A Pitch Marks Filtering Algorithm based on Restricted Dynamic Programming

Francesc Alías, Carlos Monzo and Joan Claudi Socoró

Dep. of Communications and Signal Theory. Enginyeria i Arquitectura La Salle
Ramon Llull University, Barcelona, Spain
{falias,cmonzo,jclaudi}@salle.url.edu

Abstract

In this paper, a generic pitch marks filtering algorithm (PMFA) is introduced in order to achieve reliable and smooth pitch marks from any input pitch tracking or marking algorithm. The proposed PMFA is a simple yet effective filtering process based on restricted dynamic programming, but very helpful for minimizing human intervention when creating large speech corpora. Moreover, this work introduces a novel pitch marking evaluation measure for directly comparing pitch marking algorithms with different location criteria. The experiments demonstrate that the proposed PFMA improves the results of the input state-of-the-art pitch tracking and marking algorithms dramatically.

Index Terms: pitch marking, restriced dynamic programming, gross error rate, PMA, PDA.

1. Introduction

Concatenative speech synthesizers are based on recorded speech corpora. Creating well-formed speech corpora is, in general, a time-consuming and laborious task [1], since human intervention is still necessary [2]. The huge amount of data involved in unit-selection speech corpora [3] makes highly recommendable conducting the labelling process automatically [1, 2]. In this context, the accuracy and reliability of the automatic labelling algorithms becomes critical to achieve high synthetic speech quality [2, 3].

Pitch marking is one of the processes involved in the automatic labelling of speech corpora. A pitch mark can be defined as the location of a signal period in a voiced speech segment, thus, it labels the fundamental periodicity of speech (see [4] for details). There are several processes involved in text-to-speech (TTS) synthesis where these marks play a key role [5, 6, 7, 8], e.g. locating concatenation points, pitch-synchronous prosodic labelling or signal modification, such as PSOLA, among other potential applications [9]. There are several well-known approaches for pitch tracking [4, 10] (aka Pitch Detection —or Tracking— Algorithms, or PDAs) and pitch marking [6, 9, 11] (aka Pitch Marking Algorithms or PMAs). Most of them are based on speech signal analysis, while others utilize its corresponding electroglottographic (EGG) signal. Anyhow, achieving a robust estimation of pitch marks is a difficult task since human speech is very diverse and only *pseudo*-stationary, i.e. the signal is no *perfectly* periodic [4, 11].

In order to substantially enhance the accuracy and robustness of current PMAs and towards simplifying the speech corpus building process, this paper introduces a pitch marks filtering algorithm based on restricted dynamic programming that can be applied to any PDA or PMA. Subsequently, due to the lack of an unified evaluation measure for validating the performance of PMAs, a new measure for evaluating pitch marking reliability is also introduced. Finally, the proposal is evaluated on several objective experiments.

2. Related work

The pitch marks are located into the speech period according to some characteristic property of the signal (i.e. *local criterion*), such as the maximum positive peak [5, 6] or the glottal closure instant —estimated from the speech signal [12], the wavelet transform [13], or the EGG signal [9, 11]—, among others. Whatever the local criterion is, the pitch marks are to follow the fundamental periodicity of the speech signal [6].

However, locating the pitch marks just by considering the local criterion may lead to labelling errors [6, 7], e.g. at unit boundaries [11], at sonority transitions [9], or on mixed sonority units, such as voiced fricatives, among others. Most of the current PDAs and PMAs resemble the methodology proposed in [14], which consists of three steps: pre-processing, pitch candidate generation, and post-processing by, generally, dynamic programming (DP), e.g. [6, 8, 9]. The post-processing step is devoted to resolve local inconsistencies according to a *global criterion*, i.e. a cost function [6], which usually takes into account: *i*) N candidate markers satisfying the local criterion (e.g. $N = 2$ [15], $N = 3$ [8, 12]), *ii*) the frame periodicity indicated by the PDA [7, 9, 15], and, sometimes, *iii*) the correlation between adjacent signal periods [7, 12, 14]. DP is then used to find the sequence of candidate pitch markers which minimizes the cost function, i.e. which optimally satisfies the consistency requirements, yielding a fine pitch marks tuning.

