INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Sinusoidal Partials Tracking for Singing Analysis Using the Heuristic of the Minimal Frequency and Magnitude Difference

Kin Wah Edward Lin[1], *Hans Anderson*[1], *Clifford So*[2] *and Simon Lui*[1]

[1]Singapore University of Technology and Design, Singapore
[2]Chinese University of Hong Kong, Hong Kong

{edward_lin, hans_anderson}@mymail.sutd.edu.sg, cliffso@cuhk.edu.hk, simon_lui@sutd.edu.sg

Abstract

We present a simple heuristic-based Sinusoidal Partial Tracking (PT) algorithm for singing analysis. Our PT algorithm uses a heuristic of minimal frequency and magnitude difference to track sinusoidal partials in the popular music. An Ideal Binary Mask (IBM), which is created from the ground truth of the singing voice and the music accompaniment, is used to identify the sound source of the partials. In this justifiable way, we are able to assess the quality of the partials identified from the PT algorithm. Using the iKala dataset along with the IBM and BSS Eval 3.0 as a new method of quantifying the partials quality, the comparative results show that our PT algorithm can achieve $0.8746 \sim 1.7029$ dB GNSDR gain, compared to two common benchmarks, namely the MQ algorithm and the SMS-PT algorithm. Thus, our PT algorithm can be considered as a new benchmark of the PT algorithm used in singing analysis.

Index Terms: sinusoidal partials tracking, singing analysis, time-frequency masking, global normalized source-to-distortion ratio

1. Introduction

Sinusoidal partials tracking (PT) is a peak-continuation algorithm that organizes the spectral peaks into a set of tracks and each track models a time-varying sinusoid. The tracks, which represent the deterministic part of the audio signal, are called partials. PT algorithms have been explored widely in the audio research field and they have several important applications. Examples include singing pitch extraction [1–3], sound analysis/transformation/synthesis [4, 5], note onset/offset detection [6], bird sound/singing classification [7, 8] and condition monitoring systems for wind turbines [9].

We propose to use both frequency and magnitude heuristics to guide the partials tracking. This is different from the two most widely-used approaches, the MQ algorithm and the SMS-PT algorithm, which use frequency-only heuristics. The MQ algorithm was proposed by McAulay & Quatiery [4] to analyze and synthesize speech. The SMS-PT algorithm is the `sineTracking` function used in the Spectral Modeling Synthesis (SMS) Tools[1]. Its basic model and implementation are developed by X.Serra [5]. Our comparative results (see Section 6) show that, compared to the two common benchmarks mentioned above, the partials that our PT algorithm produces are more likely to belong to either the singing voice or the music accompaniment.

We use the iKala dataset [10], along with the Ideal Binary Mask (IBM) [11] and BSS Eval Version 3.0 [12] to quantitatively evaluate the quality of the partials. The iKala dataset contains 252 publicly available 30-seconds popular music clips in CD quality; 137 of these clips are named as *Verse* and the remaining 115 clips are named as *Chorus*. Each clip is a Wave file with two channels. One is the ground truth singing voice V and the other is the ground truth music accompaniment S. There are 3 female and 3 male singers. Most of the V in the music clips are performed solely by one of these singers, and all S are performed by the professional musicians. Each clip contains non-vocal regions with varying duration. The songs are either in English, Mandarin, Taiwanese or Korean. Here we give a brief description of IBM. Let X be a $F \times T$ matrix that denotes the magnitude spectrogram, where F is the number of frequency bins, $F = (\lfloor \frac{N}{2} \rfloor + 1)$, N is the Discrete Fourier Transform (DFT) size and T is the number of frames. Given the magnitude spectrogram of the voice X_V and of the music accompaniment X_S, the $F \times T$ matrix B, which is the IBM of the singing voice is calculated as,

$$B[n,t] = \begin{cases} 1, & \text{if } X_V[n,t] > X_S[n,t] \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where $t \in [1, T]$ is the time index and $n \in [1, F]$ is the frequency bin index. Let $\overline{B} = |1 - B|$ denotes the IBM of the music accompaniment. Given B and $\overline{B}$, we can identify the sound source of each partial (see Section 5). BSS Eval Version 3.0 is the standard quality assessment tool in the research field of singing voice separation. The separation quality of the singing voice for each clip can be accessed by Source to Distortion Ratio (SDR) [12]. The overall separation quality of the singing voice for each clip is determined by the normalized SDR (NSDR), which is calculated as

