pitch detection and voice assignment from 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.

2.1 Acoustic Model

The acoustic model is a variant of the spectrogram factorisation-based model proposed in [21]. The model uses a fixed dictionary of log-spectral templates and aims to decompose an input time-frequency representation into several components denoting the activations of pitches, voice types, tuning deviations, singer subjects, and vowels. As time-frequency representation we use a normalised variable-Q transform (VQT) spectrogram [20] with a hop size of 20 msec and 20 cent resolution.

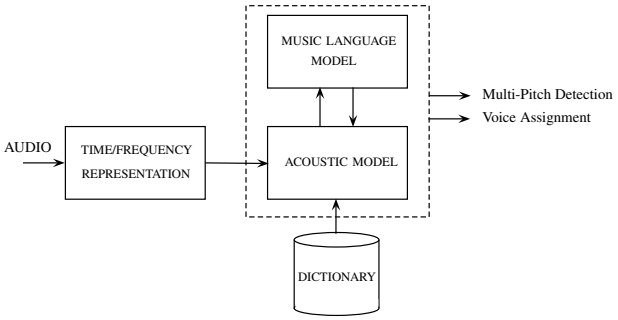


Figure 1: Proposed system diagram.

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:

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

where $P(t)$ is the spectrogram energy (known quantity). $\Phi = P(\omega|s, p, f, o, v)$ is the fixed pre-extracted spectral template dictionary (for details about the dictionary construction, refer to [21]). Variable $p \in \{21, ..., 108\}$ denotes pitch in MIDI scale, s denotes the singer index (out of the collection of singer subjects used to construct the input dictionary), o denotes the vowel type, v denotes the voice type (e.g. soprano, alto, tenor, bass), and f denotes tuning deviation from 12-tone equal temperament in 20 cent resolution ($f \in \{1, \ldots, 5\}$, with $f = 3$ denoting ideal tuning). Unlike in [21], this model decomposes the probabilities of pitch and voice type as $P(v)P_t(p|v)$. That is, $P_t(p|v)$ denotes the pitch activation for a specific voice type (eg. SATB) over time and $P(v)$ can be viewed as a mixture weight that denotes the overall contribution of each voice type to the whole input recording. 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 [1] .

The factorization can be achieved by the expectation-maximization (EM) algorithm [7], 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:

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

In the *Maximization* step, each unknown model parameter is then updated using the posterior from Eqn (2):

[1] Although $P_t(o|p)$ is not explicitly used in this proposed approach, it is kept to ensure consistency with the RWC audio dataset structure.

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

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

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

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

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

The model parameters are randomly initialised, and the EM algorithm iterates over Eqns (2)-(7). In our experiments we use 30 iterations.

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:

$$P(f',t,v) = [P(f,21,v,t)...P(f,108,v,t)] \quad (8)$$

where $f' = 1,...,880$ denotes pitch in 20 cent resolution. The overall multi-pitch detection without voice assignment, is given by $P(p,t) = \sum_v P(p,v,t)$. Finally, the voice-specific pitch activation output $P(p,v,t)$ is binarized and post-processed through a refinement step described in [21], where each pitch is aligned with the nearest peak to it in the input log-frequency spectrum.

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 approach proposed in [16], 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 multi-pitch activations $P(p,t)$, where each generates a note n with pitch $\rho(n) = p$, onset time $\delta(n) = t$, and an offset time $\tau(n) = t+1$. O_t represents this observed data at frame t.

2.2.1 HMM: State Space

In the HMM, a state S_t at frame t contains a list of M monophonic voices V_i, $1 \leq i \leq M$. The initial state S_0 contains M empty voices, and at each frame, each voice is assigned either no note, or a note with pitch $\rho(n) \in \{21,...,108\}$. Each voice contains the entire history of the notes which have been assigned to it from frame 1 to t. The state space of our model blows up exponentially (though it is reduced significantly when the model is run discriminatively as we do),, so instead of precomputed transition and emission probabilities, we 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, rather than to consider each voice to also be a list of notes. Thus, each state transition is calculated based on each voice in the previous state (though some of the probability calculations require knowledge of individual notes).

2.2.2 HMM: 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 be transformed into the corresponding $V_i \in S_t$ by assigning to it up to 1 note with onset time 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 those notes n contained by any voice in S_t where $\delta(n) = t$, and W_t is a list of integers, each representing the voice assignment index for a single note $n \in N_t$. For each index i, $1 \leq i \leq |N_t| = |W_t|$, note n_i is assigned to voice $V_{w_i} \in S_t$. Here, N_t only contains those observed notes which are assigned to a voice in S_t, not all observed notes. Since all of our voices are monophonic, no two elements in W_t may be equal.