In our opinion, these preceding approaches still present several open issues. On one hand, the cost function requires a meticulous adjustment, i.e. several parameters are to be empirically determined (see [6, 7, 8, 9, 12]). And, on the other hand, PDA errors are dragged into the PMA, reducing its efficiency, mainly when large errors are present in the pitch track [6, 7]. In the following section, we introduce a new pitch marking strategy, which faces these weaknesses and yields reliable and smooth pitch marks.

3. Pitch Marks Filtering Algorithm

Following a previous work [16], we introduce a pitch marks filtering algorithm (PMFA), as a step further towards minimizing manual inspection of results when developing speech corpora for TTS synthesis. However, the PMFA can be applied to any other task related to reliable pitch marking. The main goal of PMFA is filtering the errors of the input pitch marks ($m^i(k)$) in order to generate a reliable (i.e. smooth) sequence of final marks ($m^f(k)$), following a predefined local criterion (see figure 1). PMFA is based on restricted dynamic programming (RDP), since the dynamic search is limited by a maximum frame-to-frame slope constraint (S_{max}) [5, 17]. Furthermore, this algorithm can be applied to any PMA, and also to any PDA by introducing a simple PMA (sPMA in figure 1) —a similar idea to PMAs making use of any PDA [6, 7, 9].

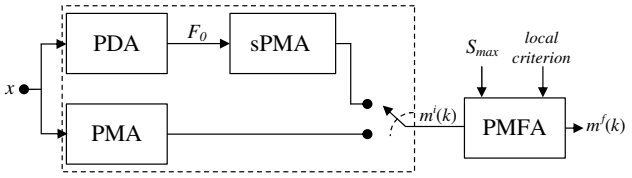

Figure 1: Block diagram of the pitch marking process from an input signal x, incorporating the PMFA as a post-processing stage.

The PMFA generates the $m^f(k)$ ($1 \leq k \leq K$, where K denotes the total number of marks) after a two-phase process based on RDP. Firstly, the input pitch errors are filtered and, secondly, the pitch marks are located following the selected local criterion.

3.1. Filtering pitch errors

The first step of the PMFA is devoted to filter the errors of $m^i(k)$ due to insertions, deletions or outliers (i.e. spurious pitch marks or wrong F_0 values). To that effect, the input pitch marks vector is windowed (*boxcar* window) at constant frame rate, obtaining T analysis frames with different number of marks per frame (e.g. when using a $5ms$ window for a speech signal with an F_0 range of $[50, 550]$ Hz, each frame will contain from 0 to 3 marks). These analysis frames are subsequently used to estimate the signal periodicity p per frame by means of algorithm (1).

$$
\begin{aligned}
&\mathbf{P}\,[i,j] := 0,\ \forall\, 1 \leq i \leq (p_{max} - p_{min} + 1),\ 1 \leq\ j \leq T;\\
&j := 1;\\
&while\ \big((j \leq T)\&(2 \leq k \leq K)\big)\\
&\quad \forall m^i(k) \in t\\
&\quad\quad if(p_{min} \leq \big(m^i(k) - m^i(k-1)\big) \leq p_{max}),\ then\\
&\quad\quad\quad \mathbf{P}\,\big[\big(m^i(k) - m^i(k-1)\big) - p_{min} + 1, j\big] := 1;\\
&\quad\quad end\\
&\quad t := t + 1;\\
&end
\end{aligned}
\tag{1}
$$

where $\mathbf{P}$ is a $[(p_{max} - p_{min} + 1) \times T]$ binary matrix containing non-null cells ($\mathbf{P}\,[i,j] = 1$) at rows i corresponding to differences between two consecutive pitch marks of frame j, i.e. the local periodicity of row i is $p_{min} + i - 1$. Therefore, if the frame is *completely* periodic, its corresponding column of $\mathbf{P}$ will only have one non-null row, in contrast to frames with no clear periodicity, containing several candidate rows (insertions or transitions) or being entirely null (deletions or unvoiced). The former rows will be useful for disambiguating the latter, according to the context information that this analysis introduces to the subsequent RDP algorithm. Moreover, the outliers will be omitted thanks to the p range constraint ($p_{min} \leq p \leq p_{max}$).

