MULTI-PITCH STREAMING OF INTERWOVEN STREAMS

Chih-Yi Kuan[1], *Li Su*[2], *Yu-Hao Chin*[1], *Jia-Ching Wang*[1]

[1]Department of Computer Science and Information Engineering, National Central University, Taiwan
[2]Institute of Information Science, Academia Sinica, Taiwan

ABSTRACT

In this paper, we discuss the multipitch streaming (MPS) problem for a multi-source audio signal having interweaving pitch contours. We propose two approaches to tackle this challenge, one relates to a feature extracted from the energy levels distributed in multi-channel recordings for better characterization of the source, and the other uses particle swarm optimization (PSO) to enlarge the search space and alleviate the initialization problem in constrained clustering of the features representing different sources. Experiments on music and speech samples having highly interweaving pitch contours are presented to assess its effectiveness.

Index Terms— Multipitch streaming, automatic music transcription, particle swarm optimization, multi-channel

1. INTRODUCTION

In *multi-source* audio signal processing, recognizing the time-varying behavior of the fundamental frequencies (F0s) of every source is a challenging task.[1] In the literature of automatic music transcription (AMT) [1], this problem was solved at different levels, such as multipitch estimation (MPE) in frame-level, note tracking (NT) in note-level, and multipitch streaming (MPS) in stream-level [2]. MPS is particularly related to a number of music and speech processing problems, including melody tracking, instrument identification [3], source separation [4], speech recognition [5, 6, 7], and prosody analysis [8], to name but a few.

In this paper, we study the problem of MPS, which aims to determine the pitch contour of every source based on a known MPE result, i.e., frame-level pitch estimations. In other words, the main difference between MPE and MPS is that MPE returns activated pitches at every time instance, while MPS returns the pitches and their corresponding source labels. In comparison to MPE, MPS is complicated by the characteristics of sources, such as the timbre of different instruments. Therefore, challenges of MPS include silence, non-pitched sounds, abrupt frequency changes in a stream [2], error propagation from the MPE result, etc. However, extra complexity of this problem is introduced by an essential but under discussed issue, the interwoven streams.

Fig. 1 illustrates what interwoven streams are. The first example shown in the left of Fig. 1 is one of Bach's chorales, which has four streams, and each of them does not cross one another. This is different from the second example, a three-part country music shown in the right of Fig. 1, where the pitch contours of vocal and guitar

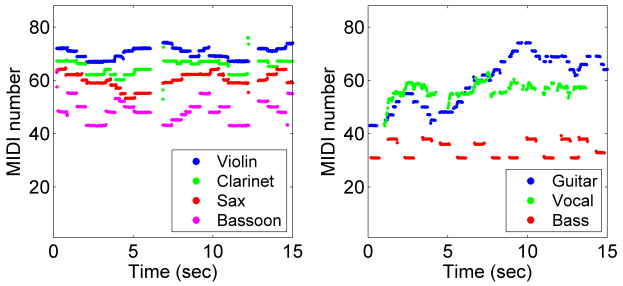

Fig. 1. Examples of voice crossing in polyphonic music. Left: Bach's *Ach Gott und Herr, wie groß und schwer* from the Bach10 dataset [11], without voice crossing. Right: *MusicDelta_Country1* from the MedleyDB dataset [12], with voice crossing.

cross each other three times in a 15-second recording. This so-called *voice crossing* phenomenon is avoided in Bach's composition [9, 10] and often prohibited in pedagogical composition in music theory, but is frequently seen in pop or folk music.[2] However, in previous studies, discussion on this issue is quite limited.

The lack of discussion motivates us to investigate how interwoven streams affect the performance of an MPS algorithm, and how to improve the algorithm in recognizing interwoven streams. On the basis of the *constrained clustering* approach [2], we propose two enhanced schemes: first, since most of the recordings are in a dual-channel format, we propose a dual-channel feature to better discriminate different sources by using directional information; second, we introduce particle swarm optimization (PSO) [13, 14] into the constrained clustering method, in order to address the issue of high sensitivity to initialization of cluster centers. Evaluation is then performed on a dataset having highly interwoven streams, and the Bach10 dataset having few interwoven streams, and the results are compared. The proposed method is shown to be useful for interwoven streams, and particularly useful in dual-channel recordings in a real-world environment, where the location of a source affects an energy ratio in both channels.

