SECOND FIDDLE IS IMPORTANT TOO: PITCH TRACKING INDIVIDUAL VOICES IN POLYPHONIC MUSIC

Mert Bay[2], Andreas F. Ehmann[2], James W. Beauchamp[2], Paris Smaragdis[1,2] and J. Stephen Downie[3]

[1]Department of Computer Science, University of Illinois at Urbana-Champaign
[2]Department of Electrical & Computer Eng., University of Illinois at Urbana-Champaign
[3]Graduate School of Library and Information Science, University of Illinois at Urbana-Champaign
{*mertbay,aehmann,jwbeauch,paris,jdownie*}*@illinois.edu*

ABSTRACT

Recently, there has been much interest in automatic pitch estimation and note tracking of polyphonic music. To date, however, most techniques produce a representation where pitch estimates are not associated with any particular instrument or voice. Therefore, the actual tracks for each instrument are not readily accessible. Access to individual tracks is needed for more complete music transcription and additionally will provide a window to the analysis of higher constructs such as counterpoint and instrument theme imitation during a composition. In this paper, we present a method for tracking the pitches (F0s) of individual instruments in polyphonic music. The system uses a pre-learned dictionary of spectral basis vectors for each note for a variety of musical instruments. The method then formulates the tracking of pitches of individual voices in a probabilistic manner by attempting to explain the input spectrum as the most likely combination of musical instruments and notes drawn from the dictionary. The method has been evaluated on a subset of the MIREX multiple-F0 estimation test dataset, showing promising results.

1. INTRODUCTION

One of the most important classes of information to be retrieved from music is its polyphonic pitch content. Recently, many researchers have attempted multiple-F0 estimation (MFE) [11, 14, 18]. (A review of state-of-the-art MFE systems can be found in [2].) Many current MFE systems such as [14] restrict themselves to the estimation of a certain number of F0s for each frame, while ignoring which instrument/timbre corresponds to each F0. While approaches for tracking solo melodic voice/timbres exist [15], the pitch tracking of additional lines and parts in polyphonic music opens new possibilities. By exposing the individual lines produced by each instrument in polyphonic music, aspects such as counterpoint, the appearance of *leitmotifs* across instruments in a piece, and hidden thematic references across musical pieces can be uncovered. Therefore, instruments and their timbres are very important components in music and should be tracked along with their F0s. Moreover, knowing each instrument's F0 track can be very beneficial for a variety of MIR user applications, such as music transcription, score alignment, audio music similarity, cover song identification, active music listening, melodic similarity, harmonic analysis, intelligent equalization, and F0-guided source separation.

As stated previously, most MFE systems produce low-level representations of the polyphonic pitch content present in music audio which report only what fundamental frequencies are present at each given time. However higher-level representations can attempt to track and link these fundamental frequencies over time to form notes such as described in [3]. It is important to note that such tracking can also improve the accuracy of MFE's based purely on individual frames, as reported reported in [18]. In [12] a classification approach is used to determine singing voice portions in music audio so as to build the pitch-track corresponding to a vocal melody. In [5, 6], a frame level multi-F0 estimation method is used followed by a constrained clustering method that uses harmonic amplitude-based features to cluster pitches into pitch tracks. In these cases, to build instrument or timbre-specific pitch tracks, bottom-up methods were used that first produced frame-based pitch estimates and subsequently sometimes attempted to build note-level representations, which may form solo phrases.

In the method presented in this paper, we use an alternative approach. Instead of building timbre tracks from the pitch content, our proposed approach uses timbre information to guide the formation of instrument-specific pitch tracks. This paper is organized as follows. Section 2 details the proposed method. Section 3 describes the evaluation datasets, measures, and results. The evaluation results are discussed in Section 4 and conclusions and future work covered in Section 5.

2. PROPOSED METHOD

To make a system that tracks pitches attributed to different musical instruments, we borrow an idea from the supervised sound source separation domain: using a spectral library [1, 13]. By training our system on example sounds from different instruments, we can track them in complex sound mixtures. The proposed method is based on probabilistic latent component analysis (PLCA) [17]. PLCA is a variant of non-negative matrix factorization (NMF) and has been used widely to model sounds in the spectral domain. Its probabilistic interpretation makes it extensible to using priors and statistical techniques. We use regular PLCA to build dictionaries of spectra indexed by F0 and instrument where the spectra are analyzed from recordings of individual notes of different musical instruments. We extend the model of [16] to represent each source/instrument not by only one spectral dictionary, but rather with a collection of dictionaries, each of which is an ensemble of spectral basis vectors that have the same F0. We model the input music signal's spectrum as a sum of basis vectors from F0-specfic spectrum dictionaries for different instruments. Update rules are designated to calculate the model parameters, which are estimated probabilities for the occurrence of each spectral basis vector, F0-spectrum dictionary, and instrument in the input mixture at a given time. Finally, we perform the Viterbi algorithm [8] to track the most likely pitch sequence for each instrument.