$$\text{NSDR}(\overline{V}, V, M) = \text{SDR}(\overline{V}, V) - \text{SDR}(M, V) \quad (2)$$

where $\overline{V}$ is the audio signal of the separated singing voice and M is the mixture signal of the ground truth singing voice and the ground truth music accompaniment. The overall separation quality of the singing voice is then determined by the global NSDR (GNSDR), which is calculated as

$$\text{GNSDR} = \frac{1}{|C|} \sum_{i \in C} \text{NSDR}(\overline{V_i}, V_i, M_i) \quad (3)$$

where C is a set of test clips, $|C|$ is the total number of the test clips. The larger the GNSDR, the better the separation quality. The overall separation quality of the music accompaniment is calculated similarly as mentioned above.

The contributions of this paper are:

• Our simple heuristic-based PT algorithm can achieve $0.8746 \sim 1.7029$ dB GNSDR gain, compared to two common benchmarks. Thus, our PT algorithm can be considered as a new benchmark of the PT algorithm used in singing analysis.

• Our evaluation approach provides a better method of quantifying how well a PT algorithm distinguishes the deterministic part of the singing voice from its music accompaniment.

[1]http://mtg.upf.edu/technologies/sms

http://dx.doi.org/10.21437/Interspeech.2017-17

2. Motivation

We use the same preprocessing and synthesis methods for all PT algorithms, in order to have a fair comparison. We use the Short-Time Fourier Transform (STFT) with spectral peak detection as the only preprocessing procedure. The resulting spectral peaks are the input for all PT algorithms. The output of all PT algorithm is a set of partials. Given this output, we carry out the procedure described in Section 5 to create a binary time-frequency mask for the singing voice and the music accompaniment. Then we use the additive synthesis [13] to resynthesize the audio signal of the singing voice and of the music accompaniment. Finally, we calculate their GNSDRs as in (3).

Let us explain why we only consider STFT for the spectrogram transform. Researchers in the field of predominant melody estimation often use the multi-resolution magnitude spectrogram transform [1] or frequency/amplitude correction (e.g. phase vocoder [2]) as additional preprocessing procedures before applying the PT algorithm. J.Salamon et al. [14] show that these preprocessing procedures contribute only a minimal improvement over the basic STFT[2]. One may argue it is a study in the context of the melody extraction. However, the singing voice in the popular music is the melody. Thus, melody peak and energy recall are the most important features of the PT algorithm in the context of singing analysis.

Our evaluation method is more objective than using reference partials, as some previous works have done. M.Lagrange et al. [6] propose a PT algorithm dedicated to the analysis of polyphonic sounds. They claim that their algorithm strives to ensure that each partial belongs to only one sound source but they did not quantitatively evaluate that claim. Instead they show that their PT algorithm produces partials which are closer to the reference partials and are immune to noise introduced by the sidelobes of the window function [15] and to noise they artificially add. Leonardo O. Nunes et al. [16] create a database for evaluating the partials tracking algorithms. However, there are two drawbacks to the use of reference partials as they propose. First, the audio signals synthesized from the reference partials are artificial. Second, there is no way to verify that the reference partials are optimal or are the only optimal solution.

In fact, the GNSDR is a suitable measure to quantify the quality of a PT algorithm. Note that, all PT algorithms are tested on the same ground truth sound source, which is unknown to the PT algorithm. All PT algorithms strives to link up the peaks of the same source and to lengthen the tracks so that the tracks would not be dropped out and can be included into the set of partials. The GNSDR is then a suitable measure to quantify how clearly each partial belongs to one and only one sound source and how much such good partials are retained.

We compare our PT algorithm with the MQ and SMS-PT algorithms, which are the common and widely used approaches, applied either directly or with some modifications [2, 3, 7–9, 16, 17]. We omit an objective comparison of our PT algorithm with some recent PT algorithms (e.g. [6]), mainly because we do not have a justifiable way to give a ground truth labelling on the partials produced from these PT algorithms. These algorithms model the trajectory of the sinusoidal partial to give a better time/frequency resolution. However, such process adds, drops or even modifies the time-frequency (T-F) bin and the magnitude of the peak. Thus, it prevents us from giving a justifiable way of labelling the partial type.

