Performance Analysis of Several Pitch Detection Algorithms on Simulated and Real Noisy Speech Data

Denis Jouvet and Yves Laprie
Speech Group, LORIA
Inria, Villers-lès-Nancy, F-54600, France
Université de Lorraine, LORIA, UMR 7503, Villers-lès-Nancy, F-54600, France
CNRS, LORIA, UMR 7503, Villers-lès-Nancy, F-54600, France
Email: {denis.jouvet,yves.laprie}@loria.fr

***Abstract*—This paper analyses the performance of a large bunch of pitch detection algorithms on clean and noisy speech data. Two sets of noisy speech data are considered. One corresponds to simulated noisy data, and is obtained by adding several types of noise signals at various levels on the clean speech data of the Pitch-Tracking Database from Graz University of Technology (PTDB-TUG). The second one, SPEECON, was recorded in several different acoustic environments. The paper discusses the performance of pitch detection algorithms on the simulated noisy data, and on the real noisy data of the SPEECON corpus. Also, an analysis of the performance of the best pitch detection algorithm with respect to estimated signal-to-noise ratio (SNR) shows that very similar performance is observed on the real noisy data recorded in public places, and on the clean data with addition of babble noise.**

***Index Terms*—Pitch, fundamental frequency, clean speech data, noisy speech data.**

I. Introduction

Numerous pitch detection algorithms have been developed in the past for computing the fundamental frequency (F0) of speech signals. Several of them operate in the time domain, as for example those based on the auto-correlation function (ACF) [1], on the robust algorithm for pitch tracking (RAPT) [2], YIN [3] and the time domain excitation extraction based on a minimum perturbation operator (TEMPO) [4], [5]. Some other operate in the frequency domain as the sawtooth waveform inspired pitch estimator (SWIPE) [6]. Other approaches combine processing in the time and in the frequency domains; this includes the pitch detection of the Aurora algorithm [7] initially developed for distributed speech recognition, and the nearly defect-free F0 (NDF) estimation algorithm [8]. New algorithms have also been release, as for example the robust epoch and pitch estimator (REAPER). Moreover, a pitch tracker has recently been developed for automatic speech recognition of tonal languages with the Kaldi toolkit [9].

Reference corpora have been developed for evaluating pitch detection algorithms. Several corpora correspond to clean speech data recorded from male and female speakers, as for example the PTDB-TUG corpus [10] (20 speakers). Another corpus for evaluating robust pitch detection is based on speech recorded in various environments, with close-talk and distant microphones: SPEECON [11] (60 speakers).

Many pitch detection algorithms have been developed for processing clean speech signals, thus, the corresponding papers provide performance evaluation on clean speech data (e.g., [12]), possibly for a limited set of sounds (e.g., [13]). Since a decade, there have been some performance evaluations conducted either on simulated noisy data (e.g., [14], where noise is added to clean speech reference signals) or on real noisy data using the SPEECON corpus (e.g., [15]–[17]). However, only a small set of pitch detection algorithms were considered in each case.

Last year, [18] has presented a bibliometric survey of the most frequently used pitch detection algorithms; the top list corresponds to Praat [1], RAPT [2], STRAIGHT [4], [5], YIN [3], and SWIPE [6]. In this paper these pitch detection algorithms are considered as well as some others for which implementations were available, and their performance are compared on clean and on noisy speech data.

The outline of the paper is the following. Section II introduces the various pitch detection algorithms considered for evaluation. Section III presents the reference speech and noise corpora, and recalls the evaluation metrics. Section IV presents and discusses the performance evaluation in various conditions: clean data, simulated noisy data, and real noisy data. Finally, performance on real and simulated noisy data are compared for one of the best algorithms.

