Real-time Pitch Tracking Based on Combined SMDSF

Jian Liu, Thomas Fang Zheng, Jing Deng and Wenhu Wu

Center for Speech Technology, State Key Laboratory of Intelligent Technology and Systems
Department of Computer Science & Technology, Tsinghua University, Beijing, 10084, China
{liuj, fzheng, dengj}@cst.cs.tsinghua.edu.cn, wuwh@tsinghua.edu.cn

Abstract

This paper presents a novel pitch tracking method in the time domain. Based on the difference function as used in YIN -- referred to as the sum magnitude difference square function (SMDSF) thereinafter -- we propose two modified types of SMDSFs, with several methods presented to calculate these SMDSFs efficiently and without bias by using the FFT algorithm. In pitch estimation, every type of SMDSF has its own estimation error characteristics. By analyzing these characteristics, we define a new function which combines the foresaid two types of SMDSFs to prevent estimation errors. A new, relatively accurate, and real-time pitch tracking algorithm is then proposed which does not need any extra pre-processing and post-processing. Experimental results show that this proposed algorithm can achieve remarkably good performance for pitch tracking.

1. Introduction

Pitch is an important parameter for speech recognition, compression, and synthesis. It plays a significant role in both the production and the perception of speech. The period of a periodic signal can be defined as the smallest positive member of the infinite set of time shifts that leaves the signal invariant. But real-world speech is not a perfectly periodic signal, it is actually quasi-periodic (i.e. a non-stationary time-varying signal). Due to its time-varying and non-stationary properties, speech signals are always processed with short-term techniques.

There are numerous time-domain short-term based pitch tracking algorithms [8], in which different functions are used, such as autocorrelation functions and difference functions. A typical autocorrelation function is the normalized autocorrelation function (NACF), which is used in many algorithms [3, 6]. The average magnitude difference function (AMDF) [4] is another well-known function. Recently several difference functions based on modified AMDF have been presented, such as the cumulative mean normalized difference function (CMNDF) [2] and the circular average magnitude difference function (CAMDF) [5].

Sampling, windowing and strong harmonic content are known to be the key factors that limit the accuracy of pitch estimation. Two typical kinds of errors in pitch estimation are period-doubling errors and period-halving errors, corresponding to "too low" errors and "too high" errors in YIN [2]. Many pitch estimation algorithms have methods to prevent these two types of errors from taking place. These methods generally consist of two stages: a pre-processing stage, using, for example, low-pass filtering [2, 7] and a post-processing stage using dynamic programming [3, 6]. However, only one certain type of time-domain functions (ACF, AMDF, et c.) is used in these algorithms during pitch candidate generation, which inevitably limits the accuracy of pitch estimation. Different time-domain functions used in pitch estimation lead to different error distributions. Some functions have a higher doubling error rate while others have a higher halving error rate. By analyzing the error characteristics of several existing SMDSFs, a combined function which has the common merit of several existing SMDSFs is proposed, and is shown to have the lowest error rate for pitch estimation.

The motivation of this paper is to find the optimal time-domain function for pitch estimation that can be calculated efficiently in real time, and based on this to design a simple yet efficient pitch tracking algorithm. The rest of the paper is organized as follows. In Section 2 we discuss several SMDSFs for pitch estimation and analyze their error characteristics. In Section 3 we propose a novel pitch tracking algorithm and describe the details. The experiments and results are given in Section 4 with discussion and analysis in the last section.

2. Definitions of SMDSFs

2.1. Bidirectional SMDSF

The sum magnitude difference square function (SMDSF) is defined as

$$d_t(\tau) = \sum_{j=t}^{N-1+t} \left(s(j) - s(j+\tau)\right)^2 \quad (1)$$

where s denotes the speech sample sequence, j is the time index (or sample point index), and N is the size of the analyzing frame. The definition in (1) was proposed in YIN [2].

