INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

A fast and accurate fundamental frequency estimator using recursive moving average filters

Ryunosuke Daido[1], *Yuji Hisaminato*[1]

[1]Yamaha Corporation

ryunosuke.daido@music.yamaha.com, yuji.hisaminato@music.yamaha.com

Abstract

We propose a fundamental frequency (F0) estimation method which is fast, accurate and suitable for real-time use. While the proposed method is based on the same framework as DIO [1, 2], it has two clear differences: it uses RMA (Recursive Moving Average) filters for attenuating high order harmonics, and the period detector is designed to work well even for signals which contain some higher harmonics. Effect of trace-back duration of post-processing was also examined. Evaluation experiments using natural speech databases showed that the accuracy of the proposed method was better than DIO, SWIPE′ [3] and YIN [4] and computation speed was the fastest compared to those existing methods.

Index Terms: fundamental frequency (F0), speech analysis, speech processing.

1. Introduction

Fundamental frequency (F0) of speech is a quantity which is strongly related to perceptual pitch and widely used as an important feature in fields of speech/singing voice synthesis [5, 6, 7], singing voice assessment [8, 9], etc. Furthermore, in today's high quality analysis-and-synthesis systems [10, 11, 12, 13] including the widely used STRAIGHT system, other features of speech, for example spectral envelope or aperiodicity index, are accurately estimated using F0 adaptive windows [14, 15, 16]. Reliable F0 estimation is even more important in such systems because it affects other analysis components' performances.

While many F0 estimation methods have been proposed [17, 18, 4, 19, 3], most focus only on accuracy and few aim to improve computation speed. Real-time voice transformation systems have been proposed in both academic [20, 21] and commercial fields [22]. For making such systems capable of doing higher quality analysis and synthesis, it is important to develop F0 estimation methods which are accurate, fast and also suitable for real-time use.

Morise *et al.* proposed a fast and accurate F0 estimation method [1, 2]. They also added offline post-processing by some heuristic rules and this system, named DIO, is used as a component of WORLD [13]: a high quality analysis-and-synthesis system. We write DIO in this paper as a generic name for the method of [1, 2] and the component of WORLD. DIO basically works by finding intervals of zero crossings and peaks of a waveform, and does not need to compute FFT (Fast Fourier Transform) or correlation frame by frame, so computation speed is fast in this part. However, DIO calculates convolution with FIR filters to attenuate high order harmonics before finding intervals, which can be a computation speed bottleneck. It is well-known that FIR filters can be computed efficiently using the FFT [23] but this technique only works efficiently when the block size is large, so it is not suitable for real-time use. Also, DIO tries to obtain an almost sinusoidal waveform of the fundamental component by using multiple low-pass filters, but it requires a large number of filters causing another increase in computational cost, and introducing a shortcoming that it does not work reliably when the amplitude of the fundamental component is small.

In this paper, we propose an F0 estimation method which is fast, accurate and suitable for real-time use by improving the shortcomings of DIO. This paper is organized as follows. Our proposed method is described in Section 2. In Section 3, evaluation results using natural speech databases are shown. Here we describe the results of comparison to some state-of-the-art F0 estimation methods in aspects of both accuracy and computation speed. Finally we conclude in Section 4.

2. Proposed method

The basic framework of our proposed method is the same as DIO and improvements are proposed for the components. Figure 1 shows the framework. An input signal is filtered by multiple parallel low-pass filters for harmonic attenuation and a period detector is applied for each band. Finally, the most suitable band is selected for estimating fundamental period and the reciprocal of the period is output. Each component of our proposed method is described in more detail below following the signal flow in Figure 1.

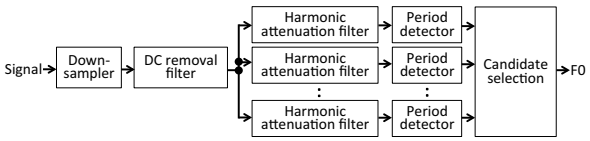

Figure 1: Overview of the proposed method.

