NOTE-LEVEL MUSIC TRANSCRIPTION BY MAXIMUM LIKELIHOOD SAMPLING

Zhiyao Duan
University of Rochester
Dept. Electrical and Computer Engineering
zhiyao.duan@rochester.edu

David Temperley
University of Rochester
Eastman School of Music
dtemperley@esm.rochester.edu

ABSTRACT

Note-level music transcription, which aims to transcribe note events (often represented by pitch, onset and offset times) from music audio, is an important intermediate step towards complete music transcription. In this paper, we present a note-level music transcription system, which is built on a state-of-the-art frame-level multi-pitch estimation (MPE) system. Preliminary note-level transcription achieved by connecting pitch estimates into notes often lead to many spurious notes due to MPE errors. In this paper, we propose to address this problem by randomly sampling notes in the preliminary note-level transcription. Each sample is a subset of all notes and is viewed as a note-level transcription candidate. We evaluate the likelihood of each candidate using the MPE model, and select the one with the highest likelihood as the final transcription. The likelihood treats notes in a transcription as a whole and favors transcriptions with less spurious notes. Experiments conducted on 110 pieces of J.S. Bach chorales with polyphony from 2 to 4 show that the proposed sampling scheme significantly improves the transcription performance from the preliminary approach. The proposed system also significantly outperforms two other state-of-the-art systems in both frame-level and note-level transcriptions.

1. INTRODUCTION

Automatic Music Transcription (AMT) is one of the fundamental problems in music information retrieval. Generally speaking, AMT is the task of converting a piece of music audio into a musical score. A complete AMT system needs to transcribe both the pitch and rhythmic content [5]. On transcribing the pitch content, AMT can be performed at three levels from low to high: frame-level, note-level, and stream-level [7]. Frame-level transcription (also called *multi-pitch estimation*) aims to estimate concurrent pitches and instantaneous polyphony in each time frame. Note-level transcription (also called *note tracking*) transcribes notes, which are characterized not only by pitch, but also by onset and offset. Stream-level transcription (also called *multi-pitch streaming*) organizes pitches (or notes) into streams according to their instruments. From the frame-level to the stream-level, more parameters and structures need to be estimated, and the system is closer to a complete transcription system.

While there are many systems dealing with frame-level music transcription, only a few transcribe music at the note level [5]. Among these systems, most are built based on frame-level pitch estimates. The simplest way to convert frame-level pitch estimates to notes is to connect consecutive pitches into notes [4, 9, 15]. During this process, non-significant errors in frame-level pitch estimation can cause significant note tracking errors. False alarms in pitch estimates will cause many notes that are too short, while misses can break a long note into multiple short ones. To alleviate these errors, researchers often fill the small gaps to merge two consecutive notes with the same pitch [2, 7], and apply minimum length pruning to remove too-short notes [4, 6, 7]. This idea has also been implemented with more advanced techniques such as hidden Markov models [12]. Besides the abovementioned methods that are entirely based on frame-level pitch estimates, some methods utilize other information in note tracking, such as onset information [10, 14] and musicological information [13, 14].

In this paper, we propose a new note-level music transcription system. It is built based on an existing multi-pitch estimation method [8]. In [8], a multi-pitch likelihood function was defined and concurrent pitches were estimated in a maximum likelihood fashion. This likelihood function tells how well the set of pitches as a whole fit to the audio frame. In this paper, we modify [8] to also define a single-pitch likelihood function. It tells the likelihood (salience) that a pitch is present in the audio frame. Then preliminary note tracking is performed by connecting consecutive pitches into notes and removing too-short notes. The likelihood of each note is calculated as the product of the likelihood of all its pitches. The next step is the key step in the proposed system. We randomly sample subsets of notes according to their likelihood and lengths. Each subset is treated as a possible note-level transcription. The likelihood of such a transcription is then defined as the product of its multi-pitch likelihood in each frame. Finally, the transcription with the maximum likelihood is returned as the output of the system. We carried out experiments on the Bach10 dataset [8] containing Bach chorales