II. Pitch detection algorithms

Pitch detection, or more precisely F0 detection from a technical point of view, has given rise to a wide variety of algorithms which differ about: **(i)** the pre-processing intended to reduce noise or vocal tract influences, **(ii)** the principle of the pitch detection which can be realized in the time domain, in the frequency domain, or by combining these two facets, **(iii)** the voicing decision either by applying a threshold or a more refined algorithm, **(iv)** the application of a post-processing algorithm intended to smooth the resulting pitch contour, to

ISBN 978-0-9928626-7-1 © EURASIP 2017

remove gross errors (pitch halving or doubling) or to realize pitch marking and correction simultaneously.

The principle of the pitch detection is considered as the main feature even if the concrete implementation has a large influence on the results. The algorithms used in this work (see Table I) operate either *in the temporal domain* as ACF [1] and CCF (Praat), AMDF [19] (snack library) , Kaldi [9], REAPER, RAPT [2] (SPTK and snack library), SRPD [20], [21] (ESTL), and YIN [3]; or *in the frequency domain* by exploiting the harmonic structure of the spectrum as in Martin [22] (JSnoori), SWIPE [6], [23] (SPTK and JSnoori), and SHS [24] (Praat); or *in both domains* as Aurora [7] (ETSI) which spots F0 candidates in the frequency domain and then refines the detection in the temporal domain, or NDF [8] (STRAIGHT) which exploits sub-band autocorrelation and computation of the instantaneous frequency.

In the temporal domain the pitch determination consists of maximizing the correspondence of a signal window with a shifted version of this window, and the challenge is to favor the emergence of the best candidates, for instance by taking account of signal amplitude variations, or normalizing the correlation so as to avoid pitch halving or doubling. In the frequency domain the determination relies on the emergence of the F0 harmonics in the spectrum and the challenge is to minimize the risk of pitch halving, for instance by changing the height of teeth of a spectra comb or exploiting the amplitude difference between harmonics and inter-harmonics valleys. The voicing decision often corresponds to applying a threshold on the numerical criterion used to detect F0.

Some algorithms, especially RAPT, REAPER and Martin (JSnoori) incorporate a DTW correction and smoothing algorithm so as to obtain a F0 curve which minimizes jumps to prevent F0 halving or doubling, and to realize a better voicing decision. This last step which was not necessarily designed with a view of processing noisy speech sometimes turns out to be counterproductive.

III. Experimental setup

Clean speech and noisy speech corpora, with associated reference pitch detection values, are considered, as well as the addition of noise signals on clean speech data.

A. Speech corpora

The Pitch-Tracking Database from Graz University of Technology (PTDB-TUG) contains speech signals from 20 English native speakers (10 female and 10 male, from 22 to 48 years old) reading out sentences from the TIMIT corpus [26]. Overall, 4720 sentences were recorded. The database is provided with reference pitch values at 10 ms intervals, which have been extracted from corresponding laryngograph waveforms.

The SPEECON Spanish corpus [11] for the evaluation of noise robust pitch detection algorithms contains about 1 minute of recordings for each of the 60 speakers (30 male and 30 female, from 19 to 79 years old). The speech was recorded in three types of environments (car, office and public places), with four microphones: a close-talking microphone and three distant microphones placed at different distances from the speaker. This database is also provided with reference pitch values that were obtain in a two-step process. In the first step, the close-talking channel speech signals have been automatically pitch-marked (epoch marked). Then, in the next step, accurate manual rechecking and correcting of pitch marks was performed thus resulting in the reference pitch-marked database.