The RDP algorithm applied to the binary matrix $\mathbf{P}$ presents two main particularities: *i)* the *forward* process is restricted by a maximum frame-to-frame periodicity variation (S_{max}), in order to control periodicity fluctuations across the *trellis* structure, and *ii)* the *backward* process is conducted by an *n*-backtracking algorithm, since there may be *n* cells yielding the same maximum accumulate metric due to the binary contents of $\mathbf{P}$ (i.e. binary cost). The best path among the several candidates is then selected as the path attaining the lowest global variation, i.e. the highest smoothness of the pitch curve throughout the speech signal.

3.2. Locating pitch marks

Once the periodicity of each frame is obtained by RDP (i.e. the best path on $\mathbf{P}$), the next step is focused on locating the corresponding pitch marks throughout the speech signal. To that effect, the RDP is applied for a second time, following the same scheme described in [5], and hence, considering a larger S_{max} in order to adjust the marks at sample level. In this work, the maximum positive peak is selected as the location criterion [5, 6]. However, PMFA allows any other placement criterion.

4. Evaluation measures

In order to analyze the performance of PDAs, there exists a well-known evaluation measure called *gross error rate* (GER), which computes as errors the estimated F_0 values 20% higher or lower than the reference values (the so-called *ground truth*) [4, 10]. In contrast, there is no *standard* testing measure to evaluate the performance of PMAs. For instance, [6] states that the PMA should be measured indirectly from the performance of its application, while other works evaluate the quality of the PMA with respect to a reference set of marks (generally extracted from its corresponding EGG signal), by direct comparison [15, 8], or by allowing a relative difference margin [12, 13]. Anyhow, all these proposals can only be considered if the evaluated and the reference marks follow the same criterion for locating the marks. To deal with this problem, [12, 18] propose to first align the marks before conducting the comparison. Nevertheless, the errors due to misalignments may lead to unreliable evaluation results.

In this paper, a new evaluation measure for validating and comparing the performance of PMAs with different locating criteria is introduced. To that effect, the relative periodicity differences (p_r) of consecutive pitch marks are considered instead of their specific location. If this difference is greater than a predefined threshold γ, that pitch mark is considered as erroneous. This measure is called *Gross Pitch Marks Error Rate* (GPMER), and somehow can be defined as a *fine* GER (see equation 2), since GER only compares PDA performance at frame level. The reference pitch marks guide the comparison process across the signal, thus, the insertions (too many marks) and deletions (no marks) are easily detected. To avoid biasing the evaluation measure, both insertions and deletions are just computed as *one* error.

$$
\mathrm{GPMER}(\%) = \frac{\#\Big(\frac{|p'_r - p_r|}{p_r}\Big) > \gamma}{\#p_r} \cdot 100 \tag{2}
$$

where p'_r is the evaluated local periodicity and p_r is the corresponding reference value, the specific position of which guides the comparison process so as to deal with the different number of pitch marks compared —in this work, $\gamma = 0.2$, as in classic GER.

5. Experiments

The following experiments are devoted to analyze the performance of the PMFA in terms of GER and GPMER (the unvoiced and voiced error rates are also included in the depicted results, due to no voicing estimation conducted by PMFA). The analysis is conducted on two databases. First, a Spanish speech corpus (DB1) recorded by a female professional speaker using three different speaking styles (and, thus, different F_0 ranges): happy (F_0: $\mu = 271$Hz, $\sigma = 89$Hz), neutral (F_0: $\mu = 167$Hz, $\sigma = 41$Hz) and sensual (F_0: $\mu = 134$Hz, $\sigma = 26$Hz) sampled at 16KHz with 16-bit

Table 1: GER and GPMER (both in %) on DB1, where sXY stands for S_{max} for the first (X) and second (Y) RDP stages. In italics, values worse than the baseline inputs to the PMFA, and in boldface, the best result per sweep.

