MULTI-LAYERED CEPSTRUM FOR INSTANTANEOUS FREQUENCY ESTIMATION

Chin-Yun Yu and Li Su

Institute of Information Science, Academia Sinica, Taiwan

ABSTRACT

We propose the multi-layered cepstrum (MLC) method to estimate multiple fundamental frequencies (MF0) of a signal under challenging contamination such as high-pass filter noise. Taking the operation of cepstrum (i.e., Fourier transform, filtering, and nonlinear activation) recursively, MLC is shown as an efficient method to enhance MF0 saliency in a step-by-step manner. Evaluation on a real-world polyphonic music dataset under both normal and low-fidelity conditions demonstrates the potential of MLC.

Index Terms— Cepstrum, multiple fundamental frequency estimation, time-frequency analysis.

1. INTRODUCTION

Multiple fundamental frequency (MF0) estimation is an essential problem in analyzing various multi-component signals including multi-talker speech [1,2], polyphonic music [3], fetal electrocardiogram (FECG), photoplethysmographic (PPG) signals [4], and IoT sensor data [5], to name but a few. Numerous methods have been proposed to enhance the saliency of F0s in the signal representation, such as the autocorrelation function (ACF) [6], generalized cepstrum [7], nonnegative matrix factorization (NMF), MUltiple SIgnal Classification (MUSIC) [8], combined frequency and periodicity (CFP) [9–11], and deep learning approaches [12]. The generalized cepstrum, being one of the earliest proposed MF0 estimation methods [7], still gains attention in recent years, by showing its effectiveness in suppressing the unwanted harmonics and in localizing the F0 components by combining it with a spectrum [9,10,13]. A recent study further proved that a cepstrum can perfectly localize the instantaneous F0s of a signal with the intrinsic mode function (IMF) type [11], a general type of non-stationary oscillatory signals.

One issue which is relatively less discussed in MF0 estimation is the so-called *convolutional* noise in signals [14]. Convolutional noise is ubiquitous; it occurs when signals are transmitted through an intermediate channel or device. An MF0 algorithm usually fails in the presence of convolutional noise, for example, in the processing of audio signals played by a smartphone, as the response of a smartphone speaker typically behaves like a high-pass filter whose cutoff frequency is around 500 Hz [9, 15]. Such a high-pass filter suppresses the true F0 peaks, as similar to the case of the "missing fundamental" effects in low-pitched musical signals [16]. In such cases, to identify the true F0 peaks merely from the interleaved harmonics of multiple components is challenging.

In this paper, we generalize the notion of cepstrum to resolve the above-mentioned issue in MF0 estimation. The proposed *multi-layered cepstrum* (MLC) refines the saliency of the F0s in a recursive manner. In a nutshell, the MLC takes the operation of cepstrum (i.e., Fourier transform, filtering and nonlinear activation) repeatedly: for a time-domain input signal $\mathbf{x} \in \mathbb{R}^N$, $l \geq 0$, the lth-layer output of MLC is:

$$\mathbf{z}^{(0)} = \sigma^{(0)}\left(|\mathbf{Fx}|\right) , \quad \mathbf{z}^{(l)} = \sigma^{(l)}\left(\mathbf{W}^{(l)}\mathbf{Fz}^{(l-1)}\right) ,$$

where $\mathbf{F} \in \mathbb{C}^{N \times N}$ is the N-point discrete Fourier transform (DFT) matrix, $\mathbf{W} \in \mathbb{R}^{N \times N}$ is a high-pass filter, and $\sigma(\cdot)$ is an element-wise power-scaled rectification unit (see Section 3). In the following sections, we will explain how the idea of MLC is motivated, how it works in MF0 estimation, and its structural resemblance to multi-layered perceptrons (MLP). We will then focus on combining two MLCs lying in both the frequency and time domains to achieve MF0 estimation by employing the CFP approach. Experiments on both synthetic and real data demonstrates the effectiveness of MLC in MF0 estimation in both normal and contaminated conditions, and indicate positively the fact that MLC performs better than other methods based on a 'shallow' DFT.