The calculation of Eq (1) is computationally expensive. Two approaches were proposed in YIN. The first one uses a recursive formula over time with a time complexity of $O(N^2)$, which is slow at a reduced frame rate. The second one employs the FFT algorithm, but is approximate with an unwanted bias.

In order to calculate Equation (1) efficiently using the FFT algorithm without any bias, we expand Equation (1) into

$$d_t(\tau) = \sum_{j=t}^{t+N-1} s^2(j) + \sum_{j=t}^{t+N-1} s^2(j+\tau) - 2\sum_{j=t}^{t+N-1} s(j)s(j+\tau) \quad (2)$$

It is well known that in short-term analysis the speech sample sequence is segmented into frames. One frame of the speech sample sequence at time index t can be defined as

$$s_t(j) = \begin{cases} s(t+j), & j = 0,1,...N-1 \\ 0, & otherwise \end{cases} \quad (3)$$

Accordingly Equation (2) becomes

$$d_t(\tau) = a_t(0) + r_t(\tau) - 2\left(a_t(\tau) + c_t(\tau)\right) \quad (4)$$

where $a_t(\tau)$, $r_t(\tau)$, and $c_t(\tau)$ are as follows

$$a_t(\tau) = \sum_{j=0}^{N-1} s_t(j)\, s_t(j+\tau) \tag{5}$$

$$r_t(\tau) = \begin{cases} a_t(0), \tau = 0 \\ r_t(\tau-1) - \left(s(t+\tau-1)\right)^2 + \left(s(t+N+\tau-1)\right)^2, otherwise \end{cases} \tag{6}$$

$$c_t(\tau) = \sum_{j=0}^{N-1} s_t(j+N-\tau)\, s_{t+N}(j) \tag{7}$$

Equation (5) is the autocorrelation function of the frame at time index t, which can be calculated efficiently using the FFT algorithm. Equation (6) is the power of the speech frame at current lag τ, which can be calculated recursively over τ in linear time. Equation (7) is the cross-correlation between two frames, which can also be calculated using FFT. Finally we can calculate Equation (1) with a time complexity of $O(N \log_2(N))$.

According to the definition, we refer to the function defined by Equation (1) as a left-to-right SMDSF. Similarly we can define a right-to-left SMDSF as

$$d_t'(\tau) = \sum_{j=t}^{t+N-1} \left(s(j+N) - s(j+N-\tau)\right)^2 \tag{8}$$

In a similar way we expand Equation (8) into

$$d_t'(\tau) = a_{t+N}(0) + r_t'(\tau) - 2\left(a_{t+N}(\tau) + c_t(\tau)\right) \tag{9}$$

where $a_{t+N}(\tau)$ and $c_t(\tau)$ are as in Equations (5) and (7), respectively, while $r'_t(\tau)$ is defined as

$$r_t'(\tau) = \begin{cases} a_{t+N}(0), \tau = 0 \\ r(\tau-1) - \left(s(t+2N-\tau)\right)^2 + \left(s(t+N-\tau)\right)^2, otherwise \end{cases} \tag{10}$$

Note that the right-to-left SMDSF can also be calculated efficiently using FFT, and that the left-to-right SMDSF and the right-to-left SMDSF are different functions, at least in form, at time index t. Figure 1 illustrates the difference between the two functions. The left-to-right SMDSF and the right-to-left SMDSF have an interior relation as described by $d_t'(\tau) = d_{t+N-\tau}(\tau)$.

Considering that either the left-to-right or the right-to-left SMDSF might introduce estimation errors, we propose a bidirectional SMDSF as

$$D_t(\tau) = \left(d_t(\tau) + d_t'(\tau)\right)/2 \tag{11}$$

Based on this equation, the raw estimated pitch at time index $t+N$ is

$$p(t+N) = \arg\min_{p_{\min} \le \tau \le p_{\max}} D_t(\tau) \tag{12}$$

where $p_{\max}$ and $p_{\min}$ are the possible maximum and minimum pitch values respectively. Moreover, the well-known parabolic interpolation can be used to estimate the pitch value.