A sinusoidal model is used for the time-frequency representation because of its compactness and also for representing each note independent of the intonation errors or tuning differences between training and test set performers. We begin by performing a short-time Fourier transform on the audio signal. Peaks in the spectrum are then determined using a frequency-dependent threshold as described in [7]. We then refine each peak's amplitude and frequency using a signal derivative method proposed by [4].

2.1 Model

We model the audio input mixture's spectrum for each frame as a sum of instrument tones where each tone is represented by a dictionary of spectral basis vectors that are learned in advance.

It is assumed that all instrument tones have harmonically related frequencies which are integer multiples of an F0 frequency. It turns out that in a mixture of such tones there is a high probability that harmonics will overlap. The input spectrum can be modeled as a distribution/histogram over a range of frequencies. E.g., each component is viewed as the probability of occurrence of that particular frequency. The magnitude at any particular frequency can be thought as an accumulation of magnitudes from various instruments due to component overlapping. The scaled version of the input mixture spectrum is modeled as a discrete distribution. The generative process is modeled as follows:

$$X_t(f) \cong P_t(f) = \sum_{i}^{I} P_t(i) \sum_{n \in p_i}^{N} P_t(p|i) \sum_{z \in z_{p_i}}^{K} P_t(z|p) P_{p_i}(f|z) \quad (1)$$

where $X_t(f_j)$ is the spectral magnitude of the peak j at frequency f_j for the observed input mixture spectrum at time t; $P_t(f)$ is an approximation of the input spectrum; $P_t(i)$ is the estimated probability of occurrence of instrument i at time t, whereas $P_t(p|i)$ is the estimated probability that pitch p is produced by instrument i; $P(f|z)$ is the learned spectral basis vector for the pitch p of instrument i; and $P_t(z|p)$ is the probability (weight) of that basis vector. The above model explains the mixture magnitude spectrum hierarchically as the sum of N individual pitches from I different instruments where the dictionary corresponding to each pitch/instrument has K elements. The independence relationships of the model can be represented by the graph I→P→Z→F. The process that generates the frequency components in the observed magnitude spectrum is as follows: First, an individual instrument library is selected from a group of instrument libraries. Second, a spectrum dictionary is drawn for each pitch from the instrument library. Third, a spectral basis vector is drawn from a particular F0-spectrum dictionary for the instrument. Fourth, a spectral component at a particular frequency is drawn from the spectral basis vector. Although it is possible that this spectral component will be generated by only a single instrument, most often it only contributes a fraction to the magnitude of the observed spectrum at that frequency.

2.2 Parameter Estimation

Parameters for the model $\theta = \{P_t(z|p), P_t(p|i), P_t(i)\}$ can be estimated using an expectation-maximization (EM) algorithm. In the E-step, current parameter values θ^{old} are used to calculate the posterior distribution.

$$P_t(i,p,z|f,\theta^{old}) = \frac{P_t(f|i,p,z)P_t(i,p,z)}{P_t(f)} \quad (2)$$

Because of the structure of the model, $P_t(f|i,p,z) = P_t(f|z)$ and $P_t(i,p,z) = P_t(i)P_t(p|i)P_t(z|p)$. We can write the posterior as

$$P_t(i,p,z|f,\theta^{old}) = \frac{P(f|z)P_t(z|p)P_t(p|i)P_t(i)}{\sum_i P_t(i) \sum_{n \in p_i} P_t(p|i) \sum_{z \in z_{p_i}} P_t(z|p)P_{p_i}(f|z)} \quad (3)$$

The posterior is used to calculate the expectation of the complete data log likelihood Q

$$Q(\theta,\theta^{old}) = \sum_f X_f \sum_i \sum_{n \in p_i} \sum_{z \in z_{p_i}} P(i,p,z|f,\theta^{old}) log(P(i,p,z,f|\theta) \quad (4)$$