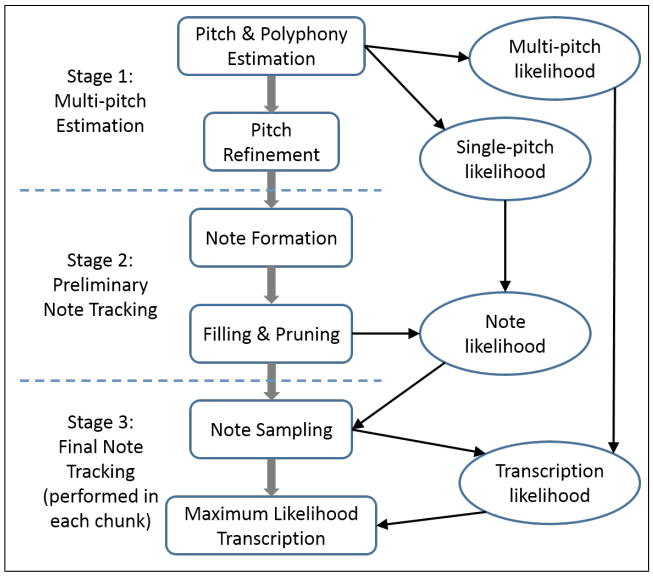

Figure 1. System overview of the proposed note-level transcription system.

with different polyphony. Experiments show that the proposed system significantly improves the transcription performance from the preliminary transcription, and significantly outperforms two state-of-the-art systems at both the note level and frame level on the dataset.

2. PROPOSED SYSTEM

Figure 1 illustrates the overview of the system. It consists of three main stages: multi-pitch estimation, preliminary note tracking, and final note tracking. The first stage is based on [8] with some modifications. The second stage adopts the common filling/prunning strategies used in the literature to convert pitches into notes. The third stage is the main contribution of the paper. Figure 2 shows transcription results obtained at different stages of the system on a piece of J.S. Bach 4-part chorale.

2.1 Multi-pitch Estimation

In [8], Duan et al. proposed a maximum likelihood method to estimate pitches from the power spectrum of each time frame. In the maximum likelihood formulation, pitches (and the polyphony) are the parameters to be estimated while the power spectrum is the observation. The likelihood function $L_{mp}(\{p_1, \cdots, p_N\})$ describes how well a set of N pitches $\{p_1, \cdots, p_N\}$ as a whole fit with the observed spectrum, and hence is called a *multi-pitch likelihood* function. The power spectrum is represented as peaks and the non-peak region, and the likelihood function is defined for both parts. The peak likelihood favors pitch sets whose harmonics can explain peaks, while the non-peak region likelihood penalizes pitch sets whose harmonic positions are in the non-peak region. Parameters of the likelihood function were trained from thousands of musical chords mixed with note samples whose ground-truth pitches were pre-calculated. The maximum likelihood estimation process uses an iterative greedy search strategy. It starts from an empty pitch set, and in each iteration the pitch candidate that results in the highest multi-pitch likelihood increase is selected. The process is terminated by thresholding on the likelihood increase, which also serves for polyphony estimation. After estimating pitches in each frame, a pitch refinement step that utilizes contextual information is performed to remove inconsistent errors.

In this paper, we use the same method to perform MPE in each frame. Differently, we change the instantaneous polyphony estimation parameter settings to achieve a high recall rate of the pitch estimates. This is because the note sampling module in Stage 3 will only remove false alarm notes but cannot add back missing notes (detailed explanation in Section 2.3). In addition, we also calculate a *single-pitch likelihood* $L_{sp}(p)$ for each estimated pitch p. We define it as the multi-pitch likelihood plugged in with the single pitch, i.e., $L_{sp}(p) = L_{mp}(\{p\})$. This likelihood describes how well the single pitch can explain the mixture spectrum, which apparently will not be very good. But from another perspective, this likelihood can be viewed as a salience of the pitch. One important property of multi-pitch likelihood is that it is not additive, i.e., the multi-pitch likelihood of a set of pitches is usually much smaller than the sum of their single-pitch likelihoods:

$$L_{mp}(\{p_1, \cdots, p_N\}) < \sum_{i=1}^{N} L_{mp}(\{p_i\}) = \sum_{i=1}^{N} L_{sp}(p_i) \quad (1)$$

The reason is that the multi-pitch likelihood definition in [8] considers the interaction between pitches. For example, in the peak likelihood definition, a peak will be explained by only one pitch in the pitch set, the one whose corresponding harmonic gives the best fit to the frequency and amplitude of the peak, even if the peak could be explained by multiple pitches. In other words, the single-pitch likelihood considers each pitch independently while the multi-pitch likelihood considers the set as a whole.