Measure	GER (%)						GPMER (%)					
Method	*Happy*		*Neutral*		*Sensual*		*Happy*		*Neutral*		*Sensual*	
Window length	$5ms$	$10ms$	$5ms$	$10ms$	$5ms$	$10ms$	$5ms$	$10ms$	$5ms$	$10ms$	$5ms$	$10ms$
RAPT	10.88		10.91		31.07		10.37		7.86		29.26	
RAPT + PMFAs13	*16.20*	*28.63*	*11.01*	*22.33*	24.05	*33.27*	*15.47*	*34.00*	6.86	*20.67*	13.76	24.58
RAPT + PMFAs24	8.88	*15.37*	7.28	10.51	21.06	23.48	6.04	*15.70*	2.86	7.08	10.93	14.28
RAPT + PMFAs34	7.61	10.42	6.61	7.98	20.64	21.65	**4.88**	8.64	2.30	3.72	10.37	12.45
RAPT + PMFAs37	**7.54**	10.83	**6.08**	8.01	19.93	21.25	5.39	9.55	**2.19**	4.06	**9.66**	12.99
RAPT + PMFAs48	7.88	9.17	6.59	7.43	20.21	20.79	5.89	7.56	2.43	3.24	10.28	12.55
RAPT + PMFAs68	7.76	**8.22**	6.21	6.76	**19.90**	**20.12**	5.88	**6.59**	2.28	**2.62**	9.81	**12.09**
RAPT + PMFAs79	7.87	8.24	6.19	**6.74**	20.02	20.20	6.54	7.09	2.32	2.64	9.83	12.16
RAPT + PMFAs912	8.59	8.89	6.38	6.84	20.46	20.35	8.95	9.41	2.61	2.95	10.18	12.57
YIN	17.44		22.35		36.86		8.06		5.16		22.06	
YIN + PMFAs13	16.82	*28.59*	12.10	*24.37*	23.18	32.79	*16.73*	*34.88*	*8.09*	*22.38*	13.97	*24.00*
YIN + PMFAs24	9.56	15.82	7.73	11.42	20.41	22.58	7.28	*17.07*	3.44	*8.03*	11.70	13.61
YIN + PMFAs34	**8.43**	11.16	7.14	8.39	20.26	21.03	**6.21**	*10.04*	**2.87**	4.41	**11.36**	**12.05**
YIN + PMFAs37	8.61	11.50	**6.98**	8.60	20.10	20.70	7.61	*11.07*	3.07	4.83	12.33	12.50
YIN + PMFAs48	8.68	10.06	7.06	7.70	**20.09**	20.23	7.80	*9.25*	3.14	3.75	12.28	12.13
YIN + PMFAs68	8.75	**8.99**	7.14	**7.15**	20.16	**20.03**	8.04	***8.42***	3.20	**3.14**	12.49	12.14
YIN + PMFAs79	8.87	9.06	7.13	7.19	20.35	**20.03**	*8.71*	*8.95*	3.30	3.23	12.72	12.33
YIN + PMFAs912	9.60	9.73	7.44	7.55	20.76	20.54	*10.85*	*11.21*	3.75	3.80	13.44	13.09
SHRp	22.45		25.16		38.85		12.25		8.86		25.55	
SHRp + PMFAs13	18.28	*31.87*	12.94	24.98	25.11	35.24	*16.72*	*36.28*	8.28	*22.92*	15.97	*27.34*
SHRp + PFMAs24	9.49	16.87	8.64	12.02	**23.09**	24.47	5.96	*16.62*	4.00	8.17	**14.79**	15.48
SHRp + PMFAs34	**8.28**	11.30	**8.02**	9.09	23.11	22.97	**4.69**	8.83	**3.46**	4.63	14.98	**14.20**
SHRp + PMFAs37	8.87	11.71	8.15	9.17	23.82	23.11	6.68	9.86	4.04	5.12	16.88	14.92
SHRp + PMFAs48	8.67	9.62	**8.02**	8.20	23.30	22.93	6.48	7.58	3.93	4.13	16.10	15.07
SHRp + PMFAs68	8.85	**8.58**	8.16	**7.83**	23.69	**22.77**	6.91	**6.43**	4.10	**3.71**	16.74	15.14
SHRp + PMFAs79	9.21	8.73	8.32	7.86	24.04	22.89	7.91	7.13	4.33	3.80	17.35	15.44
SHRp + PMFAs912	10.37	9.73	8.91	8.35	24.91	23.76	10.76	9.74	5.09	4.46	18.47	16.78