In the M-step, new parameters are estimated by maximizing the above function according to θ, resulting in the following update rules:

$$P_t(z|p)^{new} \leftarrow \frac{\sum_f P_t(i,p,z|f)X_t(f)}{\sum_{z \in z_{p_i}} \sum_f P_t(i,p,z|f)X_t(f)} \quad (5)$$

$$P_t(p|i)^{new} \leftarrow \frac{\sum_{z \in z_{p_i}} \sum_f P_t(i,p,z|f)X_t(f)}{\sum_{p \in p_i} \sum_{z \in z_{p_i}} \sum_f P_t(i,p,z|f)X_t(f)} \quad (6)$$

$$P_t(i)^{new} \leftarrow \frac{\sum_{p \in p_i} \sum_{z \in z_{p_i}} \sum_f P_t(i,p,z|f)X_t(f)}{\sum_i \sum_{p \in p_i} \sum_{z \in z_{p_i}} \sum_f P_t(i,p,z|f)X_t(f)} \quad (7)$$

We then use the new estimates to calculate the posterior in an iterative manner and repeat until convergence is achieved.

2.3 Sparsity Prior

At any given time, we expect each instrument active to be playing a single pitch. Even though the parameter estimation method would allow multiple pitches per frame for each instrument, in this project our goal is to track a monophonic pitch contour for each instrument. We also expect that not all instruments are active at the same time.

Enforcing sparsity constraints on note and instrument probabilities $P_t(p|i)$'s and $Pt(i)$'s reinforce this behavior. We use the following prior (the normalizing constant is dropped for convenience)

$$P(\phi) = \left(\sum_{\phi} (P(\phi))^{\alpha} \right)^{\beta} \tag{8}$$

Adding the above prior to the expectation of the complete data log-likelihood and maximizing it with respect to $P(i)$ and $P(p|i)$, we arrive at the following update rules

$$P(p|i)^{new} \leftarrow \sum_{z \in z_{p_i}} \sum_{f} P(i,p,z|f) X_f + \frac{\beta p (P_t(p|i))^{\alpha}}{\sum\limits_{p \in p_i} (P_t(p|i))^{\alpha}} \tag{9}$$

$$P_t(i)^{new} \leftarrow \sum_{p \in p_i} \sum_{z \in z_{p_i}} \sum_{f} P_t(i,p,z|f) X_t(f) + \frac{\beta p (P_t(i))^{\alpha}}{\sum\limits_{p \in i} (P_t(i))^{\alpha}} \tag{10}$$

(We have to rescale explicitly so that it sums up to one.)

$$P(p|i)^{new} \leftarrow \frac{P(p|i)^{new}}{\sum\limits_{p \in p_i} P(i)^{new}} \text{ and } P(i)^{new} \leftarrow \frac{P(i)^{new}}{\sum\limits_{i} P(i)^{new}} \tag{11}$$

2.4 Enforcing Continuity

Our goal is to track each instrument's F0 through time. We expect the pitch contour to be smooth, not changing drastically from frame to frame except at note transitions. Using the Viterbi algorithm, we treat the pitches as hidden states and pose the instrument tracking problem as one of inferring the most likely pitch state sequence for each instrument. Above estimated pitch distributions (which maximize the mixture likelihood $P(X_t(f)|i,p,z))$ for each instrument are considered to be the emission probability of the hidden state in a hidden Markov model (HMM). Transition probabilities are modeled as normal distributions given by

$$P(p_t|p^{'}_{t-1}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\left(f_{0\,t} - f^{'}_{0\,t-1}\right)^2}{2\sigma^2}} \tag{12}$$

where p_t denotes the hidden pitch state for instrument i at time t. f_{0t} denotes the F0 associated with the hidden pitch state for instrument i at time t. Transitions are calculated within the same instrument. We empirically choose $\sigma = 7 + f/100Hz$. The above distribution enforces the continuity of the active notes from frame to frame.

3. EVALUATION ON REAL WORLD DATA

We trained a dictionary for each pitch of each instrument using the RWC musical instrument database [9] using non-negative matrix factorization with Kullback-Leibler divergence which is numerically equal to the regular PLCA method in 2 dimensions [17]. For each pitch from the RWC dataset, 20 representative spectra were derived from 27 different tones corresponding to 3 players, 3 dynamics (piano, mezzo, forte), and 3 articulations (normal, staccato, vibrato). In the pitch tracking stage, we limit the number of instrument libraries to choose from by designating the instruments expected to be in the input mixture as input to the algorithm. We also limit the search range for F0 (which F0-spectrum dictionaries to use) of each instrument by only using the peaks estimated by the sinusoidal model that are between 50 Hz to 2500 Hz as pitch candidates.