[2]The SFTF alone has the highest peak recall and energy recall, which are 0.62 and 0.88 respectively. Other evaluation metrics of the STFT are not significant worst.

3. Preprocessing Procedure

Here we describe the preprocessing procedure in detail. For a given audio clip in the iKala dataset, we first mix the ground truth singing voice V and the ground truth music accompaniment S at 0 dB to form the combined music signal $M = V + S$. Then we apply the STFT to M to obtain the magnitude spectrogram X_M and the phase spectrogram A_M. The spectral peaks for X_M are indicated by a $F \times T$ matrix P which is called a binary time-frequency mask and is calculated as follows,

$$P[n,t] = \begin{cases} 1, & \text{if } X_M[n,t] > X_M[n-1,t] \\ & \text{and } X_M[n,t] \geq X_M[n+1,t] \\ 0, & \text{otherwise,} \end{cases} \quad (4)$$

Given the T-F bin $[n,t]$ of a spectral peak $P[n,t]$, we define $\text{Freq}(P[n,t]) = (n-1)\frac{f_S}{N}$ to be the frequency of the spectral peak in Hz, where f_S is the sampling frequency fixed at 22.05 kHz. We also define $\text{Mag}(P[n,t]) = X_M[n,t]$ to be the magnitude of the spectral peak. To simplify the notation, we denote p to represent a peak and assume it contains the T-F bin of the peak. Hence, we write Freq(p) and Mag(p) to indicate the frequency and magnitude of a peak p respectively.

In order to preserve the spectral information as much as possible, we have tested 6 configurations of the magnitude spectrogram shown in Table 1, to be applied before we pass the spectral peaks to the PT algorithm. Note that the choice for the values of the DFT size, the window size and the hop size in STFT Conf.1,2,5 and 6 are common (e.g. [10, 14, 18]), so as the choice for the zero padding factor (e.g. [2]). A $4\times$ zero padding is applied in STFT Conf.2, 4 and 6. The window size in STFT Conf.3 and 4 are chosen in an awkward size because T.S. Chan et al. [10] reports the separation quality with those sizes.

Table 1: *STFT Configurations & its GNSDRs, f_S =22.05 kHz.*

STFT Conf.	Hann Window Size	DFT Size	Hop Size	GNSDRs $\overline{V}$	GNSDRs $\overline{S}$
1	1024	1024	256	8.2715	11.2369
2	**1024**	**4096**	**256**	**9.5284**	**12.5556**
3	1411	1411	353	5.7336	11.4674
4	1411	5644	353	6.4065	12.4727
5	2048	2048	512	6.0247	12.1775
6	2048	8192	512	6.8736	13.1033

We carry out the following procedures to calculate the GNSDR of each STFT configuration. With the help of B and $\overline{B}$, we can classify each spectral peak P as representing either predominantly voice sound or predominantly musical accompaniment. To gather all the spectral peaks that represent predominantly the singing voice, we do an element-wise multiplication of P with B. Similarly, the elementwise product $P \circ \overline{B}$ gives the spectral peaks belonging predominantly to the musical accompaniment. We obtain the separated singing voice $\overline{V}$ and the separated music accompaniment $\overline{S}$ by calculating the additive synthesis [13] of $\text{Mag}(P) \circ B$ and $\text{Mag}(P) \circ \overline{B}$ using A_M. Processing $\overline{V}$, $\overline{S}$, V and S through (2), we obtain the NSDR of $\overline{V}$ and $\overline{S}$ for a clip. Finally, we calculate the GNSDR of the singing voice and of the music accompaniment as in (3). We choose STFT Conf.2 because it produces the highest GNSDR for both sets $\overline{V}$ and $\overline{S}$.