2.1. Down-sampler

A down-sampler is first applied to reduce computational cost. We employed a CIC filter [24] for this purpose. It is an applied form of RMA (Recursive Moving Average) filter and works efficiently for decimation with anti-aliasing.

2.2. DC removal filter

Since the proposed method measures zero-crossing intervals, it is important to remove non-harmonic frequency components near DC. A DC removal filter can also be realized by an applied form of RMA filter. We employed a method [25] in which a pass through signal of two cascaded RMA filters is subtracted

http://dx.doi.org/10.21437/Interspeech.2016-394

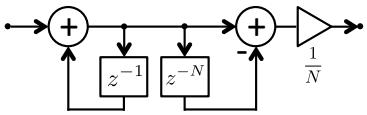

Figure 2: Block diagram of an RMA filter.

from a group delay compensated input signal.

2.3. Harmonic attenuation filters

One distinct feature of the proposed method is the design of the harmonic attenuation filters. Since we use multiple filters in parallel, they can represent a significant part of the computational cost. For making them efficient, we propose using RMA filters. A block diagram of an RMA filter is shown in Figure 2.

The transfer function of an RMA filter is equivalent to an N-point simple moving average filter:

$$H(z) = \frac{1}{N} \cdot \frac{1 - z^{-N}}{1 - z^{-1}} = \frac{1}{N} \sum_{n=0}^{N-1} z^{-n}. \quad (1)$$

Choosing a power of two for N, the multiplication by $1/N$ can be computed using right bit shift. Then this filter can be computed without any multiplications which are generally slower than additions or bit shifts. Therefore it is expected that this filter will work efficiently in many environments.

Since the attenuation by a single RMA filter is not enough, we use a set of 6 cascaded RMA filters for each band as a harmonic attenuation filter. As shown in the magnitude responses in Figure 3, it has the first zero at the frequency $k \cdot F_s/N$, where k is an integer greater than 1 and F_s is the sample rate, and works as a low-pass filter. This filter has a gentle cutoff characteristic similar to the Nuttall window which is used in DIO and the stop-band attenuation reaches around 80 dB. While high order harmonics may still remain because of the gentle cutoff characteristic, they are attenuated more as frequency increases so the possibility for fundamental period detection is increased.

This filter clearly has linear phase response so group delay compensation can be done straightforwardly in the next period detection part.

2.4. Period detectors

Another distinct feature of the proposed method is the design of the period detectors. In DIO, a period detector is designed to find an almost sinusoidal waveform of the fundamental component. We propose another period detector which can detect and measure the fundamental period even for signals which contain some high order harmonics. The proposed method has an advantage in that it reduces the required number of frequency bands of the harmonic attenuation filters. Also, the amplitude of the fundamental component in speech is sometimes very low and hard to detect due to background noise. NDF [19] combines correlation based period estimation with instantaneous frequency based harmonic detection for handling this "almost missing fundamental" case. The high accuracy of NDF encouraged our idea for the improved period estimation.

The proposed period detector runs in real-time sample by sample and detects zero crossings and peaks according to the rules below:

- The period detector has 4 modes and each mode detects positive zero crossings, positive peaks, negative zero crossings and negative peaks

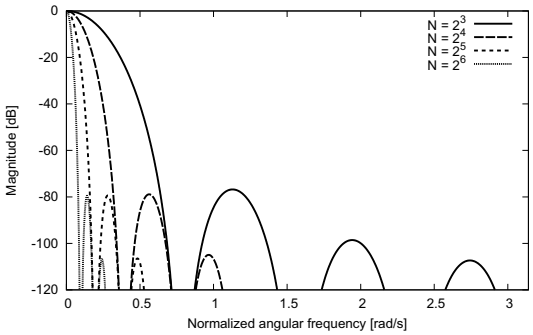

Figure 3: Magnitude response of 6 cascaded RMA filters.