2. RELATED WORK

An MPS system under our discussion has three inputs: an audio signal, pitch labels (i.e., the MPE result), and the number of sources. The system contains mainly two parts: 1) feature extraction, and 2) streaming. For the feature extraction, most of the previous works considered pitch-informed, spectral-based features, such as the activation matrix from the probabilistic latent component analysis

[1]In this paper, the term "multi-source signal" refers to a signal generated by multiple sources including instruments or speakers, with each source assumed to be *mono-phonic*; i.e., a source only generates one pitch at a time. With this assumption, a multi-source signal is *polyphonic*, while a polyphonic signal is not necessarily multi-source, such as piano solo music, which is beyond the scope of our discussion.

[2]In this paper, the terms "voice," "stream," "source", and "pitch contour" are used interchangeably.

(PLCA) [15, 16, 17], correlogram [5], cochleagram [6], and the uniform discrete cepstrum (UDC) [2]. A main challenge in feature extraction is that the harmonic peaks of the spectra of different sources tend to overlap with each other, making it difficult to extract a feature representing a clean source from a multi-source signal.

For the streaming algorithm, previous studies have adopted a number of supervised or semi-supervised algorithms, such as the hidden Markov model (HMM) [15, 18], deep neural networks [6], discriminate PLCA [19], and hidden-Markov random fields (HMRF) [17]. Besides, there are also unsupervised streaming algorithms being proposed, such as spectral clustering [17] and constrained clustering [2]. In comparison to a supervised algorithm, an unsupervised algorithm does not require a multi-track dataset for training, and becomes more favorable for the MPS of general types of music. An unsupervised algorithm is based on an assumption in which similar features belong to the same stream (i.e., cluster), and resulting streams are musically meaningful (i.e., constraints of mono-phonic activation of a source, continuity of a pitch contour, etc.).

We base our work on the system proposed in [2], which uses the UDC as feature representation and a constrained clustering scheme in determining the streams. Detailed description of our enhanced schemes is given in the next two sections.

3. FEATURE EXTRACTION

3.1. The UDC feature

Cepstral features have been widely used in timbre classification [20, 21]. A cepstrum is defined as the inverse Fourier transform (IFT) of a log-scale spectrum. To extract the feature of one source in a multi-source signal, the IFT is performed only in the region where energy of a source is distributed in the spectrum of the multi-source signal; this is the basic idea of the UDC. More specifically, a log-amplitude spectrum of the mixture signal $\mathbf{a} = [a(i)]_{i=1}^{N}$ is located in the frequency bins $\mathbf{f} = [f(i)]_{i=1}^{N}$, $N \in \mathbb{N}$, and $\hat{\mathbf{a}} = [a(i)]_{i=1}^{L}$ and $\hat{\mathbf{f}} = \left[\hat{f}(i)\right]_{i=1}^{L}$ represent a potential subset of the spectrum and the frequency bins that solely belong to the source of interest. This subset is characterized by the first 50 harmonics of the source from its given pitch obtained from an MPE algorithm. The UDC is obtained by computing the cepstrum of this subset:

$$\mathbf{c} = \begin{bmatrix} 1 & \sqrt{2}\cos(2\pi\hat{f}_1) & \dots & \sqrt{2}\cos(2\pi(p-1)\hat{f}_1) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \sqrt{2}\cos(2\pi\hat{f}_L) & \dots & \sqrt{2}\cos(2\pi(p-1)\hat{f}_L) \end{bmatrix}^T \begin{bmatrix} \hat{a}_i \\ \vdots \\ \hat{a}_L \end{bmatrix}, \tag{1}$$

where p is the order of the cepstral representation. In this paper, we set $L = 22$. More implementation details of the UDC can be found in [2] and its source codes provided online (`http://www.ece.rochester.edu/~zduan/resource/Resources.html`).