2. MOTIVATION OF MLC

The *cepstrum* $\hat{\mathbf{z}} := \mathbf{F}^{-1}\sigma(\mathbf{Fx})$ of $\mathbf{x}$ [17, 18] is a classic method to estimate the F0 of $\mathbf{x}$. A logarithm cepstrum employing $\sigma(x) := \log x$ is commonly used in single F0 estimation, while the *generalized cepstrum* (GC) employing $\sigma(x) := x^\gamma$, $0 < \gamma \leq 2$, is found more applicable in MF0 estimation, probably because it is numerically more stable than using the logarithm function and is more robust to noise [19, 20]. Cepstrum-based MF0 estimation is based on an assumption that the fast-varying components in a spectrum (e.g., harmonic sequences) are relevant to F0s, while the slow-varying ones (e.g., spectral envelope) are irrelevant to F0s. Saliency of F0s can be enhanced in the *quefrency*[1]

---

[1]Since the quefrency has the same unit as time, in this paper, the terms quefrency and time are used interchangeably.

domain by taking another Fourier transform and a high-pass filter with a cutoff quefrency at q_c on a nonlinearly-stretched spectrum, where the nonlinear operation is to fit the magnitude of the spectrum to humans' perception scale.

To motivate the MLC, we revise the above-mentioned assumption to that 1) the fast-varying and *periodic* components in a spectrum are relevant to F0s while others are not, and that 2) such periodic components still remain periodic (and therefore fast-varying) after a Fourier transform, which is a basic property of DFT. These assumptions allow us to consider repeating the operation of cepstrum on $\mathbf{x}$ to iteratively discard the slow-varying or aperiodic components and to purify the F0-relevant components, which should remain after a number of filtering processes. This idea has been preliminarily verified in [21], where the *generalized cepstrum of spectrum* (GCoS), being the 2nd-layer output of the MLC, behaves as a more succinct feature indicating the F0 peaks than a spectrum. This paper is an extension of [21] to a multi-layered time-frequency analysis tool.

Admittedly, the above-mentioned idea is also motivated by deep learning, which catches wide attention recently in MF0 estimation [12, 22–25]. Not only achieving state-of-the-art results, deep learning also provides inspiring observation when solving the MF0 estimation problem. For example, results in [12] show that the learned networks for melody tracking resemble traditional pitch detection methods: the first layer behaves like a spectral analyzer, while the second layer behaves like a comb filter. It is again intriguing to consider a multi-layered network for MF0 estimation, where the well-known pitch detection blocks such as the Fourier transform and filters are taken directly as subnetworks.

3. THE MLC-CFP METHOD

We formulate the MLC of a non-stationary signal $\mathbf{x}$ in the time-frequency or time-quefrency domain. Given a window function $\mathbf{h} \in \mathbb{R}^N$ and hop size H, the short-time Fourier transform (STFT) of $\mathbf{x}$, $\mathbf{X} := \mathbf{X}[k,n]$, is represented as $\mathbf{X} := \mathbf{X}[k,n] = \sum_{m=0}^{N-1} \mathbf{x}[m+nH]\mathbf{h}[m]e^{-\frac{j2\pi km}{N}}$, where $\mathbf{X} \in \mathbb{C}^{N \times M}$, M is the number of frames, k is the frequency index, and n is the time index. Denote $|\mathbf{X}|$ the element-wise absolute value of $\mathbf{X}$. The lth-layer short-time MLC is:

$$\mathbf{Z}^{(0)} := \sigma^{(0)}\left(|\mathbf{X}|\right), \quad \mathbf{Z}^{(l)} := \sigma^{(l)}\left(\mathbf{W}^{(l)}\mathbf{F}\mathbf{Z}^{(l-1)}\right), \quad (1)$$

where $\mathbf{Z}^{(l)} \in \mathbb{R}^{N \times M}$, $\mathbf{Z}^{(l)} := \mathbf{Z}^{(l)}[k,n]$ for even l, $\mathbf{Z}^{(l)} := \mathbf{Z}^{(l)}[q,n]$ for odd l, and q is the quefrency index. $\sigma^{(l)}$ is a nonlinear, element-wise power function such that for $\gamma_l > 0$, $\sigma^{(l)}(x) = x^{\gamma_l}$ for $x > 0$, and $\sigma^{(l)}(x) = 0$ for $x \leq 0$. With the N-point DFT matrix $\mathbf{F}$, the frequency resolution of $\mathbf{Z}^{(l)}[k,n]$ is f_s/N per bin, and the time resolution of $\mathbf{Z}^{(l)}[q,n]$ is then $1/f_s$ per bin. $\mathbf{W}^{(i)} \in \mathbb{R}^{N \times N}$ represents a high-pass filter; it filters out the elements lying below a cutoff frequency (quefrency) i_c. More precisely, $\mathbf{W}^{(l)}$ is a diagonal matrix