Evaluations are performed at frame level. Multi-F0 tracking problem is similar to melody extraction problem extended to multiple melodies as opposed multi-F0 estimation problem where the estimated number of F0s for each frame can be different than the ground-truth F0s, which is not the case in the tracking problem, where the number of estimated F0s and the reference F0s are simply equal to total number of frames. That's why we decided to use the evaluation metrics from MIREX melody extraction task. Comparing the voiced (nonzero F0) and unvoiced (zero F0) values for each frame of the estimated and ground-truth F0 tracks, the counts for true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) are calculated according to Table 1

		Estimated	
		voiced	unvoiced
Ground	voiced	TP	FN
truth	unvoiced	FP	TN

Table 1. Evaluation

TP's further break down into ones with correct F0 and the ones with incorrect F0 as $TP = TPC + TPI$. Estimated voiced F0 is correct if it is within a quarter tone (+-2.93%) range of a positive ground-truth F0 for that frame.

The precision, recall, F-measure, and accuracy for each input test file is then calculated over all frames and all instruments as:

$$\text{Precision} = \frac{\sum\limits_{i} \sum\limits_{t} TPC_{i,t}}{\sum\limits_{i} \sum\limits_{t} TP_{i,t} + FP_{i,t}} \tag{13}$$

$$\text{Recall} = \frac{\sum\limits_{i} \sum\limits_{t} TPC_{i,t}}{\sum\limits_{i} \sum\limits_{t} TP_{i,t} + FN_{i,t}} \tag{14}$$

$$\text{F-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{15}$$

$$\text{Acc.} = \frac{\sum\limits_{i} \sum\limits_{t} TPC_{i,t} + TN_{i,t}}{\sum\limits_{i} \sum\limits_{t} TP_{i,t} + FP_{i,t} + TN_{i,t} + FN_{i,t}} \tag{16}$$

where t is the frame index and i is the instrument index. We tested the proposed method on different datasets. The ground-truths for these datasets were estimated using monophonic pitch estimators (*Wavesurfer, Praat and YIN*) on the single-instrument recordings prior to mixing. The results of the monophonic pitch estimators are manually corrected where necessary.

For preliminary testing and development, we applied the method on a 11 second excerpt taken from a real world performance by bassoon, clarinet and oboe which was taken from a MIREX multitrack dataset (standard woodwind quintet transcribed from L. van Beethoven ”Variations for String Quartet”, Op.18, N.5). The three separate tracks were mixed to monaural. The results can be seen in Table 2. The proposed method scored 0.83 accuracy on average. Figure 1 shows the multiple-F0 tracks for each instrument. Without tracking, the F0’s would not be connected from frame to frame.

	Bassoon	Clarinet	Oboe	Ave.
Accuracy	0.76	0.85	0.89	0.83
Precision	0.76	0.85	0.89	0.83
Recall	0.82	0.92	0.90	0.88
F-measure	0.79	0.88	0.89	0.86

Table 2. Evaluation performances for a 11-s woodwind trio excerpt (bassoon, clarinet, oboe).

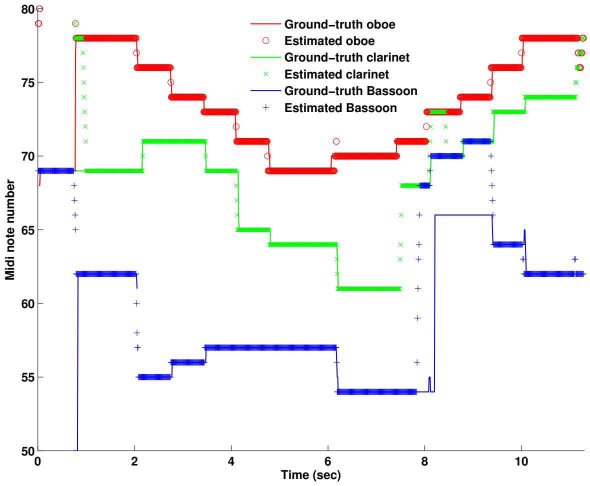

Figure 1. Pitch (in midi note numbers) vs. time using the proposed system on the 11-s woodwind trio excerpt (bassoon (lower), clarinet (middle), oboe (upper)). Thin lines represent the ground-truth.