3.2. The level-ratio feature

Most of the music recordings are in the multi-channel format. For example, in the scenario of a live concert, sound sources are typically located at different places on the stage, and are commonly recorded by two (or more) microphones. In this case, the energy of each source distributes differently in the two (or more) recording channels, which provides extra information for identifying the source. Therefore, we design features for the MPS problem by using channel information. For a C-channel recording, $\mathbf{b}^{(c)} = [b^{(c)}(i)]_{i=1}^{N}$ is a magnitude spectrum, where $1 \le c \le C$. A *normalized amplitude distribution* for a pitch detection g as $\mathbf{l}^{(c)} = [b^{(c)}(i)/\sum_{c=1}^{C} b^{(c)}(i)]_{i=g-m}^{g+m}$, where m is a constant, and g is the frequency bin corresponding to the pitch label. In this work, we set $m = 4$. A *level-ratio* feature derived from the above is represented as

$$\mathbf{d} = \left[\mathbf{l}^{(1)}, \mathbf{l}^{(2)}, \cdots, \mathbf{l}^{(C)}\right], \tag{2}$$

where the dimension is $C(2m+1)$. Next, a fusion feature concatenating a scaled timbre feature $\bar{\mathbf{c}} = \mathbf{c}/\sigma_c$ and a scaled level-ratio feature $\bar{\mathbf{d}} = \mathbf{d}/\sigma_d$ is represented as:

$$\mathbf{u} = \left[\bar{\mathbf{c}}, \bar{\mathbf{d}}\right], \tag{3}$$

where σ_c and σ_d are the standard deviation for feature dimensions of $\mathbf{c}$ and $\mathbf{d}$, respectively.

4. CLUSTERING

4.1. Constrained clustering

For an input signal having K sources, a tuple (t, f, k) denotes a detected pitch activation of a k-th source at a time t and a pitch f, where t and f are both positive real numbers, and $k = 1, 2, \dots, K$. D represents the number of all pitch activations in one music piece. To define an one-to-one mapping $\mathscr{D} : (t, f)_{\text{activated}} \to \{1, 2, \cdots, D\}$, the set of all pitch activations are indexed as $\mathbf{p} = \{p(d)\}_{d=1}^{D}$, where $p(d)$ (or $p(t, f)$) represents the stream label k of the d-th pitch activation at time t and pitch f (i.e., $p(d) = k$). In this paper, $\mathbf{p}$ is referred to as a *stream partition*, $S_k = \{i | p(i) = k\}$ is defined as the set of all indices of the k-th stream, $\mathbf{u}_d$ represents the fusion feature (i.e., Equation (3)) of the d-th pitch activation, and $\mathbf{c}_k$ represents the center of the k-th source. Then, the multipitch streaming problem is solved by minimizing the following objective function:

$$f(\mathbf{p}) = \sum_{k=1}^{K} \sum_{d \in S_k} \|\mathbf{u}_d - \mathbf{c}_k\|^2. \tag{4}$$

The minimization can be solved by using the well-known K-means algorithm. Moreover, every (t, f) in S_k should satisfy two constraints derived from the domain knowledge of music [2]:

1. *Must-link*: different sources rarely have the same pitch at the same time; therefore, a pitch activation different from another pitch activation in a neighboring frame by less than one semitone should be assigned to the same source as the other pitch activation.
2. *Cannot-link*: one source does not perform two pitches at the same time. Namely, $p(t_1, f_1)$ and $p(t_2, f_2)$ should belong to different clusters if t_1 is close to t_2, while p_1 and p_2 are far apart. Two pitches occurring at the same time must satisfy this relation. The set of all partitions satisfying this cannot-link condition is defined as Γ.

After checking the above two constraints against all pitches, a refined list of constrained pitch candidates is obtained.

4.2. PSO-based constrained clustering

The K-means algorithm suffers a possible convergence to local minima due to its is sensitivity to initialization of cluster centers. Previous studies suggested pitch order initialization [2], but this method is not applicable to the case of interwoven streams. Therefore, a

Algorithm 1 PSO-based constrained clustering.