such that $\mathbf{W}^{(l)}[i,i] = 1$ when $i > i_c$ and $i < N - i_c$, while $\mathbf{W}^{(l)}[i,i] = 0$ otherwise. For even l, $i_c := k_c$ defines the cutoff frequency at $f_c = k_c f_s/N$, and for odd l, $i_c := n_c$ defines the cutoff quefrency at $q_c = n_c/f_s$. Also note that since $\mathbf{Z}^{(l)}[:,n]$ is even for $l > 2$, we use $\mathbf{F}$ instead of $\mathbf{F}^{-1}$ when transforming from the frequency domain to the time domain. Many of the basic pitch detection functions are essentially special cases of (1). For example, $\mathbf{Z}^{(0)}$ is the STFT of $\mathbf{x}$, and $\mathbf{Z}^{(1)}$ is known as the autocorrelation function (ACF) for $\gamma_0 = 2$ and $\gamma_1 = 1$. (1) also resembles a DNN structure with zero bias vector: $\mathbf{W}^{(l)}\mathbf{F}$ resembles the fully-connected layer and $\sigma^{(l)}$ resembles the activation function.

To enhance the true F0 components and suppress the high-order harmonics, consider the last two output layers, $\mathbf{Z}^{(l_e)}$ and $\mathbf{Z}^{(l_o)}$, of an MLC, where l_e is even and l_o is odd. $\mathbf{Z}^{(l_e)}$ is in the frequency domain and $\mathbf{Z}^{(l_o)}$ is in the quefrency domain. Based on the CFP approach, we utilize the duality between them to enhance the saliency of F0 by employing a nonlinear mapping to $\mathbf{Z}^{(l_o)}$ from the quefrency domain to frequency domain, and then multiplying it by $\mathbf{Z}^{(l_e)}$ [9–11, 13]:

$$\mathbf{Y}^{(l_e,l_o)}[k,n] = \mathbf{Z}^{(l_e)}[k,n]\mathbf{Z}^{(l_o)}\left[\lfloor N/k \rfloor, n\right] \quad (2)$$

where the time-quefrency representation $\mathbf{Z}^{(l_o)}[k,n]$ is nonlinearly mapped into the frequency domain, and $\lceil \cdot \rfloor$ is the rounding function. Since $\mathbf{Z}^{(l_o)}$ suppresses the harmonic peaks in $\mathbf{Z}^{(l_e)}$ while $\mathbf{Z}^{(l_e)}$ suppresses the sub-harmonic peaks in $\mathbf{Z}^{(l_o)}$, the resulting CFP representation $\mathbf{Y}^{(l_e,l_o)} := \mathbf{Y}^{(l_e,l_o)}[k,n]$ has only true F0 peaks enhanced. The nonlinear mapping of $\mathbf{Z}^{(l_o)}$ in (2) can be implemented by a filterbank following a custom-defined indexes of k; see [10, 21] for details. We name the above process as the MLC-CFP algorithm hereafter.

4. NUMERICAL RESULTS

We illustrate the effectiveness of the MLC-CFP method in emphasizing F0 peaks of a simulated signal contaminated with both additive and convolutional noise. Given two components, $\mathbf{x}_1$ and $\mathbf{x}_2$, both are sampled at $f_s = 1$ kHz. $\mathbf{x}_1$ is a square wave having 20% duty cycle, and has a constant F0 at 2 Hz. $\mathbf{x}_2$ is a frequency-modulated sawtooth wave with positive ramp and vertical drop, and has time-varying F0 represented as $f_0[n] = 2.5 + \cos(2\pi n/10 f_s)$. To simulate the effect of convolutional noise, $\mathbf{x}_1$ is passed through a 10th-order Butterworth low-pass filter, and $\mathbf{x}_2$ is passed through a 10th-order Butterworth high-pass filter. Both filters have a cutoff frequency at 10 Hz. We consider the 2-layer MLC of two sample signals: the first one is $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$, which is with convolutional noise only, while the second one, $\tilde{\mathbf{x}} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{n} + \mathbf{d}$, is further contaminated with additive pink noise and impulse noise, where $\mathbf{n}$ is additive pink noise making the SNR of $\tilde{\mathbf{x}}$ be 10 dB and $\mathbf{d}$ is an impulse at the 80th second.