The reason of calculating the single-pitch likelihood is because we need to calculate a likelihood (salience) for each note in the second stage, which is further because we need to sample notes using their likelihood in the third stage. Since pitches in the same frame belong to different notes, we need to figure out the likelihood (salience) of each pitch instead of the likelihood of the whole pitch set.

Figure 2(a) shows the MPE result on the example piece. Compared to the ground-truth in (d), it is quite noisy and contains many false alarm pitches, although the main notes can be inferred visually.

2.2 Preliminary Note Tracking

In this stage, we implement a preliminary method to connect pitches into notes, with the ideas of filling and pruning that were commonly used in the literature [2, 4, 6, 7]. We first connect pitches whose frequency difference is less than 0.3 semitones and time difference is less than 100 ms. Each connected component is then viewed as a note. Then notes shorter than 100 ms are removed. The 0.3 semitones threshold corresponds to the range within which the pitch

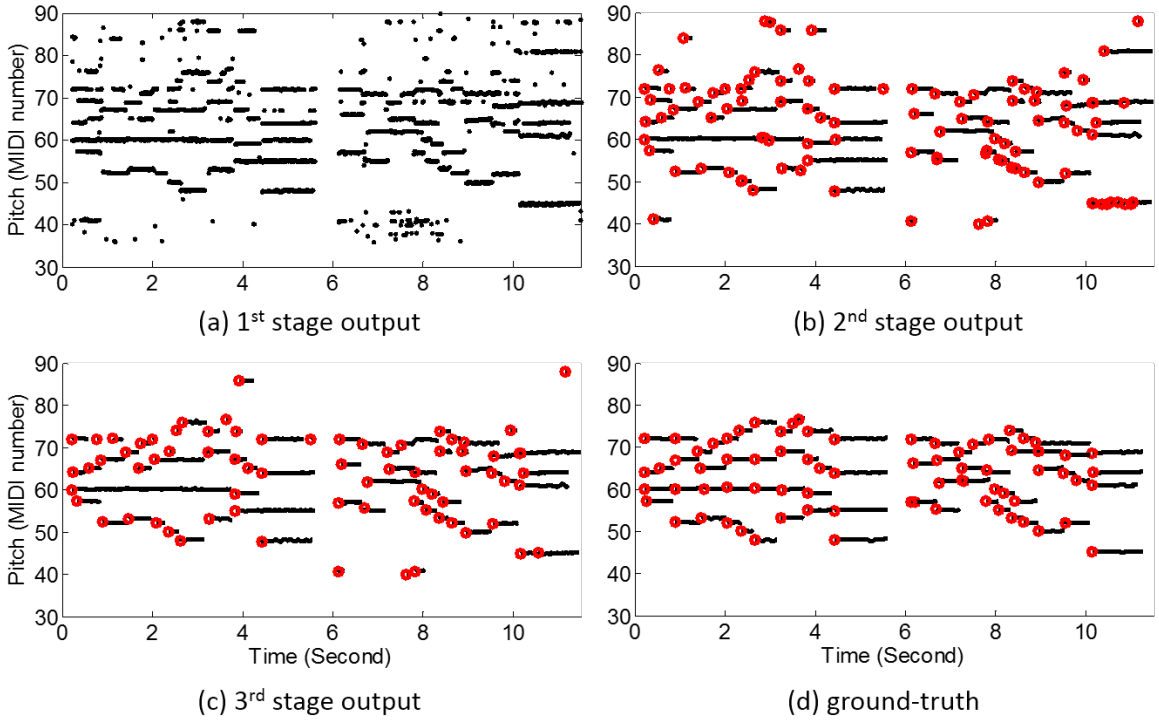

Figure 2. Transcription results on the first 11 seconds of *Ach Lieben Christen*, a piece of 4-part chorales by J.S. Bach. In (a), each pitch is plotted as a point. In (b)-(d), each note is plotted as a line whose onset is marked by a red circle.

often fluctuates within a note, while the 100 ms threshold is a reasonable length of a fast note, as it is the length of a 32nd note in music with a tempo of 75 beats per minute.