resolution, as a unit selection TTS corpus. And second, the Keele database [19] (DB2), which contains speech from 10 speakers (5 males and 5 females) sampled at 20KHz with 16-bit resolution, as a well-known reference for evaluating PDAs [4, 9, 10]. DB1 provides pitch marks manually supervised and validated, while DB2 provides pitch values at $10ms$ frame rate. The duration of DB1 (2.5h) allows a much more reliable PMA evaluation ($\approx$ 900K marks are compared) with respect to previous works, e.g. with databases ranging from 1 min [18] to 8.5 min [8]. Moreover, the PMFA performance is compared to RAPT [14] (get_f0 function included in ESPS package [10]), YIN [4] and SHRp [10] as state-of-the-art input PMA and PDAs. The sPMA of [5] is employed in order to obtain the corresponding pitch marks for YIN and SHRp. Finally, the considered F_0 range for the following experiments is $[50, 550]$ Hz [10], windowing the input marks every 5 or $10ms$.

5.1. PMFA results on a large database

The first experiment is devoted to analyze the PMFA performance on DB1 with respect to *i)* different algorithm configurations (S_{max} and windowing, according to the sampling frequency fs), and *ii)* different speaking styles (not just neutral speech, as typically studied). Taking into account [5, 16], the PMFA S_{max} value for the second pass of the RDP algorithm should be larger than the first phase value. For instance, [5] guesses a $S_{max} = \{3, 4\}$ (i.e. $s34$ in tables) for the first and second RDP stages, using a window of $40ms$ and $fs = 8$KHz, and [17] selects $S_{max} = 2$, with a $1ms$ frame rate for its RDP search.

Table 1 summarizes the results attained by PMFA on DB1 throughout the conducted S_{max} sweep for two windowing analysis configurations. Both GER and GPMER subtables prove that PMFA outperforms the baseline results, regardless the input PDA or PMA, the windowing or S_{max} configurations —excluding extreme S_{max} values, such as $s13$— and despite the analyzed speaking style. A more detailed examination concludes that $5ms$ analysis configuration yields better results ($s34$ and $s37$, as S_{max} best pairs) than $10ms$ ($s68$ as S_{max} best pair) all through the table. Therefore, the lower the windowing the higher the performance of PMFA, both in terms of GER and GPMER, despite incrementing the computational cost —which is not excessively critical for TTS speech corpus building. Moreover, in terms of speaking styles, the highest error rates after applying PMFA are obtained for the sensual subcorpus, mainly due to presence of *whispering*, followed by the happy domain, as a result of its higher F_0 mean value and deviation, and, finally, the lowest error rates are achieved on the neutral style. Nevertheless, the relative improvement achieved by PMFA for the sensual style is, in average, as good as the attained for the neutral style. Furthermore, notice that the best results are obtained when combining RAPT+PMFA. However, PMFA also improves the results obtained by YIN and SHRp noticeably (e.g. some of the highest relative reductions of both GER and GPMER are achieved when PMFA makes use of SHRp or YIN algorithms).

5.2. Validating the performance of the PMFA

The second experiment is conducted to ensure the validity of previous results on the Keele database after resampling at 16KHz. The pitch values included in DB2 are thresholded —70Hz for male and 120Hz for female speakers— to avoid the presence of incorrect reference values [20]. However, in contrast to [20] —and other works where only the *clear* voiced frames are considered for evaluating the proposals (e.g. [4])—, these frames are not excluded

from comparison. If the p value evaluated corresponding to these frames is beyond the threshold, it will be considered as an error. Hence, table 2 could differ from previous works that compute these *difficult* frames in terms of voiced and unvoiced error rates [10].