We then tested the proposed method on two datasets used in the MIREX multiple-F0 task test set [2]. The first one is a multitrack recording of the woodwind quintet mentioned above [2]. The piece is highly contrapuntal as opposed to consisting of a lone melodic voice plus accompaniment. The predominant melodies alternate between instruments. The F0 tracks often cross each other. Five non-overlapping 30-second sections were chosen from the recording. Isolated instruments were mixed from solo tracks to polyphonies ranging from 2 (duo) to 5 (quintet), resulting in a total of 20 tracks (4 different polyphonies times five sections). The average pitch-tracking results over all tracks for polyphony 2 to 5 are shown in Table 3.

Accuracy	Precision	Recall	F-measure
0.52	0.50	0.56	0.53

Table 3. Average performance on the MIREX woodwind quintet dataset.

Figure 2 shows a bar graph of the performance of the method for different polyphonies. Figure 3 shows the average accuracies for different instruments in various polyphonies.

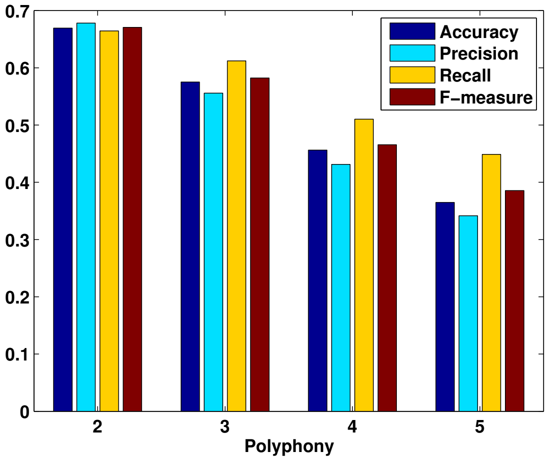

Figure 2. Performance vs. polyphony for the MIREX woodwind quintet dataset (Five 30-s segments per polyphony).

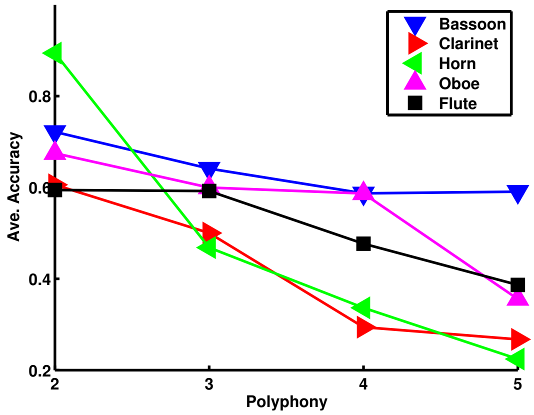

Figure 3. Ave. accuracy of individual instruments for different polyphonies for the MIREX woodwind quintet dataset. (Five 30-s segments)

The second dataset we tested our method on is a recording of a four-part J.S. Bach chorales created by [5] consisting of bassoon, clarinet, saxophone, and viola. Four 30 seconds sections were mixed from duo to quartet, resulting in 12 tracks (2 different polyphonies times 4 sections). The average results over all tracks in this dataset can be seen in Table 4.

Accuracy	Precision	Recall	F-measure
0.59	0.55	0.55	0.55

Table 4. Average performance for the MIREX Bach chorales dataset

Figure 4 shows a bar graph of the performance of the method for different polyphonies. Figure 5 shows the average accuracies of different instruments in various polyphonies. The RWC dataset of the MIREX multiple-F0 task was not evaluated because RWC samples were used for training. Also the piano dataset was not used because our project's goal is to track distinct timbres.