Each note is characterized by its pitch, onset, offset, and *note likelihood*. The onset and offset times are the time of the first and last pitch in the note, respectively. The pitch and likelihood are calculated by averaging the pitches and single-pitch likelihood values of all the pitches within the note. Again, this likelihood describes the salience of the note in the audio.

Figure 2(b) shows the preliminary note tracking result. Compared to (a), many noisy isolated pitches have been removed. However, compared to (d), there are still a number of spurious notes, caused by consistent MPE errors (e.g., the long spurious note starting at 10 seconds around MIDI number 80, and a shorter note starting at 4.3 seconds around MIDI number 60). A closer look tells us that both notes and many other spurious notes are higher octave errors of some already estimated notes. This makes sense as octave errors take about half of all errors in MPE [8].

Due to the spurious notes, the instantaneous polyphony constraint is often violated. The example piece has four monophonic parts and at any time there should be no more than four pitches. However, it is often to see more than four notes going simultaneous in Figure 2(b) (e.g., 0-1 seconds, 4-6 seconds, and 10-11 seconds). On the other hand, these spurious notes are hard to remove if we consider them independently: They are long enough from being pruned by the minimum length; They also have high enough likelihood, as the note likelihood is the average likelihood of its pitches. Therefore, we need to consider the interaction between different notes to remove these spurious notes. This leads to the next stage of the system.

2.3 Final Note Tracking

The idea of this stage is quite simple. Thanks to the MPE algorithm in Stage 1, the transcription obtained in Stage 2 inherits the high recall and low precision property. Therefore, a subset of the notes that do not contain many spurious notes but contain almost all correct notes must be a better transcription. The only question now is how can we know which subset is a good transcription. This question can be addressed by an exploration-evaluation strategy: we first explore a number of subsets, and then we evaluate these subsets according to some criterion. But there are two problems of this strategy: 1) how can we efficiently explore the subsets? The number of all subsets is two to the power of the number of notes, hence it is inefficient to enumerate all the subsets. 2) What criterion should we use to evaluate the subsets? If our criterion considers notes independently, then it would not work well, as the spurious notes are hard to distinguish from correct notes in terms of individual note properties such as length and likelihood.

2.3.1 Note Sampling

Our idea to address the exploration problem is to perform note sampling. We randomly sample notes without replacement according to their weights. The weight equals to the product of the note length and the inverse of the negative logarithmic note likelihood. Essentially, longer notes with higher likelihood are more likely to be sampled into

the subset. In this way, we can explore different note subsets, and can guarantee that notes contained in each subset are mostly correct. During the sampling, we also consider the instantaneous polyphony constraint. A note will not be sampled if adding it to the subset would violate the instantaneous polyphony constraint. The sampling process stops if there is no valid note to sample any more.

We perform the sampling process M times to generate M subsets of the notes output in Stage 2. Each subset is viewed as a transcription candidate. We then evaluate the *transcription likelihood* for each candidate and select the one with the highest likelihood. The transcription likelihood is defined as the product of the multi-pitch likelihood of all time frames in the transcription. Since multi-pitch likelihood considers interactions between simultaneous pitches, the transcription likelihood also considers interactions between simultaneous notes. This can help remove spurious notes which are higher octave errors of some correctly transcribed notes. This is because all the peaks that a higher octave error pitch can explain can also be explained by the correct pitch, hence having the octave error pitch in addition to the correct pitch would not increase the multi-pitch likelihood much.

2.3.2 Chunking

The number of subsets (i.e., the sampling space) increases with the number of notes exponentially. If we perform sampling on a entire music piece that contains hundreds of notes, it is likely to require many times of sampling to reach a good subset (i.e., transcription candidate). In order to reduce the sampling space, we segment the preliminary note tracking transcription into one-second long non-overlapping chunks and perform sampling and evaluation in each chunk. Finally, selected transcriptions of different chunks are merged together to get the final transcription of the entire piece. Notes that span across multiple chunks can be sampled in all the chunks, and they will appear in the final transcription if they appear in the selected transcription of some chunk. Depending on the tempo and polyphony of the piece, the number of notes within a chunk can be different. For the 4-part Bach chorales tested in this paper, there are about 12 notes per chunk, and we found sampling 100 subsets gives good accuracy and efficiency.

Figure 2(c) shows the final transcription of the system. We can see that many spurious notes are removed from (b) while most correct notes remain, resulting a much better transcription. The final transcription is very close to the ground-truth transcription.