Fig. 1 shows the STFT ($\mathbf{Z}^{(0)}$), GC ($\mathbf{Z}^{(1)}$), GCoS ($\mathbf{Z}^{(2)}$), and CFP representation ($\mathbf{Y}^{(2,1)}$) of $\mathbf{x}$ and $\hat{\mathbf{x}}$, respectively.
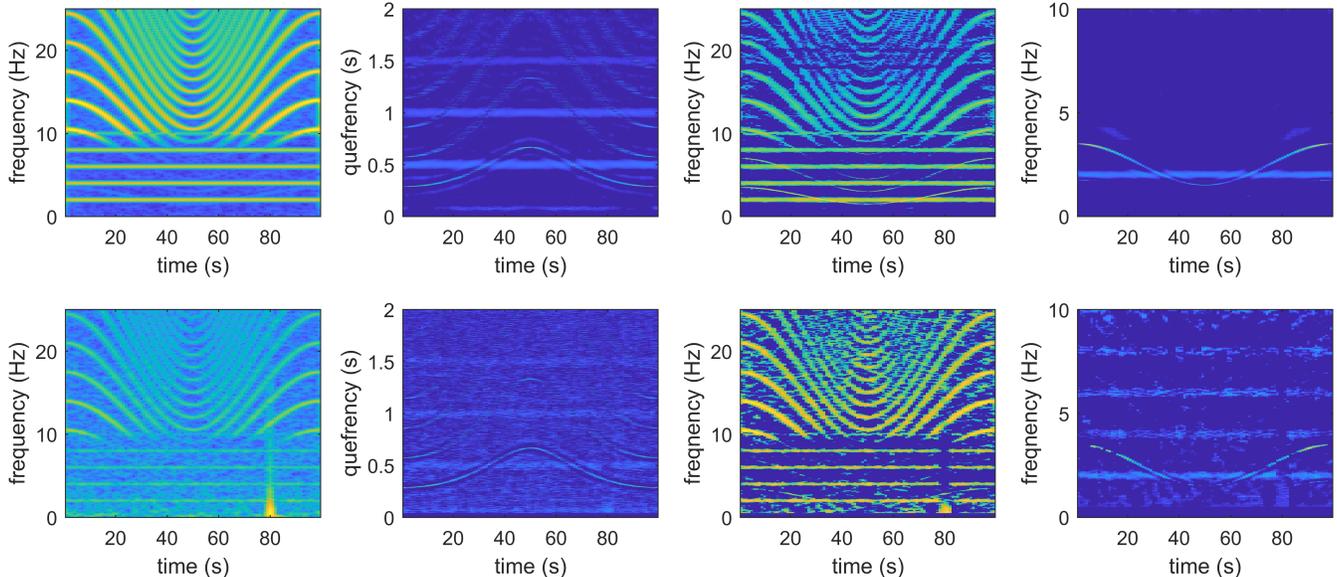
Fig. 1. Simulation result. Upper: $\mathbf{x}$; lower: $\hat{\mathbf{x}}$. From left to right: $\mathbf{Z}^{(0)}$ (power-scale spectrogram), $\mathbf{Z}^{(1)}$ (GC), $\mathbf{Z}^{(2)}$ (GCoS), and $\mathbf{Y}^{(2,1)}$ (CFP representation). The window size is 8 seconds and the hop size is 1 second.

The parameters $[\gamma_0, \gamma_1, \gamma_2] = [0.24, 0.6, 1]$ are used. For the STFT of both samples, the F0 peaks and the low-order harmonics of $\mathbf{x}_2$ are all unseen, and only the first 5 multiples of $\mathbf{x}_1$ are visible due to low-pass filtering. To capture the true F0 of $\mathbf{x}_2$ from the STFT is very challenging. However, such information can be found in the GC: although high-pass filtering noise attenuates the true F0 component in the spectrum, it does not change the period of the signal. Such a phenomenon can also be seen in the case with pink noise. The true F0 trajectory then appears in the GCoS with high resolution; this is because the nonlinear scaling of the GC provides the mechanism to *down-mixing* the frequency when it is transformed back to the frequency domain. Interestingly, we can see that although the true F0 of $\mathbf{x}_2$ is missed, the resulting F0 trajectory of $\mathbf{x}_2$ has resolution even higher than the one of $\mathbf{x}_1$. Since most of the harmonic peaks of $\mathbf{x}_1$ are eliminated, its spectrum cannot support better resolution when transformed to the quefrency domain. Finally, the CFP representation computed through (2) clearly indicates the true F0 trajectories of $\mathbf{x}_1$ and $\mathbf{x}_2$ with less interference.