[INPUT] Feature set of all pitch activations $(\mathbf{u}_i)_{i=1}^{D}$, number of particles γ, maximum number of iteration η
[OUTPUT] Optimal partition $\mathbf{p} \in \mathbb{R}^D$
for $j = 1, \ldots, \gamma$ **do**
  Initialize $\mathbf{q}^{(1,j)}(d), \mathbf{v}^{(1,j)}(d), \mathbf{p}_{\text{pb}}^{(j)}(d) \sim U(0.5, K+0.5)$, $d = 1, \ldots, D$
end for
Initialize $\mathbf{p}_{\text{gb}}(d) \sim U(0.5, K+0.5)$, $d = 1, \ldots, D$
while $i < \eta$ **do**
  for $j = 1, \ldots, \gamma$ **do**
    $\mathbf{p}^{(i,j)} \leftarrow \text{argmin}_{\mathbf{p} \subset \Gamma} \|\mathbf{q}^{(i,j)} - \mathbf{p}\|^2$
    $\mathbf{p}^{(i,j)} \leftarrow \text{FindNewPartition}(\mathbf{p}^{(i,j)})$ [2]
    if $g(\mathbf{p}^{(i,j)}) < g(\mathbf{p}_{\text{pb}}^{(j)})$ and $f(\mathbf{p}^{(i,j)}) < f(\mathbf{p}_{\text{pb}}^{(j)})$ **then**
      $\mathbf{p}_{\text{pb}}^{(j)} \leftarrow \mathbf{p}^{(i,j)}$
      if $g(\mathbf{p}^{(i,j)}) < g(\mathbf{p}_{\text{gb}})$ and $f(\mathbf{p}^{(i,j)}) < f(\mathbf{p}_{\text{gb}})$ **then**
        $\mathbf{p}_{\text{gb}} \leftarrow \mathbf{p}^{(i,j)}$
      end if
    end if
  end for
  $\mathbf{v}^{(i,j)} \leftarrow \mathbf{v}^{(i,j)} + b_1\phi_1(\mathbf{p}_{\text{pb}}^{(j)} - \mathbf{p}^{(i,j)}) + b_2\phi_2(\mathbf{p}_{\text{gb}} - \mathbf{p}^{(i,j)})$
  $\mathbf{q}^{(i+1,j)} \leftarrow \mathbf{p}^{(i,j)} + \mathbf{v}^{(i,j)}$
end while
$\mathbf{p} \leftarrow \mathbf{p}_{\text{gb}}$

different approach called particle swarm optimization (PSO) is employed in this work. PSO is inspired from behaviors in biological systems, such as a bird flock or a fish school [13, 14]. PSO solves a problem by having a population of candidate solutions (i.e., particles) and optimizing each candidate individually in terms of a *fitness function*. The fitness function can be an objective function to the problem, where the position and velocity of a particle are formulated. Movement of a particle is influenced by its local best known position, and also guided toward the best known position in the whole search space determined by other particles. Therefore, a global solution is obtained by moving the swarm toward the best position.

We use PSO in solving the constrained clustering problem. In PSO-based constrained clustering, assume that we have R different sets of candidate stream partitions $\mathbf{p}^{(1)}, \cdots, \mathbf{p}^{(R)}$, where $\mathbf{p}^{(r)} = \{p^r(d)\}_{d=1}^{D}$ and $S_k^{(r)} = \{i | \{p^r(i) = k\}$. Initially, all elements of each particle (i.e., each stream partition) $\mathbf{p}^{(r)}$ are randomly generated in the interval, $[0.5, K+0.5]$, such that every cluster has the same random search space at first. Note that in this stage, each particle is yet a meaningful stream partition since its values are not integers. If a particle element value is 3.2, its corresponding pitch can be assigned to the third cluster. To refine the initialization, the *cannot-link* constraint is first imposed on $\mathbf{p}^{(r)}$ by finding the minimum difference between the particle element values and all possible orderings of the cluster labels $\mathbf{q}^{(r)}$:

$$\hat{\mathbf{p}}^{(r)} = \underset{\mathbf{p}^{(r)} \subset \Gamma}{\text{argmin}} \|\mathbf{q}^{(r)} - \mathbf{p}^{(r)}\|^2 . \quad (5)$$

This procedure is done by an exhaustive search and generates an integer-valued $\mathbf{p}^{(r)}$. Via (5), the closest cluster labels satisfying the cannot-links constraints are found and assigned to the pitches at time t. Then, each candidate is evaluated using two fitness functions, $g(\mathbf{p})$ and $f(\mathbf{p})$:

1. $g(\mathbf{p})$ outputs a total number of must-link violations. Here, we apply an additional process used in [2], called swap operation, to trace all pitches and change their stream labels to the respective clusters, and to check if there is a stream label better satisfies the constraints and minimizes the total within-source feature distance at the same time. If the swap operation finds a better set of stream labels, we update this result to that particle. Its purpose is to find better must-links.
2. $f(\mathbf{p})$ outputs the sum of the intra-source feature distance represented as (4).