Before we use the matrix P as the input for all PT algorithms, we need to filter out some peaks. First, we classify each peak p into L magnitude levels as follows,

$$\text{Level}(p) = \lfloor (L-1) \times \frac{\text{Mag}(p) - X_M^{min}}{X_M^{max} - X_M^{min}} \rfloor + 1. \quad (5)$$

Let X_M^{max} and X_M^{min} denote the maximum and minimum magnitude of X_M respectively, where $X_M^{min} \geq 2^{-52}$. Then we filter out the peaks whose Level(p) is lower than a level E, by setting the corresponding $P[n,t]$ from one to zero. This process is repeated in a grid search until the parameters, which achieve almost the same GNSDR of the set $\overline{V}$ and of the set $\overline{S}$ mentioned above, are found. We set L to 64 and we set E to 42. Based on STFT Conf.2, the number of frequency bins is $(\lfloor \frac{4096}{2} \rfloor + 1) = 2049$ and the number of frames is $\lceil (22,050 \times 30)/256 \rceil = 2584$, then the total number of T-F bins is $2049 \times 2584 = 5,294,616$. The average number of peaks is decreased from 436,968.49 (8.2531%) to 198,460.71 (3.7483%).

4. Heuristic based Partial Tracking

In this section, we first explain how our PT algorithm works, then we explain the differences between our PT algorithm and other approaches.

Given the spectral peaks P as the input, our PT algorithm creates tracks on a frame-by-frame basis. The following description summarizes the operation at frame t. If $t = 1$ or there is no track having a peak in frame $t-1$, we skip the steps below and create a new track for each peak at frame t.

1. Sort the peaks in descending order of magnitude.
2. Find a set of tracks which have a peak i at frame $t-1$.
3. For each sorted peak j, we select a set of tracks among the tracks found in step 2 such that
   (a) The tracks are not previously selected at frame t.
   (b) The tracks from $P[i,(t-1)]$ and the peak $P[j,t]$ are within the frequency and magnitude difference limits,
   $$|\text{Freq}(P[i,(t-1)]) - \text{Freq}(P[j,t])| < \theta_f,$$
   $$20 \times |\log_{10}(\frac{\text{Mag}(P[i,(t-1)])}{\text{Mag}(P[j,t])})| < \theta_m, \quad (6)$$
   where θ_f and θ_m are thresholds that prevent large sudden changes in frequency and magnitude within a track. θ_m is a constant; θ_f is defined as
   $$\theta_f := \Delta_f \times \text{Freq}(P[j,t]) + f \quad (7)$$
   where Δ_f is the slope increase of the minimum frequency deviation and f is the minimum frequency deviation at 0 Hz. Both Δ_f and f are constant values.
4. If such set of tracks exists, among this set of tracks, we assign the peak to the track that minimizes their frequency difference.
5. Return to Step 3 to examine the next sorted peak. When all sorted peak are examined, go to Step 6.
6. Create a new track for each non-assigned peak.

After carrying out the steps above for each frame in the spectrogram, we obtain a set of tracks. We take those tracks which have at least the minimum length θ_p and include them into the set of partials R. We set θ_p to 4 which is the default value of the SMS-PT algorithm. With STFT Conf.2, this requirement implies that each partial is at least 46.44 ms long.

Our PT algorithm is an enhancement of the SMS-PT algorithm. The key difference is our addition of the magnitude difference threshold in (6). Similarly, the SMS-PT algorithm can also be considered as an enhancement of the MQ algorithm because the SMS-PT algorithm adds to the MQ algorithm a more flexible frequency difference threshold as shown in (7).

5. Synthesis Procedures

Here we describe how we process the partials R obtained from the PT algorithms for the quality evaluation. First, we use B and $\overline{B}$ to identify the type of each partial $r \in R$. Since each r is a time-series of peaks, we can compare the T-F bin of each peak with B and $\overline{B}$ to find out how many peaks in R belong to the singing voice and how many peaks belong to the music accompaniment. If the number of peaks belonging to the singing voice is larger than the numbers of peaks belonging to the music accompaniment, we identify r as a partial of the singing voice. Otherwise, it belongs to the music accompaniment. Then, given a set of partials belonging to the singing voice, we create a binary time-frequency mask $Y_{\overline{V}}$ to represent the separated singing voice. Similarly, we create another binary time-frequency mask $Y_{\overline{S}}$ to represent the separated music accompaniment. Finally, we calculate the additive synthesis [13] of $Y_{\overline{V}} \circ X_M$ and $Y_{\overline{S}} \circ X_M$ using the phase spectrogram A_M to obtain the audio signal of the separated singing voice and the separated music accompaniment respectively. The synthesis window size we use is 512, the FFT size is 1024, and the hop size is 256. The synthesis window is a triangular window divided by a normalized Blackman-Harris 92 dB window.