5. REAL-DATA EXPERIMENTS

We apply the MLC-CFP method for MF0 estimation of polyphonic music data, and investigate the effects of 1) the number of layers, 2) critical parameters γ_i of each layer, and 3) the presence of convolutional noise. We used the Bach10 dataset, containing ten quartets of four different instruments [26] for the evaluation of multi-pitch estimation (MPE). To see the behavior of the algorithm in a straightforward way, we consider three methods to optimize parameters $\{\gamma_i\}_{i=0}^{L}$ for an L-layer

MLC-CFP algorithm combining $\mathbf{Z}^{(L-1)}$ and $\mathbf{Z}^{(L)}$:

1) the *brute-force method* that searches for the optimal parameters exhaustively for each γ_i. The search range of every γ_i is from 0.1 to 0.9 with step size of 0.1.

2) the *greedy method* that searches for the optimal value of γ_l based on the optimal values of $\{\gamma_i\}_{i=0}^{l-1}$. The search range is from 0.01 to 0.99 with step size of 0.01, since its light computation allows us to use finer grids. For the post-processing of the brute-force and the greedy methods, we use the CFP-based algorithm reported in [21], as a simplified version of [9] without too detailed hand-crafted rules.

3) the *stochastic gradient descent (SGD) method* that minimizes the binary cross-entropy between $\mathbf{Y}^{(L)}$ (with 88 log-frequency filterbanks and a sigmoid function as the output layer) and the ground-truth piano roll by performing SGD over every γ_i. We apply non-repeated 10-fold cross-validation on the dataset: for each time one music piece is chosen as the test data and others as training data. The resulting F-scores are obtained by summing all the counts of the true positives, false positives, and false negatives. We initialize γ_0 to 0.24, γ_1 to 0.6, and all other γ_is to 1, and use the SGD optimizer with 0.1 learning rate, a mini-batch size of 256, and train the network up to 40 epochs.

For other parameters, we compute 7939-point DFT using Blackman-Harris window, the hop size is 10 ms, the cutoff frequency f_c is 27.5 Hz (frequency of A0), and the cutoff quefrency q_c is 0.24 ms (period of C8). Since there are many task-dependent parameters, all the source codes will be announce after this paper is accepted for reproducibility.

Table 1 lists the optimal precision (P), recall (R), F-score (F), along with γ_i for every pairs $\left[\mathbf{Z}^{(L-1)}, \mathbf{Z}^{(L)}\right]$ for all opti-

Table 1. Comparison of different layer combinations and testing method on the Bach10 dataset.

Layers	Brute Force									Greedy				SGD (10-fold CV)		
$\{L, L+1\}$	γ_0	γ_1	γ_2	γ_3	γ_4	γ_5	P (%)	R (%)	F (%)	γ_i	P (%)	R (%)	F (%)	P (%)	R (%)	F (%)
0th & 1st	0.3	1	x	x	x	x	77.72	83.51	80.51	0.33	78.05	83.57	80.71	77.67	78.43	78.05
1st & 2nd	0.3	0.5	1	x	x	x	80.55	87.20	83.74	0.48	81.05	85.23	83.09	79.35	88.10	83.50
2nd & 3rd	0.2	0.6	0.9	1	x	x	79.33	89.93	84.30	0.71	80.28	86.66	83.35	**82.61**	85.99	**84.26**
3rd & 4th	0.1	0.9	0.9	0.5	1	x	81.06	90.80	85.66	0.65	80.16	87.08	83.47	81.02	85.83	83.36
4th & 5th	0.1	0.9	0.9	0.7	0.8	1	81.14	90.67	85.64	0.78	80.20	87.27	83.58	81.80	**86.22**	83.95
5th & 6th	0.1	0.9	0.9	0.7	0.8	0.5	**82.35**	**91.10**	**86.50**	0.70	**80.29**	**87.57**	**83.77**	81.97	82.18	82.07