3. EXPERIMENTS

3.1 Data Set

We use the Bach10 dataset [8] to evaluate the proposed system. This dataset consists of real musical instrumental performances of ten pieces of J.S. Bach four-part chorales. Each piece is about thirty seconds long and was performed by a quartet of instruments: violin, clarinet, tenor saxophone and bassoon. Both the frame-level and note-level ground-truth transcriptions are provided with the dataset. In order to evaluate the system on music pieces with different polyphony, we use the dataset-provided matlab script to create music pieces with different polyphony, which are different combinations of the four parts of each piece. Six duets, four trios and one quartet for each piece was created, totaling 110 pieces of music with polyphony from 2 to 4.

3.2 Evaluation Measure

We evaluate the proposed transcription system with commonly used note-level transcription measures [1]. A note is said to be correctly transcribed, if it satisfies both the pitch condition and the onset condition: its pitch is within a quarter tone from the pitch of the ground-truth note, and its onset is within 50 ms from the ground-truth onset. Offset is not considered in determining correct notes. Then precision, recall, and F-measure are defined as

$$P = \frac{TP}{TP+FP}, R = \frac{TP}{TP+FN}, F = \frac{2PR}{(P+R)}, \quad (2)$$

where TP (true positives) is the number of correctly transcribed notes, FP (false positives) is the number of reported notes not in the ground-truth, and FN (false negatives) is the number of ground-truth notes not reported .

Although note offest is not used in determining correct notes, we do measure the Average Overlap Ratio (AOR) between correctly transcribed notes and their corresponding ground-truth notes. It is defined as

$$AOR = \frac{\min(offsets) - \max(onsets)}{\max(offsets) - \min(onsets)} \quad (3)$$

AOR ranges between 0 and 1, where 1 means that the transcribed note overlaps exactly with the ground-truth note.

To see the improvement of different stages of the proposed system, we also evaluate the system using frame-level measures. Again, we use precision, recall, and F-measures defined in Eq. (2), but here the counts are on the pitches instead of notes. A pitch is considered correctly transcribed if its frequency is within a quarter tone from a ground-truth pitch in the same frame.

3.3 Comparison Methods

3.3.1 Benetos et al.'s System

We compare our system with a state-of-the-art note-level transcription system proposed by Benetos et al. [3]. This system first uses shift-invariant Probabilistic Latent Component Analysis (PLCA) to decompose the magnitude spectrogram of the music audio with a pre-learned dictionary containing spectral templates of all semitone notes of 13 kinds of instruments (including the four kinds used in the Bach10 dataset). The activation weights of the dictionary elements provide the soft version of the frame-level transcription. It is then binarized to obtain the hard version of the frame-level transcription. Note-level transcription is obtained by connecting consecutive pitches, filling short gaps between pitches, and pruning short notes.

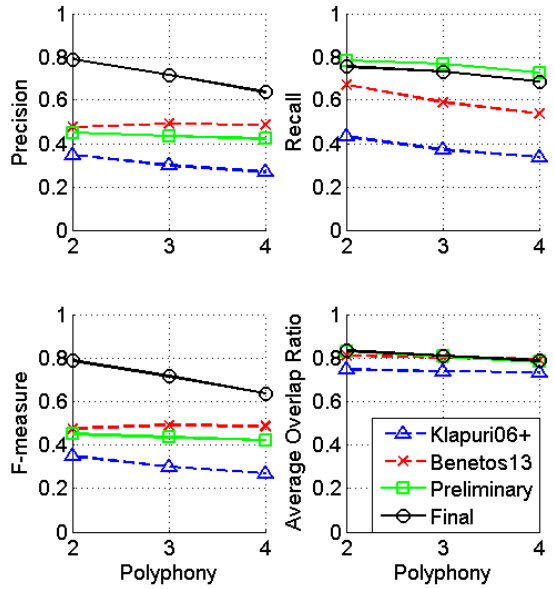

Figure 3. Note-level transcription performances.