A locally optimal candidate, $\mathbf{p}_{\text{pb}}$, has the best fitness function value in all iteration steps of that candidate, and a globally optimal candidate, $\mathbf{p}_{\text{gb}}$, has the one with the best fitness function value among all candidates. The direction of optimization (i.e., the velocity of the swarm) of each particle is therefore defined as follows for the i-th iteration and the j-th particle:

$$\mathbf{v}^{(i,j)} \leftarrow \mathbf{v}^{(i,j)} + b_1\phi_1(\mathbf{p}_{\text{pb}}^{(j)} - \mathbf{p}^{(i,j)}) + b_2\phi_2(\mathbf{p}_{\text{gb}} - \mathbf{p}^{(i,j)}), \quad (6)$$

where b_1 and b_2 are two constants, and ϕ_1 and ϕ_2 are two Gaussian random variables with a zero mean and unit standard deviation. In this paper, $b_1 = b_2 = 0.2$, and we choose $\gamma = 7$ particles, five of which are initialized randomly, one by a simple K-means result, and one by pitch ordering. The maximal number of iteration η is set to 5. The proposed algorithm is detailed in Algorithm 1.

4.3. Post-processing

We apply two post-processing steps. First, we examine the pitches against the must-link constraints, and group every 10 frame-level pitches (i.e., every 0.3s) satisfying the must-link relation into a non-overlapping segment. We make a vote over the stream labels, and assign the 10 pitches to the voting result. This is a smoothing process which is usually required in refining the frame-level estimation [22]. Then, if there are still two pitches violating the cannot-link constraint, we take the pitch farther from the cluster center as a false positive and remove it.

5. EXPERIMENT AND RESULT

5.1. Data

We collect a dataset consisting of five clips by selecting 2-3 tracks having highly interwoven streams from several music and speech datasets (see Table 1). To quantify how much the streams are interwoven in a clip, we consider the number of frames whose stream labels in pitch order is not equal to the average pitch order of the full clip; a higher number indicates more interwoven streams. Then, we define a term named *interwoven rate* (IR), which is the ratio between this number and the total frame number in a clip. The IR of each clip in the "Interwoven" dataset is listed in Table 1.

We also consider the Bach10 dataset, which has ten of Bach's chorales played by four different instruments (violin, clarinet, saxophone, bassoon) [11, 2]. The four streams in the dataset interweave rarely. The average IR of the Bach10 dataset is only 4.761%.

All recordings are sampled at 44.1 kHz. We use a Hamming window of size 46.4 ms and a hop size of 30 ms for feature extraction. To simulate a room performance recorded by two microphones, all songs are remixed using the open-source room acoustics simulator Roomsim [23]. For PSO-based clustering, each setting is repeated for 3 rounds in order to get stable results. We evaluate the accuracy of MPE using the average frame-level F-score, which counts the average F-score over all clips and rounds.

Table 1. The dataset with interwoven streams. The first three clips are selected from the MedleyDB dataset [12], the fourth is from the MIREX dataset, and the final from the CHAIN-Corpus [24].

Filename	Parts (selected)	IR (%)
MusicDelta_ChineseJiangNan	erhu, guzheng, ruan	12.92
MusicDelta_ChineseHenan	erhu, guzheng, liuqin	19.18
MusicDelta_Country1	vocal, guitar, bass	56.22
MIREX Multi-F0 training data	clarinet, horn	30.96
CHAIN-Corpus	two male talkers	10.45

Table 2. Comparison of average F scores (in %) over all clips and rounds for various features and clustering algorithms between the Bach10 and the Interwoven datasets using ground-truth MPE result.