- The mode is changed in the order above when the period detector detects a zero crossing or a peak (and goes back to the first mode after the final mode)
- The positive peak detection mode detects only peaks whose peak amplitude is positive, and it ignores peaks whose amplitude is less than α $(0 < \alpha < 1)$ times the absolute amplitude of the last negative peak
- The negative peak detection mode detects only peaks whose peak amplitude is negative, and it ignores peaks whose absolute value of the amplitude is less than α $(0 < \alpha < 1)$ times the last positive peak amplitude
- Any zero crossings and peaks in τ_{min} from the last detection are ignored
- When τ_{max} elapsed with no detection of any zero crossings or peaks, it is regarded as "no period"

Figure 4 shows an example of period detection by the proposed method. Filled circles indicate zero crossings and peaks detected. Each of the positive/negative zero crossings and positive/negative peaks are detected by an interval of the fundamental period. Note that DIO's period detector can not detect fundamental period from this signal because it also detects other zero crossings and peaks than those indicated by filled circles.

The period detectors output period T, period detection time t and "Fundamental-period likeliness" λ. Those are computed as:

$$T = \frac{1}{4}(l_{PZ} + l_{PP} + l_{NZ} + l_{NP}) \quad (2)$$

$$t = \frac{1}{4}(t_{PZ} + t_{PP} + t_{NZ} + t_{NP}) - \frac{1}{2}T - D \quad (3)$$

$$\lambda = 1 - \min\{\frac{1}{4T}(|l_{PZ} - T| + |l_{PP} - T| + |l_{NZ} - T| + |l_{NP} - T|), 1\} \quad (4)$$

where t_{PZ} and t_{NZ} are the detection times of the last positive and negative zero crossings, t_{PP} and t_{NP} are the detection times of the last detected positive and negative peaks, l_{PZ}, l_{NZ}, l_{PP} and l_{NP} are the time intervals between the last and the one before last for each of these, D is the total group delay of the down-sampler, the DC removal filter and the harmonic attenuation filter of each band. Note that zero crossing times and peak times are computed in fractional samples using linear or quadric interpolation to improve accuracy. "Fundamental-period likeliness" of eq.4 is based on the similar idea of "Fundamental-ness" of [2] and measures reliability of the detected period as fundamental period based on variance of the four intervals. In eq.4, we avoided multiplications for computational efficiency.

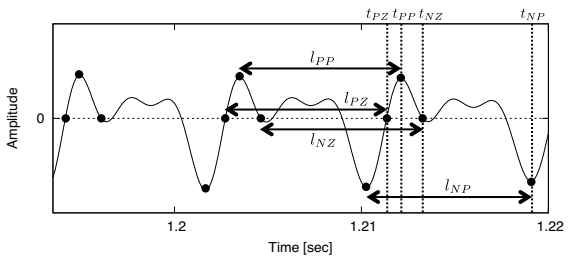

Figure 4: Example of period detection by the proposed method.

Applying this period detector for each harmonic attenuation filter output, a set of period T, period detection time t and Fundamental-period likeliness λ is output four times per period so the output interval is not constant. In the proposed method, period and Fundamental-period likeliness are computed by linear interpolation in a band at an arbitrary time.

2.5. Candidate selection

The best suitable fundamental period should be selected from the outputs of the period detectors. The simplest way is to select the band which gave the best Fundamental-period likeliness at an arbitrary time or regard as unvoiced if all bands detected "no period". We also tried to improve accuracy by applying the dynamic programming-based post-processing of RAPT [18]. It finds candidate series which minimize a sum of local costs and transition costs based on F0 continuity. The post-processing introduces some latency for real-time use, however its trace-back duration, noted as τ_b in this paper, can be set arbitrarily to balance between required accuracy and latency for each situation. In RAPT, local maximum values of a normalized cross correlation function are used for defining the local cost. In our proposed method, Fundamental-period likeliness λ is used instead. The energy of a frame used for defining transition cost in RAPT was replaced by a peak amplitude which is detected by the period detectors. Also, voiced candidates whose Fundamental-period likeliness are less than a threshold Λ_{th}, peak amplitudes are less than a threshold A_{th} or periods out of estimation range were discarded in our post-processing.

3. Evaluation

The proposed method was evaluated using natural speech databases for both accuracy and computation speed. We implemented the proposed method in C++.