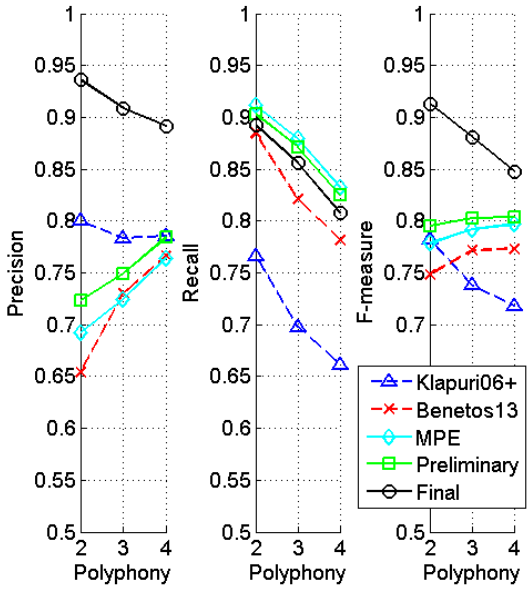

Figure 4. Frame-level transcription performances.

The author's own implementation is available online to generate the soft version frame-level transcription. We then implemented the postprocessing steps according to [3]. Since the binarization threshold is very important in obtaining good transcriptions, we performed a grid search between 1 and 20 with a step size of 1 on the trio pieces. We found 12 gave the best note-level F-measure and used it in all experiments. The time threshold for filling and pruning were set to 100 ms, same as the other comparison methods. We denote this comparison system by "Benetos13".

3.3.2 Klapuri's System

Klapuri's system [11] is a state-of-the-art general-purposed frame-level transcription system. It employs an iterative spectral subtraction approach. At each iteration, a pitch is estimated according to a salience function and its harmonics are subtracted from the mixture spectrum. We use Klapuri's original implementation and suggested parameters. Since Klapuri's system does not output note-level transcriptions, we employ the preliminary note tracking stage in our system to convert Klapuri's frame-level transcriptions into note-level transcriptions. We denote this comparison system by "Klapuri06+".

3.4 Results

Figure 3 compares the note-level transcription performance of the preliminary and final results of the proposed system with Benetos13 and Klapuri06+. It can be seen that the precision of the final transcription of the proposed system is improved significantly from the preliminary transcription for all polyphony. This is accredited to the note sampling stage of the proposed system. As shown in Figure 2, note sampling removes many spurious notes and leads to higher precision. On the other hand, the recall of the final transcription is just slightly decreased (about 3%), which means most correct notes survive during the sampling. Therefore, the F-measure of the final transcription is significantly improved from the preliminary transcription for all polyphony, leading to a very promising performance on this dataset. The average F-measure on the 60 duets is about 79%, which is about 35% higher than the preliminary result in absolute value. The average F-measure on the 10 quartets is about 64%, which is also about 22% higher than the preliminary transcription.

Compared to the two state-of-the-art methods, the final transcription of the proposed system also achieves much higher F-measure. In fact, the preliminary transcription is a little inferior to Benetos13. However, the note sampling stage makes the final transcription surpass Benetos13.

In terms of average overlap ratio (AOR) of the correctly transcribed notes with the ground-truth notes, both preliminary and the final transcription of the proposed system and Benetos13 achieve a similar performance, which is about 80% for all polyphony. This is about 5% higher than Klapuri06+. It is noted that 80% AOR indicates a very good estimation of the note lengths/offsets.

Figure 4 presents the frame-level transcription performance. In this comparison, we also include the MPE result which is the output of Stage 1. There are several interesting observations. First of all, similar to the results in Figure 3, the final transcription of the proposed system improves from the preliminary transcription significantly in both precision and F-measure, and degrades slightly in recall. This is accredited to the note sampling stage. Second, preliminary transcription of the proposed system has actually improved from the MPE result in F-measure. This validates the filling and pruning operations in the second stage, although the increase is only about 3%. Third, the final transcription of the proposed system achieves significantly higher precision and F-measure than the two com-

parison methods, leading to about 91%, 88%, and 85% F-measure for polyphony 2, 3, and 4, respectively. This performance is very promising and may be accurate enough for many other applications.