TABLE I
Summary of pitch detection algorithms evaluated

Name in this paper	Algorithm	Toolkit
ACF (Praat)	ACF [1]	Praat[a][25]
AMDF (Snack)	AMDF [19]	Snack library[b]
Aurora (ETSI)	Aurora [7]	ETSI[c]
CCF (Praat)	CCF	Praat[a]
Kaldi	enhanced RAPT [9]	Kaldi[d]
Martin (JSnoori)	Spectral-based [22]	JSnoori[e]
NDF (STRAIGHT)	NDF [8]	STRAIGHT[f]
REAPER	REAPER	REAPER[g]
RAPT (SPTK)	RAPT [2]	SPTK[h]
RAPT (Snack)	RAPT [2]	Snack library[b]
SHS (Praat)	SHS [24]	Praat[a]
SRPD (ESTL)	SRPD [20], [21]	ESTL[i]
SWIPE (JSnoori)	SWIPE [6], [23]	JSnoori[e]
SWIPE (SPTK)	SWIPE [6], [23]	SPTK[h]
TEMPO (STRAIGHT)	TEMPO [4], [5]	STRAIGHT[f]
YIN (AdC)	YIN [3]	YIN[j]
YIN (JSnoori)	YIN [3]	JSnoori[e]

[a] http://www.fon.hum.uva.nl/praat/
[b] http://www.speech.kth.se/snack/
[c] http://webapp.etsi.org/WorkProgram/Report_WorkItem.asp?WKI_ID=17236
[d] https://github.com/kaldi-asr/kaldi
[e] https://raweb.inria.fr/rapportsactivite/RA2015/multispeech/uid43.html
[f] http://www.wakayama-u.ac.jp/∼kawahara/STRAIGHTadv/
[g] https://github.com/google/REAPER
[h] http://sp-tk.sourceforge.net/
[i] http://www.cstr.ed.ac.uk/projects/speech_tools/
[j] http://audition-backend.ens.fr/adc/

B. Simulated noisy data

As the PTDB-TUG corpus contains only clean speech signals, in most of the following experiments, some noise is added. Noise recordings are taken from the NOISEX-92 corpus [27], which is often used in the field of noise robust automatic speech recognition. The following noise types have been considered: *babble* (people speaking in a canteen), *factory1* (sound recorded near plate-cutting and electrical welding equipment in a factory), *factory2* (sound recorded in a car production hall), *pink* (acquired by sampling a high-quality analog noise generator (Wandel & Goltermann), yielding equal energy per 1/3 octave), and *white* (acquired by sampling the same analog noise generator, with equal energy per Hz bandwidth).

All these noise signals have been downsampled to 16 kHz before being added to the clean speech signal with the filtering and noise adding tool (FaNT)[1] [28]. It allows adding noise at a given signal-to-noise ratio, and it was used here for adding

[1] http://dnt.kr.hs-niederrhein.de/index964b.html

the noise recordings at 20 dB, 15 dB, 10 dB, 5 dB, 0 dB and -5 dB SNR levels.

C. SNR Estimation

As SPEECON utterances are real noisy speech signals, their SNR have been estimated using the SNR estimation tool from ISIP[2]. The estimation process relies on the cumulative density function of the distribution of the frame energy. The energy corresponding to the 20% lowest values provides an estimate of the noise energy. The energy corresponding to the 85% lowest values provides an estimate of the speech+noise energy. These two values are then used to compute the SNR of the speech utterance. Note that the choice of 20% and 85% for the thresholds is arbitrary.

D. Evaluation measures

Standard pitch detection evaluation measures are used (e.g., [29]). However, because of space constraints, only the two following measures are analyzed. The *voicing decision error* (VDE) is the proportion of frames for which a voicing decision error is made: voiced frames classified as unvoiced and unvoiced frames classified as voiced. The *F0 frame error* (FFE) provides a global error measure, it is the proportion of frames for which an error is observed: either voicing decision error or gross pitch error (i.e., voiced frames classified as voiced and for which the estimated F0 differs from the reference F0 by more than 20%).

Each algorithm provides F0 estimates at 10 ms intervals with some time offset that depends on its implementation. Thus, for each algorithm, an optimal time offset with respect to the annotated database is determined using a grid search with clean or close-talk data; the optimal time offset is the one which minimizes the corresponding F0 frame error.

E. Configurations

All signals have been processed at 16 kHz, after downsampling when necessary. And, for each pitch detection algorithm, the F0 values are computed at 10 ms intervals.