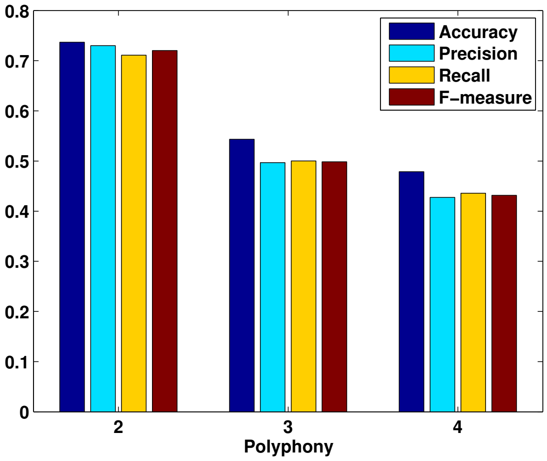

Figure 4. Ave. Performance vs. polyphony for the MIREX Bach chorales dataset. (Four 30-s segments)

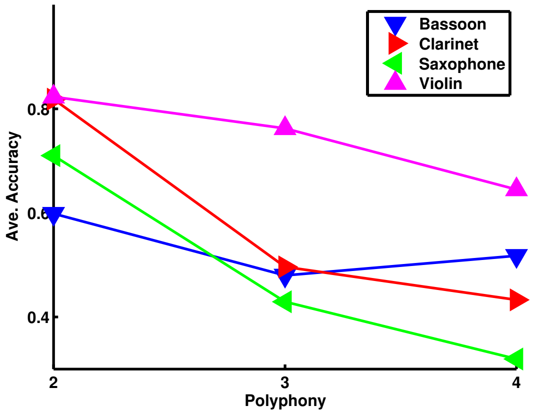

Figure 5. Ave. accuracy of instruments in different polyphonies for the MIREX Bach chorales quintet dataset.(Four 30-s segments)

4. DISCUSSION

The proposed method on average performed with 84% accuracy on identifying the instrument tracks for the trio case (Figure 1) which was used for the development of the algorithm. Accuracies were 52% and 59% for the MIREX woodwind quintet and Bach chorales quartet datasets. By examining the MIREX dataset results, we see that most problems are caused by an inactive instrument following the dominant instrument's F0 track. Some instruments in this dataset are inactive during 70-80% of the entire duration of the input. Accuracy-vs.-instrument results for the MIREX woodwind quintet (Figure 3), indicate that instruments horn and clarinet have lower performance in the 4 and 5 polyphony cases, due to their F0 tracks being highly sparse. In addition to remaining silent much of the time, they often play soft notes when they are active.

The tracking system reports note probabilities for every non-silent frame, which are then used in an HMM to estimate the F0 tracks. Voicing decisions are based strictly on the rms amplitude of the mixture input signal to decide whether the signal is silent. This results in a high number of false positives when an individual instrument is silent in parts of a track, a condition which happens often in the MIREX dataset. This behavior also results in a very low number of true and false negatives (since the system reports very few negatives) resulting in a higher recall than lower precision which can be seen in Figures 2 and 4. In the future, we would like to explore methods to infer whether instruments are active or not at any given point in time.

Another kind of error that frequently occurs is when a less dominant instrument tracks a more dominant one that has a similar timbre. This is probably caused by the instrument spectra having significant correlation with each other. Looking at the performance-vs.-polyphony results for each instrument in Figure 5, we see that instruments like violin, which have a unique timbre, have better average accuracy.

Possible solutions include training the instrument spectrum dictionaries not in isolation but in combination with other instrument spectra, which may assist the basis vectors behaving in a more discriminant way. Methods for discriminant non-negative tensor factorizations are explored in [19]. This issue can also be addressed in the testing part. Pitch probabilities in EM iterations can be estimated to be as maximally different as possible, while still explaining the overall mixture by appropriate use of priors. Another method that might improve this issue would be to use a factorial HMM to jointly estimate the pitch tracks.

On average, the proposed method scored 0.53 for Accuracy on the MIREX dataset, which is an improvement over the only past result (0.21) for the multiple-F0 tracking task evaluated at MIREX [10]. However, we note that the MIREX multiple-F0 task did not offer the opportunity to utilize instrument names for the mixture input test files.

One reason the proposed method works comparatively well for the trio and the Bach chorales case (see Table 1 and Figure 1) is that most instruments were active most of the time. The encouraging results from this case lead us to believe that the pitch tracking problems can be improved effectively by addressing the issues discussed above.

Finally, we note that the restriction that sounds must consist soley of harmonic partials can be relaxed. E.g., pitch estimates for instruments like xylophone, which do not have strict harmonic structures but do have predictable inharmonic structures, can be determined.

5. CONCLUSION

A new method for pitch and instrument tracking of individual instruments in polyphonic music has been designed and evaluated on an established dataset that has previously been used for multiple-F0 estimation under MIREX. Current results are encouraging, but several problems need to be resolved (as described in the Discussion section) for the method to be an effective tool.