The PMFAs34 with $5ms$ windowing configuration has been applied to the baseline PMA and PDAs. As a result, a 75% and a 57% of GER relative average reductions for female and male speakers are achieved, respectively. Therefore, the baseline results are dramatically improved by the PMFA. Again, the combination RAPT+PMFA seems to be the best one (87% and 65% of relative female and male GER improvements), however, YIN and SHRp + PMFA also achieve very good results.

Table 2: GER (%) results on DB2 for male (M1-M5) and female (F1-F5) speakers.

Method	*M1*	*M2*	*M3*	*M4*	*M5*	*Mean*
RAPT	22.93	17.42	4.72	14.29	8.33	13.54
+PMFAs34	12.28	5.15	**0.89**	**2.47**	**3.10**	**4.78**
YIN	12.02	17.47	1.85	7.62	6.89	9.17
+PMFAs34	**11.72**	**4.48**	**0.89**	2.60	6.19	5.18
SHRp	29.30	21.29	16.91	24.97	25.37	23.57
+PMFAs34	13.94	7.65	2.33	7.17	12.67	8.75
Method	*F1*	*F2*	*F3*	*F4*	*F5*	*Mean*
RAPT	6.62	4.29	5.44	7.68	2.01	5.21
PMFAs34	**0.61**	**0.43**	**0.20**	**0.93**	0.44	**0.52**
YIN	3.72	1.07	1.88	4.21	0.38	2.25
PMFAs34	1.69	0.54	0.47	1.40	**0.27**	0.87
SHRp	10.85	6.53	10.56	20.71	8.15	11.36
PMFAs34	0.88	0.86	0.95	4.38	1.14	1.64

6. Discussion and conclusions

In this paper, a generic pitch marks filtering algorithm (PMFA) based on restricted dynamic programming (RDP) has been introduced. PMFA can be applied to any PDA or PMA to achieve reliable and smooth pitch marks. The experiments have shown that PMFA outperforms the input algorithms for practically *any* configuration and speaking style. However, $5ms$ windowing plus $s34$ RDP configuration has yielded the best results when RAPT was used as the input PMA ($fs = 16K$Hz). Moreover, $s68$ has been the best S_{max} pair for $10ms$ windowing, due to half frame rate.

Furthermore, a new measure for evaluating PMAs inspired on GER, called GPMER, has also been introduced for comparing the performance of different PMAs, despite their criterion for locating pitch marks. Notice that the lower absolute values of GMPER with respect to GER are due to the larger number of comparisons (every two pitch marks *vs.* every frame). GPMER has been defined as a fine GER, since, for instance, GPMER avoids missing erroneous frames, e.g. containing an over-marked segment plus an under-marked segment (a transition sonority frame) that yields an average periodicity value close to the reference. Moreover, the pattern attained by GPMER on DB1 for the baseline algorithms (*Neutral* $<$ *Happy* $<$ *Sensual*) is better correlated with *actual* results than GER according to the different levels of difficulty.

Finally, note that one of the main strengths of this approach is its simplicity. PMFA *i)* does not use any complex cost function —just S_{max} values have to be tuned—, *ii)* filters PDA or PMA errors using a simple binary voting scheme, and *iii)* makes no attempt to discriminate speech sonority —unvoiced periods become smooth transitions between voiced neighbours [5]. Nevertheless, there is room for further research towards improving PMFA performance on speaking styles involving complex pitch marking.

7. Acknowledgements

This work was partly supported by the SALERO project (IST-FP6-027122) of the European Commission. The authors would like to thank Alain de Cheveigné and Antonio Bonafonte for providing the YIN and RAPT algorithms, respectively, and Xuejing Sun for giving free access to the SHRp code.

8. References

[1] T. Saito and M. Sakamoto, "A VoiceFont Creation Framework for Generating Personalized Voices," *IEICE Transactions*, vol. 88-D, no. 3, pp. 525–534, 2005.