6. Experimental Results

Now we are ready to illustrate the advantages of our PT algorithm which we call it the FM algorithm. Table 2 lists the parameters which we use to configure the PT algorithms. Each parameter combination gives one analysis configuration of a PT algorithm. Given one analysis configuration and one audio clip, after we carry out the steps in the previous 3 sections, the PT algorithm produces one NSDR for $\overline{V}$ and one NSDR for $\overline{S}$. With the parameters we choose above, the MQ algorithm and the SMS-PT algorithm produce 10 GNSDRs for $\overline{V}$ and for $\overline{S}$; the FM algorithm produces 100 GNSDRs for $\overline{V}$ and for $\overline{S}$.

Table 2: *Analysis Configurations of each PT algorithm.*

Algorithm	**Parameters**	**Number Of Conf.**
FM	$f := 5, 10, 15, 20, 25, 30, 35, 40, 45, 50$ $\Delta_f := 0.01$ $\theta_m := 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$	100
SMS-PT	$f := 5, 10, 15, 20, 25, 30, 35, 40, 45, 50$ $\Delta_f := 0.01$	10
MQ	$\theta_f := 5, 10, 15, 20, 25, 30, 35, 40, 45, 50$	10

Figure 1 plots the GNSDRs of each PT algorithm against the frequency threshold (or the constant frequency deviation). Note that, for the plot of the FM algorithm, we set θ_m to 4 for each step of f, as it produces the highest GNSDRs of the singing voice for both the *Verse* and *Chorus* clips. These two plots show that our PT algorithm outperforms the other approaches. Based on the best GNSDR of the singing voice, for our FM algorithm, we set f to be 30 Hz; for the SMS-PT algorithm, we set f to be 10 Hz; for the MQ algorithm, we set θ_f to be 20 Hz. Comparing the best GNSDRs across all 3 PT algorithms, our PT algorithm achieves $0.9542 \sim 1.7029$ dB and $0.8746 \sim 1.6252$ dB GNSDR gain in the singing voice and the music accompaniment respectively. To confirm the statistical significance of these results, we perform a one-way ANOVA to assess whether the GNSDR differences between each pair of the algorithms are significant. Table 3 summaries the ANOVA result. It shows our FM algorithm achieves a statistical significant GNSDR difference ($p < 0.05$).

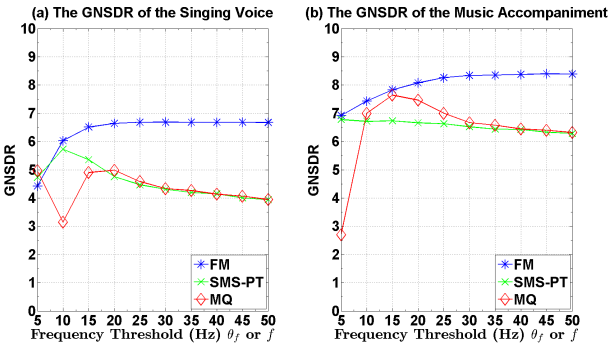


Figure 1: *The GNSDRs of each PT algorithm. The ideal GNSDR of the singing voice and the music accompaniment are* 9.5122 *dB and* 12.5395 *dB respectively.*

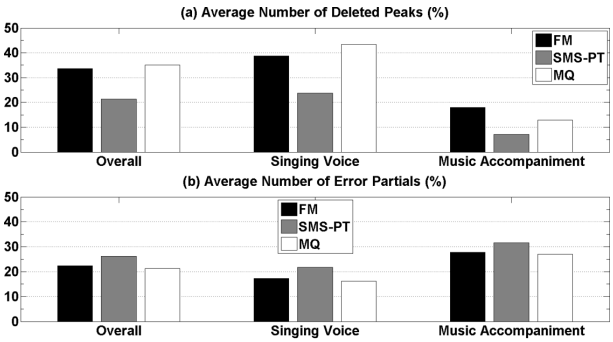
Table 3: *One-way ANOVA result for the significant difference of GNSDR between each pair of the PT algorithms.*

Pair	Singing Voice		Music Accompaniment	
	F(1,502)	p-value	F(1,502)	p-value
FM,SMS-PT	7.3439	0.0070	14.0864	0.0002
FM,MQ	22.8218	2.34×10^{-6}	5.0942	0.0244
SMS-PT,MQ	4.1748	0.0416	4.0706	0.0442