We now define the HMM transition probability $\mathrm{P}(S_t|S_{t-1})$ as $\mathrm{P}(T_{S_{t-1},N_t,W_t})$:

$$P(T_{S_{t-1},N_t,W_t}) = \Psi(W_t)\prod_{1\leq i\leq |N_t|}\Theta(S_{t-1},n_i,w_i)\Lambda(V_{w_i},n_i). \quad (9)$$

The first term in this product is defined as

$$\Psi(W) = \prod_{1\leq j\leq M}\begin{cases}\Upsilon & j \in W \\ 1-\Upsilon & j \notin W\end{cases} \quad (10)$$

where the parameter Υ represents the probability that a given voice contains any note in a frame.

$\Theta(S_{t-1},n,w)$ is a penalty function used to minimize the voice crossings. It returns by default 1, but its output is multiplied by a parameter θ—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 $\chi(V_{w-1}) > \rho(n)$
2. $w < |M|$ and $\chi(V_{w+1}) < \rho(n)$

$\chi(V)$ represents the pitch of a voice, calculated as a weighted sum of the pitches of its most recent notes. Cases 1 and 2 apply when a note is out of pitch order with the preceding or succeeding voice in the state respectively.

$\Lambda(V,n)$ is used to calculate the probability of a note n being assigned to a voice V, and is the product of a pitch score Δ_p and a gap score Δ_g:

$$\Lambda(V,n) = \Delta_p(V,n)\Delta_g(V,n) \quad (11)$$

The pitch score, used to minimise melodic jumps within a voice, is computed as shown in Eqn (12), where $\mathcal{N}(\mu,\sigma)$

represents a normal distribution with mean μ and standard deviation σ, and σ_p is a parameter. The gap score is used to prefer temporal continuity within a voice, and is computed using Eqn (13), where $\tau(V)$ 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 1 if V is empty.

$$\Delta_p(V, n) = \mathcal{N}(\rho(n) - \chi(V), \sigma_p) \tag{12}$$

$$\Delta_g(V, n) = \max\left(\ln\left(-\frac{\delta(n) - \tau(V)}{\sigma_g} + 1\right) + 1, g_{min}\right) \tag{13}$$

2.2.3 HMM: Emission Function

A state S_t emits a set of notes containing only those which have an onset at frame t, and a state containing a voice with a note at frame t must emit that note. The probability of a state S_t emitting the note set O_t is shown in Eqn (14), using the voice posterior $P_t(v|p)$ from the acoustic model.

$$\mathrm{P}(O_t|S_t) = \prod_{n \in O_t} \begin{cases} P_t(v = i|p = \rho(n)) & n \in V_i \in S_t \\ 1 & \text{otherwise} \end{cases} \tag{14}$$

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, we allow a state 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 Eqn (10).

2.2.4 HMM: Inference

To find the most likely final state given our observed note sets, we use the Viterbi algorithm [26] 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.

2.3 Model Integration

In this section, we describe the integration of the acoustic model and the music language model into a single system which jointly performs multi-pitch detection and voice assignment from audio. This integration is done in two stages. First, using only the acoustic model from subsection 2.1, the EM algorithm is run for 15 iterations, when the multi-pitch detections converge. Next, the system runs for 15 more EM iterations, this time also using the music language model from subsection 2.2. In each iteration, the acoustic model is run first, and then the language model is run on the resulting multi-pitch detections. To intergrate the two models, we apply a fusion mechanism inspired by the one used in [9] 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, Eqn (6) is modified as:

$$P_t^{new}(p|v) = \alpha P_t(p|v) + (1 - \alpha)\phi_t(p|v), \tag{15}$$

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

$$\phi_t(p|v) \propto P_t^a(p|v)P_t(p|v). \tag{16}$$

$P_t^a(p|v)$ 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 Eqn (12). 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 1 when the V is an empty voice (thus becoming a uniform distribution over all pitches). The hyperparameter of Eqn (16) acts as a soft mask, reweighing the pitch contribution of each voice regarding only the pitch neighbourhood previously detected by the model.

The final output of the integrated system is a list of the detected pitches at each time frame which are assigned to a voice in the most probable final HMM state $S_{t_{max}}$, along with the voice assignment for each. Figure 2 shows an example output of the integrated system.

3. EVALUATION

3.1 Datasets

We evaluate the proposed model on two datasets of *a capella* recordings [2] : one of 26 Bach Chorales and another of 22 Barbershop quartets, in total 104 minutes. 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 [15] on each individual monophonic track. Experiments are conducted using the mix down of each audio file (i.e. 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 MIREX multiple-F0 estimation evaluations [2], 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.