For almost all pitch detection algorithms (i.e., ACF, AMDF, CCF, Kaldi, RAPT (SPTK & Snack versions), SHS, SRPD, SWIPE (JSnoori & SPTK versions), and YIN (AdC)), the default or recommended values have been used, with a rather large F0 range ($[60-600]$ Hz). For the STRAIGTH versions (TEMPO & NDF), according to the returned parameters, the F0 range was equal to $[40-800]$, although it was set smaller as input. For REAPER the F0 range was set to $[40-500]$ according to the default values specified in epoch_tracker.h. With JSnoori, the gender was specified as unknown. Note also that no parameter is available for the Aurora approach. The Kaldi toolkit provides an F0 estimate for each frame, whether voiced or not; so, an arbitrary threshold of 0.5 was applied on the normalized cross correlation function to make a voiced/unvoiced decision for each frame. A last point to mention is that for the NDF (STRAIGHT) approach, the pitch is estimated every millisecond, and then a downsampling is applied to obtain the pitch values every 10 ms. This leads to much better results than a direct computation every 10 ms, mainly because of a much lower voicing decision error rate.

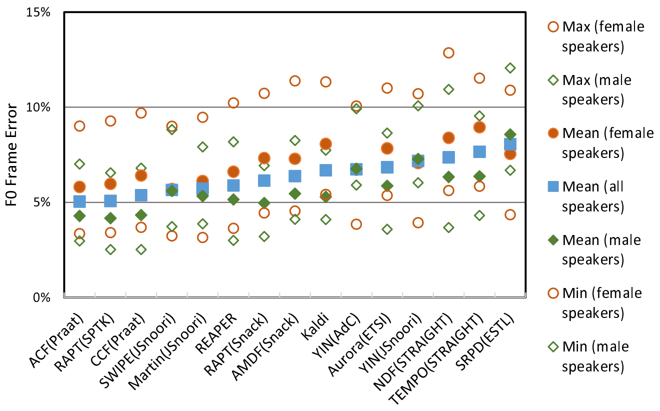

Fig. 1. F0 Frame Errors on PTDB-TUG clean data.

IV. Performance evaluation

The various algorithms are evaluated on clean speech data, on simulated and on the real noisy data.

A. On clean speech data

Figure 1 reports the F0 frame errors (FFE) for the PTDB-TUG clean data. However as SWIPE (SPTK) leads to exactly the same performance as SWIPE (JSnoori), it is not mentioned in the Figure. Moreover, as the results with SHS (Praat) were much worse than those of ACF and CCF, they are not reported neither. On the figure, systems are ranked according the average FFE over all 20 speakers (blue square), which ranges from 5.05% for ACF (Praat) up to 8.06% for SRPD (ESTL). The green diamonds indicate the FFE over the male speakers, and the orange circles indicate the FFE over the female speakers (min, max, and average values). For most of the algorithms the average FFE over male speakers is lower than that over female speakers, and there are rather large variations over the speakers.

B. On simulated noisy data

Figure 2 shows the evolution of the FFE on simulated noisy data (average over the five types of noise) with respect to the SNR level specified when adding the various noise signals to the clean PTDB-TUG data. Note that the order of the legend corresponds to the order of the curves for the 10 dB SNR level. Most of the approaches behave in a rather similar way, ending with an FFE around 25% for -5 dB SNR. The majority of these errors are due to a wrong voicing decision (mainly voiced frames classified as unvoiced). There are two exceptions, Martin (JSnoori) and AMDF (Snack) for which the main errors are unvoiced frames classified as voiced. Although not represented on the figure, babble noise leads to worse performance than the other types of noise (factory, pink or white); see also Figure 4 and associated comments for

[2] https://www.isip.piconepress.com/projects/speech/software/legacy/signal_to_noise/