mization approaches. When using the brute-force method, the resulting P, R and F all increase consistently when the number of layers increases. Note that the recall increases dramatically with higher-order layer because the weak F0 peaks are enhanced, similar to the simulation example in Fig. 1. When using the greedy method, we observe that although the performance is sub-optimal, it still increases as L increases. When using gradient descent, there is no consistent improvement in F-scores related to the number of layers, possibly because the hand-crafted rules such as harmonic selection [9] could not be applied to this model. However, it still gives competitive performance for $L = 3$. More specifically, when compared to other MF0 estimation methods, all the three methods in our discussion outperform the state-of-the-art matrix factorization approaches such as constrained NMF and probabilistic latent component analysis (PLCA) (see Table III in [9] for more details). The brute-force method also outperform the CFP result reported in [9], in which the parameters for the hand-crafted rules are also based on brute-force searching.

Fig. 2 shows the resulting F-scores on the Bach10 dataset with high-pass degradation, which is simulated by a 4th-order Butterworth high-pass filters with cutoff frequency from 10 to 1000 Hz. The parameters used here are those found from the brute force method; see Table 1. Fig. 2 clearly shows that with more layers the results are better: even under a severe cutoff frequency at 1 kHz, a 6-layer MLC still gives 72.74% F-score, 25% higher than the one of a 1-layer MLC. Even more interestingly, results in some degradation conditions are better than those in the normal condition. For example, pairs $\mathbf{Z}^{(5)}$ and $\mathbf{Z}^{(6)}$ achieve 87.10% F-score with cutoff frequency at 100 Hz; this is probably because the simulated degradation suppresses the low-frequency noise incidentally.

To summarize, our experiments show that 1) in the MLC-CFP algorithm, a deep DFT do performs better than a shallow DFT, and 2) MLC is highly robust to high-pass noise.

6. DISCUSSION AND CONCLUSION

We have introduced the MLC, explained its physical insight in signal processing, and verified its potential in detecting multiple F0s through simulation and real-world data under normal and noisy conditions. We also emphasize that the MLC is not only useful in the signals contaminated with high-pass convo-
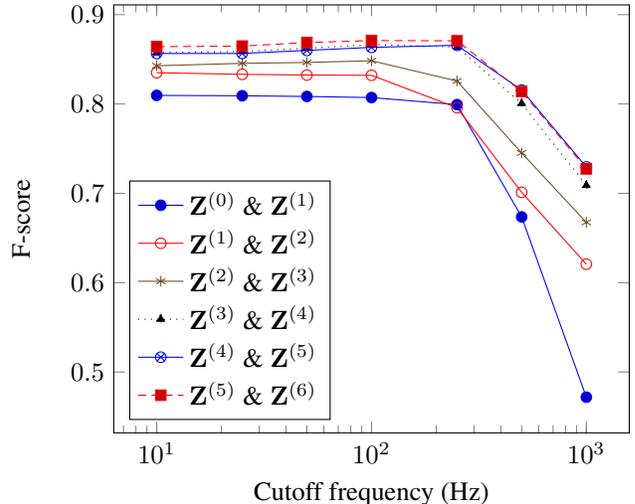
Fig. 2. F-scores of Bach10 dataset under high-pass noise.

lutional noise, but also useful in all signals which inherently have weak F0 magnitude. For example, the F0 of bassoon in the Bach10 dataset is usually weaker than other instruments (violin, clarinet and saxophone), but experiments show that this issue is addressed by the MLC. From the result in the filter test we also suggest that in practice using MLC with a little high-pass filter can get better result.

One risk of MLC is that it may generate fake F0 peaks. For example, two F0s at a Hz and b Hz may produces a cross term at $|a - b|$ Hz due to effect of nonlinear activation. This cross term cannot be suppressed and may be even enhanced because it appears in both the time-domain and frequency-domain representations. To restrict the strength of cross terms, the results indicate a possible strategy: setting a small γ_0 and larger γ_is at the higher layers. Our pilot study also shows that even fake cross terms are produced, they are mostly weak unless all γ_is approach zero.

Developing a more efficient way to optimize the parameters efficiently on a large-scale dataset with complicated sources and harmonic structures will be another focus in the future. From the current finding, we summarize that MLC-CFP works well in general and is relatively easy to optimize for $L \leq 3$, but a great potential exists in a deeper MLC.