[2] Original recordings available at `http://www.pgmusic.com/{bachchorales.htm|barbershopquartet.htm}`.

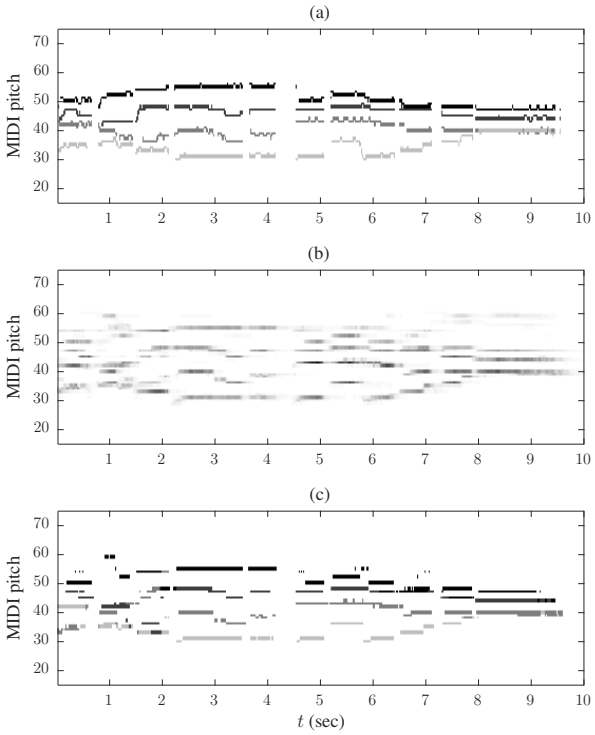


Figure 2: Multi-pitch detection and voice assignment for a 10-sec excerpt of "O Sacred Head Sore Wounded" from the Bach Chorales dataset. Each vocal part is shown in a distinct shade of grey. (a) Ground truth. (b) Pitch activation $P(p, t)$. (c) Output of the integrated system.

3.3 Training

To train the acoustic model, we use recordings from the RWC dataset [10] to generate the 6-dimensional dictionary of log-spectral templates specified in Section 2.1, following the procedure described in [21]. The recordings used to generate the dictionary contain sequences of notes following a chromatic scale, in five distinct English vowels (/a/, /æ/, /i/, /ɒ/, /u/). The dictionary contains templates generated from 15 distinct singers (9 male and 6 female, consisting of 3 human subjects for each voice type: bass, baritone, tenor, alto, soprano).

For all parameters in the music language model, we use the values reported in [16] that were used for voice separation in fugues. We also introduce two new parameters to the system: the voice order probability θ and the voice probability Υ. We use MIDI files of 50 Bach Chorales [3] (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 $\theta = 0.006$ and $\Upsilon = 0.99$, 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 dataset. This results in a value of 0.6 when trained on the Chorale recordings and 0.3 when trained on the Barbershop recordings. To avoid overfitting, we employ cross-validation, using the α value that maximises the Chorales' F_{va} when evaluating the Barbershop quartets, and vice versa.

[3] MIDI files available at `http://kern.ccarh.org/`.

3.4 Results

We compare our model's multi-pitch detection results with those of three baseline methods: VINC+ [25], which uses an adaptive spectral decomposition based on unsupervised NMF; PERT+ [18], which selects candidates among spectral peaks, validating candidates through additional audio descriptors; and MSINGERS†+ [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. To the authors' knowledge, there is no existing system for multi-pitch detection and voice assignment that can be used as a baseline for our model's voice assignment. However, for the sake of comparison, we include results from voice assignments derived from the model proposed in [21], which we call MSINGERS-VA, despite the fact that the original model was not designed for the task.

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 1, it can be seen that MSINGERS†+ achieves the highest F_{mp} on the Bach chorales, narrowly edging out VOCAL4-VA, but VOCAL4-VA achieves state-of-the-art results on the Barbershop quartets. In both datasets, VOCAL4-VA outperforms VOCAL4-MP substantially, indicating that the music language model is able to drive the acoustic model to a more meaningful factorisation. The voice assignment results are shown in Table 2, where it is clear that VOCAL4-VA outperforms the other models, suggesting that perhaps a language model is almost necessary for the task. Also interesting to note is that it performs significantly better on the bass voice than on the other voices. 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 3, 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 continues to improve until it finally converges after iteration 30.