3.1. Compared methods

For comparative evaluations, we used recent state-of-the-art methods: DIO, NDF, SWIPE′ and YIN. For DIO, we used the implementation in C++ by its authors as a component of WORLD [26]. Harmonic attenuation filters were convolved using the FFT in this implementation. For SWIPE′, we used the implementation in C by Gorman [27]. For NDF and YIN, we used the MATLAB implementations by their authors [28, 29].

3.2. Databases

We used publicly available natural speech databases which contain speech signals and simultaneously recorded EGG signals. From the CMU Arctic database [30], *bdl*, *jmk* and *slt* were used as DB1-DB3 and from Bagshaw's database [31, 32], *rl* and *sb* were used as DB4-DB5. The total duration is 176 minutes. The ground truth of F0 and voicing decision were obtained from differentiated EGG signals using NDF. We used NDF here because it is reported that it achieved the best accuracy in multiple recent papers [19, 1, 33] and also is supposed to have a good temporal resolution [34].

3.3. Conditions

For all methods, F0 estimation interval (i.e. hop size) was set to 1 ms and the floor and ceiling of F0 estimation range were set to $F_{floor} = 40$, $F_{ceil} = 800$ Hz.

Parameters for the proposed methods were set as follows. For the down-sampler, we chose an integer r for decimation ratio $R = 2^r$ which makes the down-sampled sample rate the closest to 10 kHz. Therefore DB1-3 were downsampled from 32 kHz to 8 kHz and DB4-5 were from 20 kHz to 10 kHz. For RMA filters in the DC removal filter, $N = 2^6$ in eq.1 was set. Four bands of harmonic attenuation filters were used and $N = 2^3, 2^4, 2^5, 2^6$ was set for RMA filters in each band. For the period detectors, the maximum interval of zero crossings and peaks was set to $\tau_{max} = 1.2/(4 \cdot F_{floor})$ s, the minimum interval was set to $\tau_{min} = N/(4 \cdot 0.8 \cdot F_s')$ s for each band, where F_s' is the down-sampled sample rate. The peak amplitude ratio threshold was set to $\alpha = 1/8$. For the post-processing, the threshold of Fundamental-period likeliness was set to $\Lambda_{th} = 0.6$ and the amplitude threshold was set to $A_{th} = 0.00005$.

Parameters of each method were set to the values that are proposed in the original papers. For SWIPE′, 96 ch./oct. search step and 0.1 ERB frequency sampling step were set. For DIO, 2 ch./oct. filter bands were set [2]. DIO from WORLD also has an option of down-sampling rate and it was set to the same value as the proposed method because an appropriate value was not proposed in the original papers. For NDF, we did not give any explicit parameters and just called *MulticueF0v14()* only with an input signal and a sample rate.

3.4. Accuracy evaluation

We employed GER (Gross Error Rate) [4] and RMSE (Root Mean Square Error) [35] as accuracy evaluation measures. GER is a rate of times where more than 20% (in Hz) errors occurred. 20% in Hz means errors of over 300 cents so we think it indicates a rate of rather big errors. RMSE is computed without such a threshold so it measures the amount of estimation error more directly. Each estimation result file was time-aligned with the ground truth to adjust time lag between speech and EGG signals, and also to make differences of window positioning of each method heve no effect. The measures above were computed using times where all methods and the ground truth agreed as voiced.

Table 1 shows GER and Table 2 shows RMSE for each DB. Rows are sorted by the result of all DBs. About the trace-back duration τ_b of the post-processing of the proposed method, we did preliminary experiments for the range $0 \leq \tau_b \leq 100$ ms and confirmed that both GER and RMSE decrease along with increase of τ_b and that they each almost converge at $\tau_b = 30$ ms. Therefore those tables shows the results in which only $\tau_b \in \{0, 30\}$ ms and no post-processing conditions were examined.

For GER, the result using all DBs shows that NDF achieved the best result and the proposed method ($\tau_b = 30$) followed. For RMSE, it is also shown that NDF achieved the best result and the proposed methods followed. It is interesting that there are large differences between the proposed methods and SWIPE′ in RMSE while the difference was small in GER. For more de-

Table 1: GER [%] for each methods. All means the evaluation result using all the databases.