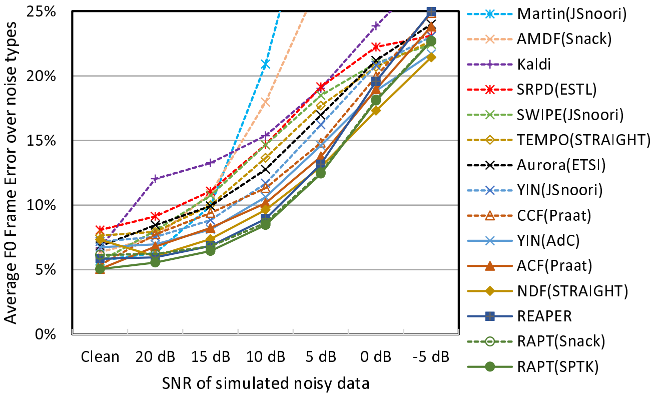

Fig. 2. F0 Frame Errors on PTDB-TUG noisy data (different types of noise were added at various SNR levels). The order in the legend matches the order of the curves at the 10 dB SNR level.

TABLE II
VOICING DECISION ERRORS (%) FOR BABBLE NOISE AT VARIOUS SNR LEVELS

Approach	VDE (v→uv + u→v)		
	no noise	10 dB SNR	0 dB SNR
RAPT(SPTK)	4.6 (1.6+3.0)	11.8 (4.7+ 7.1)	21.5 (14.3+ 7.2)
REAPER	5.0 (1.7+3.3)	8.3 (6.8+ 1.5)	20.8 (11.5+ 9.4)
NDF(STRAIGHT)	6.7 (0.7+6.0)	10,0 (8.7+ 1.3)	19.0 (16.4+ 2.6)
ACF(Praat)	4.6 (2.0+2.6)	16.2 (4.6+11.6)	25.0 (13.2+11.8)
YIN(AdC)	6.4 (3.6+2.8)	12.1 (9.0+ 3.1)	21.7 (18.9+ 2.8)

the RAPT (SPTK) algorithm results. For all algorithms and babble noise, there is a significant amount of unvoiced frames classified as voiced, whereas for the other types of noise, this type of error is much less frequent. At around 10 dB SNR, the best approaches are RAPT (SPTK and Snack versions) with 8.5% and 8.6% FFE, REAPER (8.9%), NDF (9.6%), ACF (10.2%), and then YIN (AdC) with 10.6% FFE; these are not exactly the same as the best approaches for clean speech. Table II details the VDE for these approaches, at various SNR levels (no noise, 10 dB and 0 dB), for the babble noise case only. The VDE is also decomposed according to the voiced frames classified unvoiced (v→uv) and the unvoiced frames classified voiced (u→v).

C. *On real noisy data*

Figure 3 shows the F0 frame error on the SPEECON real data. The close-talk microphone provides rather clean (Spanish) data and leads to FFEs comparable to those obtained on clean (English) PTDB-TUG data. Microphone 3, which is the farthest away from the speaker leads to the noisiest data, and also to the worst FFE. It is also interesting to note that the best performance is not always achieved with the same algorithm, this varies with the level of noise (somewhat correlated to the distance between the speaker and the microphone) and the type of noise (dependent on the environment).

D. *Real vs. simulated noisy data*

To compare the performance over the simulated noisy data (PTDB-TUG data with added noise) and over real noisy data (SPEECON data), the SNR of each utterance has been estimated using the SNR estimation tool from ISIP (Section III-C). The estimated SNRs have been quantized in bins of 5 dB width, in order to draw the curves of Figure 4 for the RAPT (SPTK) results. The solid lines corresponds to SPEECON data: car environment (green line with circles), public places (red line with squares), and office (yellow line with triangles). For any given SNR level, slight variations are observed between the various environments. The dashed lines correspond to the simulated noisy data: babble noise (red line with squares), white noise (blue line with plus marks), factory noise (green line with cross marks), pink noise (purple line with star marks). As said before, the performance for babble noise is significantly worse than for the other types of noise (factory, pink and white).