The voice assignment results, however, are less straightforward. After the significant improvement on the 15th iteration, the results either remain relatively stable (in the Barbershop quartets) or even drop slightly (in the Bach

Model	Bach Chorales	Barbershop Quartets
VINC+	53.58	51.04
PERT+	67.19	63.85
MSINGERS†+	**70.84**	71.03
VOCAL4-MP	63.05	59.09
VOCAL4-VA	69.66	**73.46**

Table 1: Multi-pitch detection results.

Model	Bach Chorales				
	F_{va}	F_s	F_a	F_t	F_b
MSINGERS-VA	18.02	15.37	17.59	26.32	12.81
VOCAL4-MP	21.84	12.99	10.27	22.72	41.37
VOCAL4-VA	45.31	26.07	37.63	49.61	67.94
Model	Barbershop Quartets				
	F_{va}	F_s	F_a	F_t	F_b
MSINGERS-VA	12.29	9.70	14.03	27.93	9.48
VOCAL4-MP	18.35	2.40	10.56	16.61	43.85
VOCAL4-VA	46.92	40.01	35.57	29.76	82.34

Table 2: Voice assignment results.

chorales) before convergence. This slight drop is due to the fact that the language model initially receives noisy multi-pitch detections that include false positives (mainly overtones). Incorporating these overtones into the voice assignment can cause the removal of correct pitch detections, which in turn reduces the voice assignment F-measures.

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. 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. An example of both of these types of errors can be found in Figure 4.

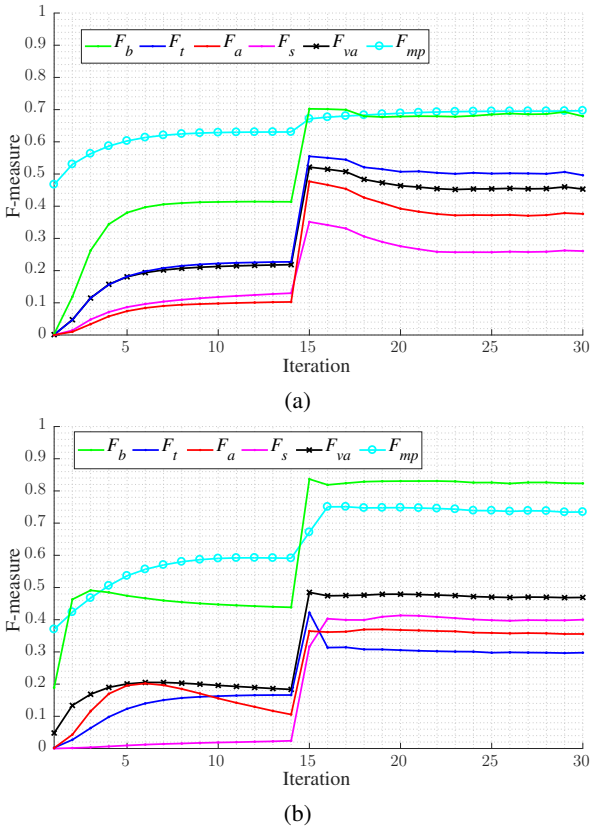
4. CONCLUSION

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 [4].

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. Our model outperforms baseline multi-pitch detection systems on the Barbershop quartets, and achieves near state-of-the-art performance on the chorales. We have shown that integrating the music language model improves multi-pitch detection performance compared with 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 MIR tasks might also be of some benefit, since they can guide acoustic models using musicological principles.

We also presented results for voice assignment, and show that while our model performs well given the difficulty of the task, there is certainly room for improvement. Avenues for future work include a better handling of overtones in the acoustic model, and better recognition of vibrato in both the acoustic and the music language model. We will also investigate the use of timbral information for further improving voice assignment performance. Additionally, our model could be applied to different styles of music (e.g., instrumental, or those containing both instruments and vocals) by learning a new dictionary for the acoustic model and retraining the parameters of the music language model, and we intend to investigate the generality of our model in that context.

[4] Supporting material for this work is available at `http://inf.ufrgs.br/~rschramm/projects/msingers`



(a)

(b)

Figure 3: The VOCAL4-VA model's F-measures after each EM iteration, averaged across all songs in each dataset: (a) Bach Chorales. (b) Barbershop Quartets.

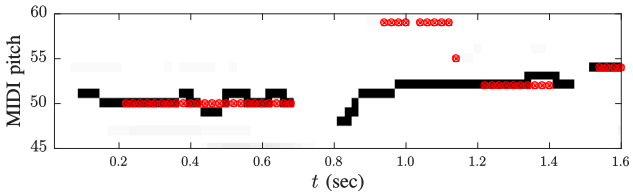


Figure 4: Pitch detections (red) and ground truth (black) for the soprano voice at the beginning of the excerpt from Figure 2, showing errors from both overtones and vibrato.