Method	All	DB1	DB2	DB3	DB4	DB5
NDF	0.86	1.40	0.62	0.59	0.58	0.34
Proposed ($\tau_b = 30$ ms)	1.49	2.41	0.95	1.07	1.43	1.28
SWIPE′	1.65	2.56	1.09	1.27	1.42	1.50
Proposed ($\tau_b = 0$ ms)	1.85	2.62	1.21	1.62	2.05	3.09
Proposed (without P.P.)	1.87	2.68	1.51	1.39	2.90	2.13
DIO (without P.P.)	2.94	4.58	2.92	1.41	4.25	2.43
YIN	3.75	5.68	4.19	1.62	4.92	1.65
DIO (with P.P.)	3.92	8.80	1.83	0.98	3.38	8.08

Table 2: RMSE [Hz] for each methods. All means the evaluation result using all the databases.

Method	All	DB1	DB2	DB3	DB4	DB5
NDF	8.7	10.9	7.3	7.3	7.3	8.1
Proposed ($\tau_b = 30$ ms)	9.1	10.6	5.8	9.7	8.0	15.5
Proposed ($\tau_b = 0$ ms)	11.1	11.3	7.0	12.6	9.6	29.2
Proposed (without P.P.)	11.9	14.1	7.7	11.8	17.1	24.4
DIO (with P.P.)	15.2	21.0	8.3	9.3	12.1	55.2
SWIPE′	21.5	33.4	6.8	14.1	15.0	27.2
YIN	28.6	42.1	12.8	21.3	25.4	33.7
DIO (without P.P.)	36.7	58.3	12.0	20.9	29.3	35.3

tailed observation, density distributions of estimation error were estimated using kernel density estimation. Figure 5 shows the result. It is shown that the error density of NDF and the proposed method is concentrated well around 0 Hz compared to other methods. We think the RMSE difference reflects such differences of distributions of fine errors not counted as Gross Errors.

Comparing the proposed method to DIO without post-processing conditions, the proposed method achieved better results for both GER and RMSE for most of the DBs. In DIO, its post-processing made the results even worse for some DBs. We found DIO sometimes mistakes low frequency noise components for fundamental components of speech. Airflow just after plosives, air conditioner noise and power line hum were observed as causes. Then its post-processing continued the error to make the result even worse. We did not find such errors in the results of the proposed method. It can be explained that the period detectors of the proposed method can detect fundamental period even for signals which contain some high order harmonics and can choose a higher band where the amplitudes of speech harmonics are dominant. If noise is present, the Fundamental-period likeliness is evaluated higher in this higher band than lower bands where the low frequency noise has a severe effect.

Throughout the results, it is shown that the accuracy of the proposed method is worse than NDF but better than SWIPE′, DIO and YIN when trace-back of $\tau_b \geq 30$ is applied, and better than DIO and YIN even without post-processing. Although it could be arguable that the proposed method can decide difficult to estimate portions as unvoiced, which were not computed in this evaluation scheme, ratios of times where only the proposed method decided as unvoiced were only 0.14 % in the maximum ($\tau_b = 30$ ms) case and 0.04 % in the minimum (without post-processing) case. Therefore the GER ranking will not change even if we count all of them as Gross Errors.

3.5. Speed evaluation

For an evaluation of computation speed, we computed processing time per 1 s speech input by dividing measured processing time for processing all the files in DB1-5 by total input duration. Processing time for file IO was excluded from measurement. Measurements were conducted 10 times for each method and a median was employed for each file. In this evaluation, only the proposed method, DIO and SWIPE′ which are implemented in C or C++ were used. The measurements were run on a MacBook Pro (Retina, 15-inch, Early 2013) with a 2.4 GHz Intel Core i7 and 16 GB 1600 MHz DDR3.