Although our FM algorithm actually add a new magnitude constraint on the partials tracking, and there are partial crossing and overlapping[3], this constraint is more likely to make the track of the singing voice to be extended by a peak of the singing voice, and therefore to make the track long enough to be a partial. Comparing our FM algorithm to the ideal GNSDRs, we see that the 2.8284 dB GNSDR loss in the singing voice is lower than the 4.2006 dB GNSDR loss in the music accompaniment. This result supports our argument. Another way to support this argument is to look at (i) the average number of peaks which are deleted by the PT algorithms, and (ii) the average number of error partials which have both type of peaks. Figure 2 shows both results for each PT algorithm. Our FM algorithm deletes a lot more peaks than the SMS-PT algorithm, but it deletes fewer peaks belonging to the singing voice than the MQ algorithm. Our FM algorithm has fewer error partials by percentage than the SMS-PT algorithm, but it has a slightly more error partials than the MQ algorithm. As our FM algorithm does not delete too many peaks belonging to the singing voice and it also does not have too many error partials belonging to the singing voice, our FM algorithm is more likely to reveal the singing partials.

To understand why the additional constraint does help to track the evolution of the singing partials, we start to investigate the normalized vibrato of each partials produced from each PT algorithm. It has been shown that vibrato is a distinctive and musically meaningful element in the singing voice [1, 19, 20]. We first calculate the vibrato frequency and width of each partial similarly to the method in [21], then we linearly scale both values in each partial to the range of $[+1, -1]$. Figure 3 shows an example of the normalized vibrato frequency versus the normalized vibrato width. Our FM algorithm produces more fairly separable partials than the other algorithms. The others even produce some partials whose vibrato characteristics are overlapping. This result suggests our FM algorithm is more likely to reveal the vibrato characteristic of the singing voice from the accompaniment.

[3]Singer may sing notes that are present in the accompaniment.



Ave. Number of	Overall	Singing voice	Music Accompaniment
Peaks in PT input	198,460.71	100,922.05	97,538.66
Partials by FM	14,939.47	7,739.98	7,199.49
Partials by SMS-PT	14,457.49	7,891.92	6,565.57
Partials by MQ	14,610.82	7,718.75	6,892.07

Figure 2: *The partial quality of each PT algorithm.*

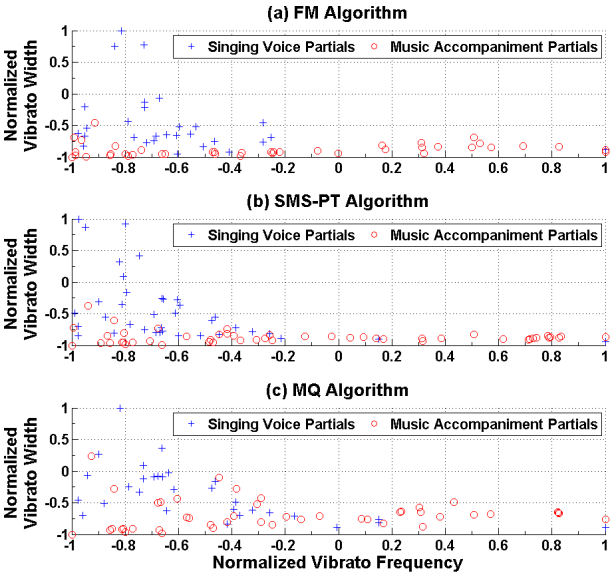


Figure 3: *The normalized vibrato frequency versus the normalized vibrato width. Each partials is produced by each PT algorithm with 21054_chorus.wav. The maximum magnitude level of each partial is at 60.*

7. Conclusion and Future Work

We presented a simple sinusoidal partial tracking algorithm using a threshold of frequency and magnitude deviation. Using the iKala dataset along with the Ideal Binary Masks and BSS Eval Version 3.0 as a new method of quantifying the partials quality, we show that our PT algorithm is better at identifying the deterministic part of the singing voice in the polyphonic music than two common approaches in the literature. Some audio samples and spectrogram plots with partials highlight are available at http://people.sutd.edu.sg/~1000791. In the future work, we will investigate (i) a justifiable way of labelling the partial type, so that we are able to compare our PT algorithm with the recent PT algorithms, and (ii) the partials classification for solving the problem of the singing voice separation.