		K-means	CC	PSOCC
Bach10	UDC	70.63	94.24	94.81
	LR	71.82	85.96	86.40
	UDC+LR	75.21	92.27	92.64
Interwoven	UDC	59.32	67.63	67.99
	LR	72.77	73.55	73.08
	UDC+LR	67.06	71.48	71.78

5.2. Result

We perform two MPS experiments. The first one takes the ground truth pitch annotation as the input, i.e., the result in the MPE stage is assumed to be perfect. Results are shown in Table 2. We compare three different kinds of features, including the UDC, the proposed level-ratio (LR) and the fusion of both (UDC+LR). We also compare three clustering methods, namely the *K*-means, constrained clustering (CC) [2] and the proposed PSO-based constrained clustering (PSOCC) method. Table 2 shows that although LR does not improve the F-scores with respect to the UDC in the Bach10 dataset (except for the case of *K*-means), it outperforms the UDC in the Interwoven dataset for all cases. In general, the performance of UDC+LR is better than the UDC while worse than LR. For the clustering method, we found that both CC and PSOCC outperform *K*-means, and the performance of PSOCC and CC are similar.

The results can be explained by two reasons; first is that the UDC is pitch-dependent while LR is not, and second is that imposing the constraints actually prefers *not* to interweave streams having less distinguishable features when interweaving of streams actually occurs. This is why the pitch-dependent UDC outperforms LR in the pitch-ordered Bach10 dataset, while LR outperformed the UDC in the Interwoven dataset. The result of UDC+LR having an intermediate performance between the UDC and LR again explains that when *the notes are highly interwoven, the MPS task is solved with a trade-off between feature similarity and the pitch-linking constraints.* Comparing the clustering method, we found that randomly-initialized PSOCC performs slightly better than pitch-ordered initialized CC in both datasets. Although the improvement is marginal, the effectiveness of this clustering scheme is demonstrated under the same objective function and constraints as CC. In addition, this method can be further improved in the future by using different number of particles and velocity parameters in the optimization.

Fig. 2 illustrates the MPS results of a selected clip using baseline (UDC+CC) [2] and the proposed method ({UDC+LR}+PSOCC). The major difference between the two methods can be illustrated with two interwoven streams, vocal and guitar. The baseline method prefers to keep the pitch order of the streams, namely, keeping the guitar stream higher than the vocal stream. Conversely, the proposed method has more flexibility in assigning stream labels by employing the information of LR, which explains why the proposed method works effectively in those "voice-crossing" events in the clip.

Table 3. Comparison of average F scores (in %) between the baseline and proposed experiment schemes using real-world MPE.

	Duan	CFP
Baseline (UDC+CC)	55.99	68.89
Proposed ({UDC+LR}+PSOCC)	66.21	68.45

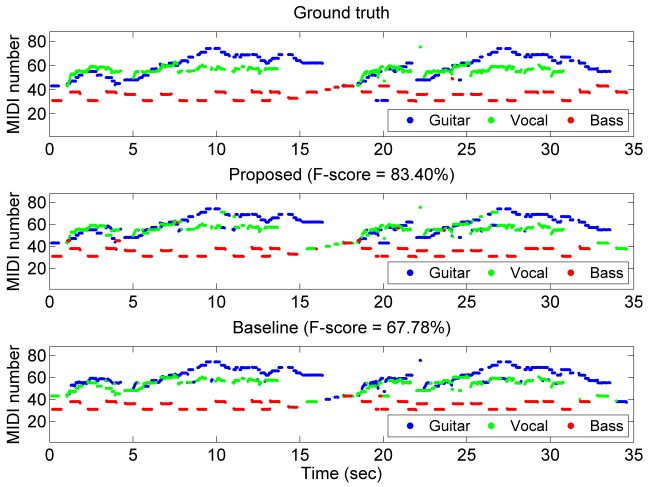

Fig. 2. Illustration of the MPS result of *MusicDelta_Country1*. Top: Ground truth. Middle: baseline. Bottom: proposed.

The second experiment uses the results of real-world MPE algorithms as the input. We consider two MPE algorithms; one is proposed by Duan *et al.* [11], and the other is the combined frequency and periodicity (CFP) method [22]. The average F-scores of MPE on Bach10 dataset are 72.25% for [11] and 76.78% for CFP. Since both methods have an average F-scores lower than 40% on the Interwoven dataset, the error of MPE dominates the result. Therefore, we consider the comparison of MPS on the Interwoven dataset unreliable, and report only the results on Bach10 dataset, see Table 3. One can observe that, for an MPE result having a lower F-score, the proposed method performs better. This is mainly because when frame-level pitch activations are noisy, missing or false alarm pitches can virtually create a scenario similar to interweaving streams.