[2] R.A.J. Clark, K. Richmond, and S. King, "Multisyn voices from ARCTIC data for the Blizzard challenge," in *Proc. of InterSpeech*, Lisbon, Portugal, 2005, pp. 101–104.

[3] P. Taylor, A.W. Black, and R. Caley, "Building Voices in the Festival Speech Synthesis System," in *http://festvox.org/bsv/*, 2000-2003.

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

[5] V. Goncharoff and P. Gries, "An algorithm for accurately marking pitch pulses in speech signals," in *Proc. of IASTED Int. Conf. on Signal and Image Processing*, Las Vegas, USA, 1998, pp. 281–284.

[6] R. Veldhuis, "Consistent Pitch Marking," in *Proc. of ICSLP*, Beijing, China, 2000, vol. 3, pp. 207–210.

[7] V. Colotte and Y. Laprie, "Higher precision pitch marking for TD-PSOLA," in *Proceedings of the XI European Signal Processing Conference (EUSIPCO)*, Toulouse, France, 2002, vol. 1, pp. 419–422.

[8] C-Y. Lin and J-S. R. Jang, "A two-phase pitch marking method for TD-PSOLA synthesis," in *Proc. of ICSLP*, Jeju Island, South Korea, 2004, pp. 1189–1192.

[9] P. Dikshit, S.A. Zahorian, and Nagulapati, S., "A two-phase pitch marking method for TD-PSOLA synthesis," in *Proceedings of ICASSP*, Philadephia, USA, 2005, vol. 1, pp. 233–236.

[10] X. Sun, "Pitch determination and voice quality analysis using subharmonic-to-harmonic ratio," in *Proc. of ICASSP*, Orlando, USA, 2002, vol. 1, pp. 333–336.

[11] A. Ferencz, J. Kim, Y-B. Lee, and J-W. Lee, "Automatic pitch marking and reconstruction of glottal closure instants from noisy and deformed electro-glotto-graph signals," in *Proc. of ICSLP*, Jeju Island, South Korea, 2004, pp. 2437–2440.

[12] A. Kounoudes, P.A. Naylor, and M. Brookes, "The DYPSA algorithm for estimation of glottal closure instants in voiced speech," in *Proceedings of ICASSP*, Orlando, USA, 2002, vol. 1, pp. 349–352.

[13] M. Sakamoto and T. Saito, "An Automatic Pitch-Marking Method using Wavelet Transform," in *Proc. of ICSLP*, 2000, pp. 650–653.

[14] D. Talkin, "A Robust Algorithm for Pitch Tracking (RAPT)," in *Speech Coding and Synthesis*, W. B. Kleijn and K. K. Paliwal, Eds., chapter 14, pp. 495–518. Elsevier Science, Amsterdam, NL, 1995.

[15] J-H. Chen and Y-A. Kao, "Pitch Marking Based on an Adaptable Filter and a Peak-Valley Estimation Method," *Comput. Linguistics and Chinese Language Processing*, vol. 6, no. 2, pp. 1–12, 2001.

[16] F. Alías and I. Iriondo, "Automatically pitch marking based on dynamic programming," *Procesamiento del Lenguaje Natural*, vol. 27, pp. 225–231, 2001 *(In Spanish)*.

[17] J-P. Hosom, "$F0$ Estimation for Adult and Childrens Speech," in *Proc. of InterSpeech*, Lisbon, Portugal, 2005, pp. 317–320.

[18] S. Harbeck, A. Kießling, R. Kompe, H. Niemann, and E. Nöth, "Robust pitch period detection using dynamic programming with an ANN cost function," in *Proc. of EuroSpeech*, Madrid, Spain, 1995, vol. 2, pp. 1337–1340.

[19] F. Plante, G. Meyer, and W.A. Ainsworth, "A pitch extraction reference database," in *Proc. of EuroSpeech*, 1995, pp. 837–840.

[20] K. Kasi and S.A. Zahorian, "Yet another algorithm for pitch tracking," in *Proc. of ICASSP*, Orlando, USA, 2002, vol. 1, pp. 361–364.