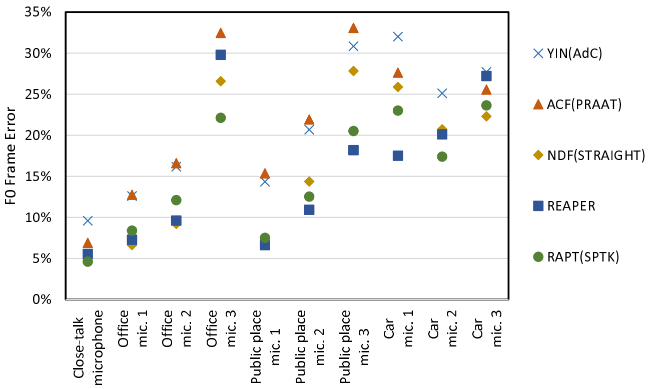

Fig. 3. F0 Frame Errors on SPEECON noisy data with respect to environment and microphone.

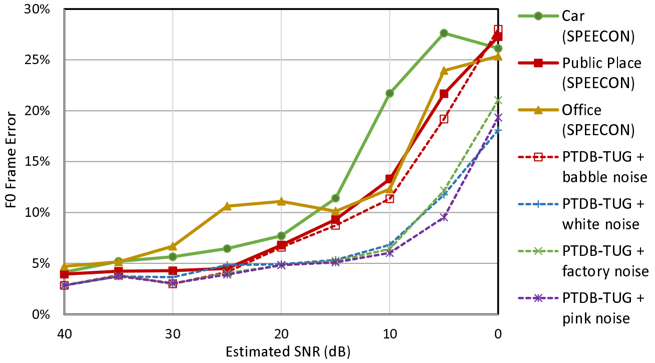

Fig. 4. F0 Frame Errors on real and simulated noisy data with respect to the estimated SNR level. The order in the legend matches the order of the curves at the 10 dB SNR level.

One very interesting fact is that the curves for SPEECON public places and for PTDB-TUG with added babble noise almost overlap each other. This means that the simulated noisy data with added babble noise provides a good representation of the type of data recorded in public places, where babble noise is also present.

ISBN 978-0-9928626-7-1 © EURASIP 2017

V. Conclusion

This paper has presented an analysis of the performance of a large bunch of pitch detection algorithms on clean data, as well as on simulated and real noisy data. On clean data, large performance variations are observed across speakers, for an average F0 frame error varying between 5% and 8% for the 15 approaches considered in Figure 1. When the level of noise increases, the performance degrades, and the voicing decision is always the main cause of errors. In many cases, the dominant error is the mis-classification of voiced frames as unvoiced. Babble noise is also more harmful than the other types of noise. However all algorithms do not behave the same way with respect to the type of noise and the SNR level. This suggests that an adequate combination of several approaches should allow a more robust pitch estimation in noisy conditions. Also, the results displayed in Figure 4 have shown that the simulated noisy speech data with babble noise provides a very good representation of the public place real data where babble noise is also the most frequent type of noise.

Acknowledgment

This work was carried out in the framework of the ProsodCorpus operation supported by the CPER LCHN (*Contrat Plan Etat Région "Langues, Connaissances et Humanités Numériques"*). Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).

References

[1] P. Boersma, "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound," in *Proc. of the Institute of Phonetic Sciences*, vol. 17, no. 1193. Amsterdam, 1993, pp. 97–110.

[2] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," in *Speech Coding and Synthesis*, W. B. Kleijn and K. K. Paliwal, Eds. Elsevier, 1995, pp. 495–518.

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

[4] H. Kawahara, H. Katayose, A. De Cheveigné, and R. D. Patterson, "Fixed point analysis of frequency to instantaneous frequency mapping for accurate estimation of f0 and periodicity," in *Eurospeech*, 1999, pp. 2781–2784.