As mentioned in the Introduction, knowing the F0 tracks can be very beneficial for a variety of MIR tasks. In the future, we would like to explore the use of voicing detection to determine which instruments are active. We would also like to perform instrument identification in the front-end so the method does not require prior knowledge of the instrumentation. We also plan to experiment with different discriminant learning methods and to re-infer the dictionaries based on the input mixture. Finally, we propose to explore using an automatic key detection algorithm and a more musicologically informed pitch transition matrix for the hidden Markov model.

6. REFERENCES

[1] M. Bay and J. Beauchamp. Harmonic source separation using prestored spectra. *Proc. Independent Component Analysis and Blind Signal Separation*, pages 561–568, 2006.

[2] M. Bay, A.F. Ehmann, and J.S. Downie. Evaluation of multiple-F0 estimation and tracking systems. In *Proc. 10th Int. Conf. for Music Information Retrieval (ISMIR 2009)*, pages 315–320, 2009.

[3] W.C. Chang, A.W.Y. Su, C. Yeh, A. Roebel, and X. Rodet. Multiple-f0 tracking based on a high-order hmm model. In *Proc. 11th Int. Conf. Digital Audio Effects (DAFx-08)*, pages 379–386, 2008.

[4] M. Desainte-Catherine and S. Marchand. High-precision Fourier analysis of sounds using signal derivatives. *J. Audio Eng. Soc*, 48(7/8):654–667, 2000.

[5] Z. Duan, J. Han, and B. Pardo. Harmonically informed multi-pitch tracking. In *Proc. 10th Int. Conf. on Music Information Retrieval (ISMIR 2009)*, pages 333–338, 2009.

[6] Z. Duan, J. Han, and B. Pardo. Song-level multi-pitch tracking by heavily constrained clustering. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2010)*, pages 57–60, 2010.

[7] M.R. Every and J.E. Szymanski. Separation of synchronous pitched notes by spectral filtering of harmonics. *IEEE Trans. Audio, Speech and Language Processing*, 14(5):1845–1856, 2006.

[8] G.D. Forney Jr. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[9] M. Goto. Development of the RWC music database. In *Proc. 18th Int. Congress on Acoustics (ICA 2004)*, volume 1, pages 553–556, 2004.

[10] IMIRSEL. Multiple fundamental frequency estimation and tracking results wiki. `http://www.music-ir.org/mirex/wiki/2010:Multiple_Fundamental_Frequency_Estimation_%26_Tracking_Results`, 2010.

[11] A. Klapuri. Multipitch analysis of polyphonic music and speech signals using an auditory model. *IEEE Trans. on Audio, Speech, and Language Processing*, 16(2):255–266, 2008.

[12] R. Marxer, J. Janer, and J. Bonada. Low-latency instrument separation in polyphonic audio using timbre models. *Proc. 10th Int. Conf. on Latent Variable Analysis and Signal Separation*, pages 314–321, 2012.

[13] G. Mysore, P. Smaragdis, and B. Raj. Non-negative hidden markov modeling of audio with application to source separation. *Proc. 8th Int. Conf. on Latent Variable Analysis and Signal Separation*, pages 140–148, 2010.

[14] A. Pertusa and J.M. Inesta. Multiple fundamental frequency estimation using gaussian smoothness. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing,(ICASSP 2008)*, pages 105–108, 2008.

[15] G.E. Poliner, D.P.W. Ellis, A.F. Ehmann, E. Gómez, S. Streich, and B. Ong. Melody transcription from music audio: Approaches and evaluation. *IEEE Trans. on Audio, Speech, and Language Processing*, 15(4):1247–1256, 2007.

[16] B. Raj and P. Smaragdis. Latent variable decomposition of spectrograms for single channel speaker separation. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, (WASPAA 2005)*, pages 17–20, 2005.

[17] P. Smaragdis, B. Raj, and M. Shashanka. A probabilistic latent variable model for acoustic modeling. *Proc. 20th Conf. on Neural Information Processing Systems (NIPS 2006)*, 148, 2006.

[18] C. Yeh, A. Roebel, and X. Rodet. Multiple fundamental frequency estimation and polyphony inference of polyphonic music signals. *IEEE Trans. on Audio, Speech, and Language Processing*, 18(6):1116–1126, 2010.

[19] S. Zafeiriou. Discriminant nonnegative tensor factorization algorithms. *IEEE Trans. on Neural Networks*, 20(2):217–235, 2009.