This work is supported by the MOE Academic fund AFD 05/15 SL.

8. References

[1] C.-L. Hsu and J.-S. R. Jang, "Singing pitch extraction by voice vibrato/tremolo estimation and instrument partial deletion," in *International Society for Music Information Retrieval Conference (ISMIR)*, Aug 2010, pp. 525–530.

[2] J. Salamon and E. Gómez, "Melody extraction from polyphonic music signals using pitch contour characteristics," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, pp. 1759–1770, Aug 2012.

[3] A. Degani, R. Leonardi, P. Migliorati, and G. Peeters, "A pitch salience function derived from harmonic frequency deviations for polyphonic music analysis," in *International Conference on Digital Audio Effects (DAFx)*, Sep 2014, pp. 195–201.

[4] R. McAulay and T. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 4, pp. 744–754, Aug 1986.

[5] X. Serra, "A system for sound analysis / transformation / synthesis based on a deterministic plus stochastic decomposition," Ph.D. dissertation, Stanford University, 1989.

[6] M. Lagrange, S. Marchand, and J. B. Rault, "Enhancing the tracking of partials for the sinusoidal modeling of polyphonic sounds," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1625–1634, Jul 2007.

[7] Z. Chen and R. C. Maher, "Semi-automatic classification of bird vocalizations using spectral peak tracks," *The Journal of the Acoustical Society of America*, vol. 120, no. 5, pp. 2974–2984, 2006.

[8] J. R. Heller and J. D. Pinezich, "Automatic recognition of harmonic bird sounds using a frequency track extraction algorithm," *The Journal of the Acoustical Society of America*, vol. 124, no. 3, pp. 1830–1837, 2008.

[9] T. Gerber, N. Martin, and C. Mailhes, "Time-frequency tracking of spectral structures estimated by a data-driven method," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 10, pp. 6616–6626, Oct 2015.

[10] T. Chan, T. Yeh, Z. Fan, H. Chen, L. Su, Y. Yang, and R. Jang, "Vocal activity informed singing voice separation with the ikala dataset," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr 2015, pp. 718–722.

[11] D. Wang, *On Ideal Binary Mask As the Computational Goal of Auditory Scene Analysis*. Springer US, 2005, pp. 181–197.

[12] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 4, pp. 1462–1469, Jul 2006.

[13] U. Zolzer, *DAFX: Digital Audio Effects*, 2nd ed. Wiley Publishing, 2011, pp. 406-411.

[14] J. Salamon, E. Gómez, and J. Bonada, "Sinusoid extraction and salience function design for predominant melody estimation," in *International Conference on Digital Audio Effects (DAFx)*, Sep 2011, pp. 73–80.

[15] M. Lagrange and S. Marchand, "Assessing the quality of the extraction and tracking of sinusoidal components: Towards an evaluation methodology," in *International Conference on Digital Audio Effects (DAFx)*, Sep. 2006, pp. 239–245.

[16] L. O. Nunes, L. W. Biscainho, and P. A. Esquef, "A database of partial tracks for evaluation of sinusoidal models," in *International Conference on Digital Audio Effects (DAFx)*, Sep 2010, pp. 1–8.

[17] D. Lloyd, N. Raghuvanshi, and N. K. Govindaraju, "Sound synthesis for impact sounds in video games," in *Symposium on Interactive 3D Graphics and Games (I3D)*, Feb 2011, pp. 55–62.

[18] Z. C. Fan, J. S. R. Jang, and C. L. Lu, "Singing voice separation and pitch extraction from monaural polyphonic audio music via dnn and adaptive pitch tracking," in *2016 IEEE Second International Conference on Multimedia Big Data (BigMM)*, April 2016, pp. 178–185.

[19] L. Regnier and G. Peeters, "Singing voice detection in music tracks using direct voice vibrato detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2009, pp. 1685–1688.

[20] J. Salamon, B. Rocha, and E. Gómez, "Musical genre classification using melody features extracted from polyphonic music signals," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar 2012.

[21] P. Herrera and J. Bonada, "Vibrato extraction and parameterization in the spectral modeling synthesis framework," in *International Conference on Digital Audio Effects (DAFx)*, Nov 1998.