[5] H. Kawahara, J. Estill, and O. Fujimura, "Aperiodicity extraction and control using mixed mode excitation and group delay manipulation for a high quality speech analysis, modification and synthesis system straight." in *MAVEBA*, 2001, pp. 59–64.

[6] 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.

[7] A. Sorin, T. Ramabadran, D. Chazan, R. Hoory, M. McLaughlin, D. Pearce, F. C. Wang, and Y. Zhang, "The ETSI extended distributed speech recognition (DSR) standards: client side processing and tonal language recognition evaluation," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. I, 2004, pp. 129–132.

[8] 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.

[9] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 2014, pp. 2494–2498.

[10] G. Pirker, M. Wohlmayr, S. Petrik, and F. Pernkopf, "A pitch tracking corpus with evaluation on multipitch tracking scenario," in *Interspeech*, 2011, pp. 1509–1512.

[11] "Speecon manually pitch-marked reference database for Spanish," ISLRN : 866-498-919-979-7, ELRA ID: ELRA-S0218, Catalogue ELRA (http://catalog.elra.info/).

[12] A. De Cheveigné and H. Kawahara, "Comparative evaluation of f0 estimation algorithms." in *INTERSPEECH*, 2001, pp. 2451–2454.

[13] 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.

[14] L. Sukhostat and Y. Imamverdiyev, "A comparative analysis of pitch detection methods under the influence of different noise conditions," *Journal of Voice*, vol. 29, no. 4, pp. 410–417, 2015.

[15] B. Kotnik, H. Höge, and Z. Kacic, "Evaluation of pitch detection algorithms in adverse conditions," in *Proc. 3rd Int. Conf. on Speech Prosody*, 2006, pp. 149–152.

[16] I. Luengo, I. Saratxaga, E. Navas, I. Hernáez, J. Sanchez, and I. Sainz, "Evaluation of pitch detection algorithms under real conditions," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2007, pp. IV–1057.

[17] B. Kotnik, H. Höge, and Z. Kačič, "Noise robust f0 determination and epoch-marking algorithms," *Signal Processing*, vol. 89, no. 12, pp. 2555–2569, 2009.

[18] S. Strömbergsson, "Today's most frequently used f 0 estimation methods, and their accuracy in estimating male and female pitch in clean speech," *Interspeech 2016*, pp. 525–529, 2016.

[19] M. Ross, H. Shaffer, A. Cohen, R. Freudberg, and H. Manley, "Average magnitude difference function pitch extractor," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 22, no. 5, pp. 353–362, 1974.

[20] Y. Medan, E. Yair, and D. Chazan, "Super resolution pitch determination of speech signals," *IEEE Trans. on Signal Processing*, vol. 39, no. 1, pp. 40–48, 1991.

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

[22] P. Martin, "Comparison of pitch detection by cepstrum and spectral comb analysis," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 1982, pp. 180–183.

[23] A. Camacho, "SWIPE: A sawtooth waveform inspired pitch estimator for speech and music," Ph.D. dissertation, University of Florida, 2007.

[24] D. J. Hermes, "Measurement of pitch by subharmonic summation," *The journal of the acoustical society of America*, vol. 83, no. 1, pp. 257–264, 1988.

[25] P. P. G. Boersma *et al.*, "Praat, a system for doing phonetics by computer," *Glot international*, vol. 5, 2002.

[26] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," *NASA STI/Recon Technical Report N*, vol. 93, 1993.

[27] A. Varga and H. J. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Communication*, vol. 12, no. 3, pp. 247–251, 1993.

[28] H.-G. Hirsch, "FaNT – filtering and noise adding tool," Hochschule Niederrhein, Tech. Rep., 2005.

[29] T. Drugman and A. Alwan, "Joint robust voicing detection and pitch estimation based on residual harmonics," in *Interspeech*, 2011, pp. 1973–1976.