7. REFERENCES

[1] G. Pirker, M. Wohlmayr, S. Petrik, and F. Pernkopf, "A pitch tracking corpus with evaluation on multipitch tracking scenario," in *INTERSPEECH*, 2011.

[2] M. Wu, D.L. 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.

[3] 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.

[4] Z. Zhang, Z. Pi, and B. Liu, "TROIKA: A general framework for heart rate monitoring using wrist-type photoplethysmographic signals during intensive physical exercise," *IEEE Trans. Bio-Med. Eng.*, vol. 62, no. 2, pp. 522–531, 2015.

[5] Z. Li, J. Wang, and J. Han, "ePeriodicity: Mining event periodicity from incomplete observations," *IEEE Trans. Knowledge and Data Engineering*, vol. 27, no. 5, pp. 1219–1232, 2015.

[6] L. Rabiner, "On the use of autocorrelation analysis for pitch detection," *IEEE Trans Acoust., Speech, Signal Process.*, vol. 25, no. 1, pp. 24–33, 1977.

[7] T. Kobayashi and S. Imai, "Spectral analysis using generalized cepstrum," *IEEE Trans. Acoust., Speech, Signal Process*, vol. 32, no. 5, pp. 1087–1089, 1984.

[8] M. Græsbøll Christensen, P. Stoica, A. Jakobsson, and S. Holdt Jensen, "Multi-pitch estimation," *Signal Processing*, vol. 88, no. 4, pp. 972–983, 2008.

[9] 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.

[10] L. Su, T.-Y. Chuang, and Y.-H. Yang, "Exploiting frequency, periodicity and harmonicity using advanced time-frequency concentration techniques for multipitch estimation of choir and symphony," in *Proc. ISMIR*, 2016, pp. 393–399.

[11] C.-Y. Lin, L. Su, and H.-T. Wu, "Wave-shape function analysis–when cepstrum meets time-frequency analysis," *Journal of Fourier Analysis and Applications*, pp. 1–55, 2017.

[12] P. Verma and R. W. Schafer, "Frequency estimation from waveforms using multi-layered neural networks," in *INTERSPEECH*, 2016, pp. 2165–2169.

[13] G. Peeters, "Music pitch representation by periodicity measures based on combined temporal and spectral representations," in *Proc. ICASSP*, 2006.

[14] H. Hermansky, N. Morgan, and H.-G. Hirsch, "Recognition of speech in additive and convolutional noise based on rasta spectral processing," in *Proc. ICASSP*. IEEE, 1993, vol. 2, pp. 83–86.

[15] M. Mauch and S. Ewert, "The audio degradation toolbox and its application to robustness evaluation.," in *ISMIR*, 2013, pp. 83–88, `https://code.soundsoftware.ac.uk/projects/audio-degradation-toolbox` (v0.2).

[16] T. D. Rossing, F. R. Moore, and P. A. Wheeler, *The Science of Sound*, Addison Wesley San Francisco, 2002.

[17] B.P. Bogert, M.J.R. Healy, and J.W. Tukey, "The quefrency analysis of time series for echoes: Cepstrum, pseudo-autocovariance, cross-cepstrum and shape cracking," in *Time Series Analysis*, 1963, vol. 15, pp. 209–243.

[18] A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing*, Prentice Hall, 3rd edition, 2009.

[19] T. Tolonen and M. Karjalainen, "A computationally efficient multipitch analysis model," *IEEE Speech Audio Process.*, vol. 8, no. 6, pp. 708–716, 2000.

[20] A. Klapuri, "Multipitch analysis of polyphonic music and speech signals using an auditory model," *IEEE Trans. Audio, Speech, Language Process.*, vol. 16, no. 2, pp. 255–266, 2008.

[21] L. Su, "Between homomorphic signal processing and deep neural networks: Constructing deep algorithms for polyphonic music transcription," in *APSIPA ASC*, 2017.

[22] S. Sigtia, E. Benetos, and S. Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 5, pp. 927–939, 2016.

[23] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for f_0 estimation in polyphonic music," in *Proc. ISMIR*, Oct. 2017.

[24] J. Thickstun, Z. Harchaoui, and S. Kakade, "Learning features of music from scratch," *arXiv preprint arXiv:1611.09827*, 2016.

[25] A. Elowsson, "Polyphonic pitch tracking with deep layered learning," *arXiv preprint arXiv:1804.02918*, 2018.

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