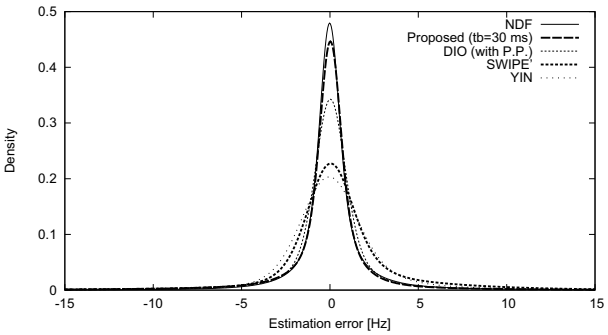

Figure 5: Distribution of estimation error.

Table 3 shows the results. The proposed method was the fastest and approximately 30 times faster than SWIPE′. While the post-processing increased the processing time a little, dependency on trace-back duration τ_b is not significant. Therefore, in a practical sense, we can set an arbitrary value of τ_b for real-time use considering a balance between accuracy and latency for each situation, and can set $\tau_b \geq 30$ ms to do accurate and fast estimation for offline use where the trace-back latency causes no problem. Also in a practical sense, speed measures of NDF and YIN would be useful information although they can not directly be compared to the result above because they are implemented in MATLAB. As the result of the same (however only 3 iterations) procedures, YIN took 75 ms and NDF took 1857 ms per 1 s input speech.

Table 3: Processing durations [ms] per 1 second input for each method.

Proposed (without P.P.)	1.7
Proposed ($\tau_b = 0$ ms)	2.0
Proposed ($\tau_b = 30$ ms)	2.1
DIO (without P.P.)	5.7
DIO (with P.P.)	5.7
SWIPE′	64.8

4. Conclusion

In this paper, we proposed an F0 estimation method which is fast, accurate and also suitable for real-time use. The accuracy evaluation results showed it is more accurate than SWIPE′, DIO and YIN when ≥ 30 ms trace-back is applied, and more accurate than DIO and YIN even without post-processing. Also the speed evaluation result showed that it is the fastest in recent state-of-the-art methods. As future work, we plan to do evaluations and improvements for targeting singing voice for which real-time voice processing applications are more widely used.

5. References

[1] M. Morise, H. Kawahara, and H. Katayose, "Fast and reliable f0 estimation method based on the period extraction of vocal fold vi-

bration of singing voice and speech," in *Audio Engineering Society Conference: 35th International Conference: Audio for Games*. Audio Engineering Society, 2009.

[2] M. Morise, H. Kawahara, and T. Nishiura, "Rapid f0 estimation for high-snr speech based on fundamental component extraction," *The IEICE Transactions on Information and Systems (Japanese Edition)*, vol. 93, no. 2, pp. 109–117, 2010.

[3] A. Camacho and J. G. Harris, "A sawtooth waveform inspired pitch estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 124, no. 3, pp. 1638–1652, 2008.

[4] A. De Cheveigné and H. Kawahara, "Yin, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[5] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis," in *Proc. Eurospeech*, 1999, pp. 2347–2350.

[6] K. Saino, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda, "An hmm-based singing voice synthesis system." in *INTERSPEECH*, 2006.

[7] H. Kenmochi and H. Ohshita, "Vocaloid-commercial singing synthesizer based on sample concatenation." in *INTERSPEECH*, vol. 2007. Citeseer, 2007, pp. 4009–4010.

[8] T. Nakano, M. Goto, and Y. Hiraga, "An automatic singing skill evaluation method for unknown melodies using pitch interval accuracy and vibrato features," *Rn*, vol. 12, p. 1, 2006.

[9] R. Daido, M. Ito, S. Makino, and A. Ito, "Automatic evaluation of singing enthusiasm for karaoke," *Computer Speech & Language*, vol. 28, no. 2, pp. 501–517, 2014.

[10] H. Kawahara, I. Masuda-Katsuse, and A. De Cheveigne, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds," *Speech communication*, vol. 27, no. 3, pp. 187–207, 1999.

[11] J. Bonada, "Wide-band harmonic sinusoidal modeling," in *11th International Conference on Digital Audio Effects DAFx*, vol. 8, 2008.

[12] H. Kawahara, M. Morise, T. Takahashi, R. Nisimura, T. Irino, and H. Banno, "Tandem-straight: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, f0, and aperiodicity estimation," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 3933–3936.