6. CONCLUSION

To solve the MPS problem of a multi-source signal having interwoven streams, we have presented the level-ratio feature using directional information, and also the PSO-based constrained clustering method providing extra freedom in selecting better solution. Evaluation on recordings having either a high or low interwoven rate gives a promising result, while at the same time reveals a trade-off between feature similarity and the pitch-linking constraint. The performance gaps between interwoven streams and non-interwoven streams, and between ground-truth MPE input and real-world MPE input also suggest more room for improvement in handling this problem. Possible future work can target at how to better exploit timbre and directional information, how to incorporate musical knowledge into a clustering scheme, and how to control the computational complexity.

7. REFERENCES

[1] 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.

[2] Z. Duan, J. Han, and B. Pardo, "Multi-pitch streaming of harmonic sound mixtures," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 1, pp. 138–150, 2014.

[3] G. Grindlay and D. P. W. 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.

[4] M. Karjalainen and T. Tolonen, "Multi-pitch and periodicity analysis model for sound separation and auditory scene analysis," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 1999, vol. 2, pp. 929–932.

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

[6] Y. Liu and D. Wang, "Speaker-dependent multipitch tracking using deep neural networks," in *Proceedings of Interspeech*, 2015, pp. 3279–3283.

[7] Z. Jin and D. Wang, "A multipitch tracking algorithm for noisy and reverberant speech," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 4218–4221.

[8] S. Ananthakrishnan and S. S. Narayanan, "An automatic prosody recognizer using a coupled multi-stream acoustic model and a syntactic-prosodic language model," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2005, vol. 1, pp. 269–272.

[9] D. Deutsch, "Grouping mechanisms in music," in *Psychology of music*, chapter 6, pp. 183–248. Elsevier, 2013.

[10] D. Huron, "The avoidance of part-crossing in polyphonic music: perceptual evidence and musical practice," *Music Perception: An Interdisciplinary Journal*, vol. 9, no. 1, pp. 93–103, 1991.

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

[12] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "Medleydb: A multitrack dataset for annotation-intensive mir research.," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 155–160.

[13] J Kennedy, "Particle swarm optimization," in *Encyclopedia of machine learning*, pp. 760–766. Springer, 2011.

[14] R Poli, J Kennedy, and T Blackwell, "Particle swarm optimization," *Swarm intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[15] M. Bay, A. F. Ehmann, J. W. Beauchamp, P. Smaragdis, and S. Downie J., "Second fiddle is important too: Pitch tracking individual voices in polyphonic music.," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2012, pp. 319–324.

[16] V. Arora and L. Behera, "Musical source clustering and identification in polyphonic audio," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 6, pp. 1003–1012, 2014.

[17] V. Arora and L. Behera, "Multiple F0 estimation and source clustering of polyphonic music audio using PLCA and HMRFs," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 2, pp. 278–287, 2015.

[18] M. Wohlmayr, M. Stark, and F. Pernkopf, "A probabilistic interaction model for multipitch tracking with factorial hidden markov models," *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, no. 4, pp. 799–810, 2011.

[19] V. Arora and L. Behera, "Discriminative PLCA based polyphonic source identification," in *21st European Signal Processing Conference (EUSIPCO)*, 2013, pp. 1–5.

[20] A. Eronen and A. Klapuri, "Musical instrument recognition using cepstral coefficients and temporal features," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2000, pp. 753–756.

[21] Li-Fan Yu, Li Su, and Yi-Hsuan Yang, "Sparse cepstral codes and power scale for instrument identification," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 7460–7464.

[22] L. Su and Y.-H. Yang, "Combining spectral and temporal representations for multipitch estimation of polyphonic music," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 10, pp. 1600–1612, 2015.

[23] D. Campbell, K. Palomaki, and G. Brown, "A matlab simulation of "shoebox" room acoustics for use in research and teaching," *Computing and Information Systems*, vol. 9, no. 3, pp. 48, 2005.

[24] F. Cummins, M. Grimaldi, T. Leonard, and J. Simko, "The chains corpus: Characterizing individual speakers," in *Proc. of SPECOM*, 2006, vol. 6, pp. 431–435.