4. CONCLUSIONS AND DISCUSSIONS

In this paper, we built a note-level music transcription system based on an existing frame-level transcription approach. The system first performs multi-pitch estimation in each time frame. It then employs a preliminary note tracking to connect pitch estimates into notes. The key step of the system is to perform note sampling to generate a number of subsets of the notes, where each subset is viewed as a transcription candidate. The sampling was based on the note length and note likelihood, which was calculated using the single-pitch likelihood of pitches in the note. Then the transcription candidates are evaluated using the multi-pitch likelihood of simultaneous pitches in all the frames. Finally the candidate with the highest likelihood is returned as the system output. The system is simple and effective. Transcription performance was significantly improved due to the note sampling and likelihood evaluation step. The system also significantly outperforms two other state-of-the-art systems on both note-level and frame-level measures on music pieces with polyphony from 2 to 4.

The technique proposed in this paper is very simple, but the performance improvement is unexpectedly significant. We think the main reason is twofold. First, the note sampling step lets us explore the transcription space, especially the good regions of the transcription space. The single-pitch likelihood of each estimated pitch plays an important role in sampling the notes. In fact, we think that probably any kind of single-pitch salience function that have been proposed in the literature can be used to perform note sampling. The second reason is that we use the multi-pitch likelihood, which considers interactions between simultaneous pitches, to evaluate these sampled transcriptions. This is important because notes contained in a sampled transcription must have high salience, however, when considered as a whole, they may not fit with the audio as well as another sampled transcription. One limitation of the proposed sampling technique is that it can only remove false alarm notes in the preliminary transcription but not adding back missing notes. Therefore, it is important to make the preliminary transcription have a high recall rate before sampling.

5. ACKNOWLEDGEMENT

We thank Emmanouil Benetos and Anssi Klapuri for providing the source code or executable program of their transcription systems for comparison.

6. REFERENCES

[1] M. Bay, A.F. Ehmann, and J.S. Downie, "Evaluation of multiple-F0 estimation and tracking systems," in *Proc. ISMIR*, 2009, pp. 315-320.

[2] J.P. Bello, L. Daudet, M.B., Sandler, "Automatic piano transcription using frequency and time-domain information," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 6, pp. 2242-2251, 2006.

[3] E. Benetos, S. Cherla, and T. Weyde, "An efficient shift-invariant model for polyphonic music transcription," in *Proc. 6th Int. Workshop on Machine Learning and Music*, 2013.

[4] E. Benetos and S. Dixon, "A shift-invariant latent variable model for automatic music transcription," *Computer Music J.*, vol. 36, no. 4, pp. 81-94, 2012.

[5] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: challenges and future directions," *J. Intelligent Information Systems*, vol. 41, no. 3, pp. 407-434, 2013.

[6] A. Dessein, A. Cont, G. Lemaitre, "Real-time polyphonic music transcription with nonnegative matrix factorization and beta-divergence," in *Proc. ISMIR*, 2010, pp. 489-494.

[7] Z. Duan, J. Han, and B. Pardo, "Multi-pitch streaming of harmonic sound mixtures," *IEEE Trans. Audio Speech Language Processing*, vol. 22, no. 1, pp. 1-13, 2014.

[8] Z. Duan, B. Pardo, and C. Zhang, "Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions," *IEEE Trans. Audio Speech Language Processing*, vol. 18, no. 8, pp. 2121-2133, 2010.

[9] G. Grindlay and D. Ellis, "Transcribing multi-instrument polyphonic music with hierarchical eigeninstruments," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1159-1169, 2011.

[10] P. Grosche, B. Schuller, M. Mller, and G. Rigoll, "Automatic transcription of recorded music," *Acta Acustica United with Acustica*, vol. 98, no. 2, pp. 199-215, 2012.

[11] A. Klapuri, "Multiple fundamental frequency estimation by summing harmonic amplitudes," in *Proc. ISMIR*, 2006, pp. 216-221.

[12] G. Poliner, and D. Ellis, "A discriminative model for polyphonic piano transcription," in *EURASIP J. Advances in Signal Processing*, vol. 8, pp. 154-162, 2007.

[13] S.A. Raczyǹski, N. Ono, and S. Sagayama. "Note detection with dynamic bayesian networks as a post-analysis step for NMF-based multiple pitch estimation techniques," in *Proc. WASPAA*, 2009, pp. 49-52.

[14] M. Ryynänen and A. Klapuri, "Polyphonic music transcription using note event modeling," in *Proc. WASPAA*, 2005, pp. 319-322.

[15] E. Vincent, N. Bertin, and R. Badeau, "Adaptive harmonic spectral decomposition for multiple pitch estimation," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 528-537, 2010.