[13] M. Morise, "An attempt to develop a singing synthesizer by collaborative creation," in *Proc. SMAC*, 2013, pp. 287–292.

[14] T. Nakano and M. Goto, "A spectral envelope estimation method based on f0-adaptive multi-frame integration analysis," in *SAPA-SCALE Conference*, 2012, pp. 11–16.

[15] H. Kawahara and M. Morise, "Simplified aperiodicity representation for high-quality speech manipulation systems," in *2012 IEEE 11th International Conference on Signal Processing*, 2012, pp. 579–584.

[16] M. Morise, "Cheaptrick, a spectral envelope estimator for high-quality speech synthesis," *Speech Communication*, vol. 67, pp. 1–7, 2015.

[17] L. Rabiner, M. J. Cheng, A. E. Rosenberg, and C. A. McGonegal, "A comparative performance study of several pitch detection algorithms," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 24, no. 5, pp. 399–418, 1976.

[18] D. Talkin, "A robust algorithm for pitch tracking (rapt)," *Speech coding and synthesis*, vol. 495, p. 518, 1995.

[19] H. Kawahara, A. de Cheveigné, H. Banno, T. Takahashi, and T. Irino, "Nearly defect-free f0 trajectory extraction for expressive speech modifications based on straight." in *Interspeech*, 2005, pp. 537–540.

[20] A. Röbel and X. Rodet, "Efficient spectral envelope estimation and its application to pitch shifting and envelope preservation," in *International Conference on Digital Audio Effects*, 2005, pp. 30–35.

[21] E. Azarov, M. Vashkevich, D. Likhachov, and A. Petrovsky, "Real-time pitch modification system for speech and singing voice," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[22] Ircam trax v3. [Online]. Available: http://www.fluxhome.com/products/plug_ins/ircam_trax-v3

[23] A. V. Oppenheim, R. W. Schafer, J. R. Buck *et al.*, *Discrete-time signal processing*. Prentice hall Englewood Cliffs, NJ, 1989, vol. 2.

[24] E. Hogenauer, "An economical class of digital filters for decimation and interpolation," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 29, no. 2, pp. 155–162, 1981.

[25] R. Yates and R. Lyons, "Dc blocker algorithms," *IEEE Signal Processing Magazine*, vol. 25, no. 2, p. 132, 2008.

[26] World. [Online]. Available: http://ml.cs.yamanashi.ac.jp/world/english/

[27] Swipe' pitch estimator, v. 1.5. [Online]. Available: https://github.com/kylebgorman/swipe

[28] Straight, a speech analysis, modification and synthesis system. [Online]. Available: http://www.wakayama-u.ac.jp/ kawahara/STRAIGHTadv/index_e.html

[29] Alain de cheveign. [Online]. Available: http://audition.ens.fr/adc/

[30] Cmu arctic speech synthesis databases. [Online]. Available: http://festvox.org/cmu arctic/

[31] P. C. Bagshaw, S. M. Hiller, and M. A. Jack, "Enhanced pitch tracking and the processing of f0 contours for computer aided intonation teaching." *Eurospeech*, pp. 1003–1006, 1993.

[32] Evaluating pitch determination algorithms. [Online]. Available: http://www.cstr.ed.ac.uk/research/projects/fda/

[33] A. Tsanas, M. Zañartu, M. A. Little, C. Fox, L. O. Ramig, and G. D. Clifford, "Robust fundamental frequency estimation in sustained vowels: Detailed algorithmic comparisons and information fusion with adaptive kalman filtering," *The Journal of the Acoustical Society of America*, vol. 135, no. 5, pp. 2885–2901, 2014.

[34] H. Kawahara, K.-I. Sakakibara, H. Banno, M. Morise, T. Toda, and T. Irino, "Aliasing-free implementation of discrete-time glottal source models and their applications to speech synthesis and f0 extractor evaluation," in *Proceedings of APSIPA Annual Summit and Conference*, vol. 16, no. 19, 2015.

[35] M. G. Christensen and A. Jakobsson, "Multi-pitch estimation," *Synthesis Lectures on Speech & Audio Processing*, vol. 5, no. 1, pp. 1–160, 2009.