5. ACKNOWLEDGEMENT

RS is supported by a UK Newton Research Collaboration Programme Award (grant no. NRCP1617/5/46). AM and MS are supported by a gift from the 2017 Bloomberg Data Science Research Grant program and EU ERC H2020 Advanced Fellowship GA 742137 SEMANTAX. EB is supported by a UK Royal Academy of Engineering Research Fellowship (grant no. RF/128).

6. REFERENCES

[1] M. Bay, A. F. Ehmann, J. W. Beauchamp, P. Smaragdis, and J. Stephen Downie. Second fiddle is important too: Pitch tracking individual voices in polyphonic music. In *ISMIR*, pages 319–324, 2012.

[2] M. Bay, A. F. Ehmann, and J. S. Downie. Evaluation of multiple-F0 estimation and tracking systems. In *ISMIR*, pages 315–320, October 26-30 2009.

[3] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri. Automatic music transcription: challenges and future directions. *J. Intell. Inf. Syst.*, 41(3):407–434, 2013.

[4] E. Benetos and T. Weyde. An efficient temporally-constrained probabilistic model for multiple-instrument music transcription. In *ISMIR*, pages 701–707, 2015.

[5] C. Bohak and M. Marolt. Transcription of polyphonic vocal music with a repetitive melodic structure. *J. Audio Eng. Soc*, 64(9):664–672, 2016.

[6] E. Cambouropoulos. Voice and stream: Perceptual and computational modeling of voice separation. *Music Perception: An Interdisciplinary Journal*, 26(1):75–94, 2008.

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. Royal Statistical Society*, 39(1):1–38, 1977.

[8] Z. Duan, J. Han, and B. Pardo. Multi-pitch streaming of harmonic sound mixtures. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(1):138–150, Jan 2014.

[9] D. Giannoulis, E. Benetos, A. Klapuri, and M. D. Plumbley. Improving instrument recognition in polyphonic music through system integration. In *ICASSP*, pages 5222–5226, 2014.

[10] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC music database: Music genre database and musical instrument sound database. In *ISMIR*, pages 229–230, 2004.

[11] G. Grindlay and D. P. W. Ellis. Transcribing multi-instrument polyphonic music with hierarchical eigeninstruments. *IEEE J. Selected Topics in Signal Processing*, 5(6):1159–1169, October 2011.

[12] D. Huron. Tone and voice: A derivation of the rules of voice-leading from perceptual principles. *Music Perception*, 19(1):1–64, 2001.

[13] R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt, and G. Widmer. On the potential of simple framewise approaches to piano transcription. In *ISMIR*, pages 475–481, 2016.

[14] P. B. Kirlin and P. E. Utgoff. VOISE: learning to segregate voices in explicit and implicit polyphony. In *ISMIR*, pages 552–557, 2005.

[15] M. Mauch and S. Dixon. pYIN: A fundamental frequency estimator using probabilistic threshold distributions. In *ICASSP*, pages 659–663, 2014.

[16] A. McLeod and M. Steedman. HMM-based voice separation of MIDI performance. *Journal of New Music Research*, 45(1):17–26, 2016.

[17] G. J. Mysore and P. Smaragdis. Relative pitch estimation of multiple instruments. In *ICASSP*, pages 313–316, 2009.

[18] A. Pertusa and J. M. Iñesta. Efficient methods for joint estimation of multiple fundamental frequencies in music signals. *EURASIP Journal on Advances in Signal Processing*, 2012.

[19] M. P. Ryynanen and A. Klapuri. Polyphonic music transcription using note event modeling. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005.*, pages 319–322, Oct 2005.

[20] C. Schörkhuber, A. Klapuri, N. Holighaus, and M. Dörfler. A Matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution. In *AES 53rd Conference on Semantic Audio*, January 2014.

[21] R. Schramm and E. Benetos. Automatic transcription of a cappella recordings from multiple singers. In *AES International Conference on Semantic Audio*, June 2017.

[22] S. Sigtia, E. Benetos, and S. Dixon. An end-to-end neural network for polyphonic piano music transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(5):927–939, May 2016.

[23] D. Temperley. A probabilistic model of melody perception. *Cognitive Science*, 32(2):418–444, 2008.

[24] D. Tymoczko. Scale theory, serial theory and voice leading. *Music Analysis*, 27(1):1–49, 2008.

[25] E. Vincent, N. Bertin, and R. Badeau. Adaptive harmonic spectral decomposition for multiple pitch estimation. *IEEE Trans. Audio, Speech, and Lang. Processing*, 18(3):528–537, March 2010.

[26] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269, 1967.