Speech can be thought of a stationary signal in a short-term sense. But actually, the pitch changes a little in one certain speech segment. The estimated pitch at time index $t+N$ is related to two adjacent frames $s_t(j)$ and $s_{t+N}(j)$. If we consider the time index as a bidirectional variable, the right-to-left SMDSF of two adjacent frames $s_t(j)$ and $s_{t+N}(j)$ should be equal to the left-to-right SMDSF of two adjacent frames $s_{t+N}(j)$ and $s_t(j)$, and this indeed is the case. So pitch estimation using bidirectional SMDSF can lead to an average pitch value during two adjacent frames. Table 1 gives the gross error rate (GER) (see Section 4 for the definition of GER and the details of experiment conditions) of pitch estimation using different SMDSFs without any pre-processing and post-processing. The experiment shows that pitch estimation based on the bidirectional SMDSF has the lowest GER. We can also see that the proposed bidirectional pitch estimation method causes more doubling errors than halving errors.

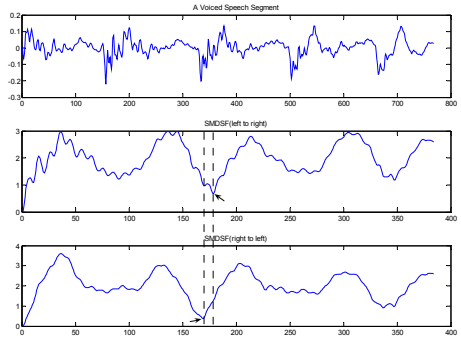

Figure 1: The difference between left-to-right SMDSF and right-to-left SMDSF

Table 1: Gross error rates and doubling/halving error rates for pitch estimation using the left-to-right SMDSF, the right-to-left SMDSF, and the bidirectional SMDSF

Method	GER (%)	Doubling/Halving (%)
Left-to-right	7.3	5.8/1.5
Right-to-left	12.4	6.9/5.5
Bidirectional	**6.6**	**4.9/1.7**

2.2. Circular SMDSF

In the previous section, we defined a bidirectional SMDSF where the doubling errors and the halving errors are not balanced. If we can have another method with relatively more halving errors, it can be imagined that by combining it with the previous method, we can have a novel method with lower doubling and halving errors. A new type of SMDSF for this purpose is defined as

$$D_t'(\tau) = \sum_{j=t}^{t+2N-1} \left(s(j) - s\left(t + (j+\tau-t)\%(2N)\right)\right)^2 \tag{13}$$

where "%" represents the modulo operation. The function defined by Equation (13) is referred to as a circular SMDSF due to the modulo operation on the sample point index. The analyzing frame size used in the circular SMDSF is $2N$, two

times the frame size used in the bidirectional SMDSF. We can derivate Equation (14) from Equation (13)

$$D'_t(\tau) = 2a'_t(0) - 2\left(a'_t(\tau) + a'_t(2N-\tau)\right) \quad (14)$$

where $a'_t(\tau)$ has a similar definition to Equation (5) except that the analyzing frame size is $2N$ instead of N.

The time complexity for the calculation of the circular SMDSF is $O(N \log_2(N))$. The raw pitch estimation using the circular SMDSF is similar to that using the bidirectional SMDSF and Equation (12) can be applied to estimate the pitch at time index $t+N$.

The circular SMDSF has different characteristics from the bidirectional SMDSF as illustrated in Figure 2. The circular SMDSF has a more obvious ascending trend with larger lag than the bidirectional SMDSF. If the frame size is not an integer multiple of the pitch period, the terms after the modulo operation may make the circular SMDSF value larger. This characteristic can prevent doubling errors during the pitch tracking, but sometime introduce more halving errors.

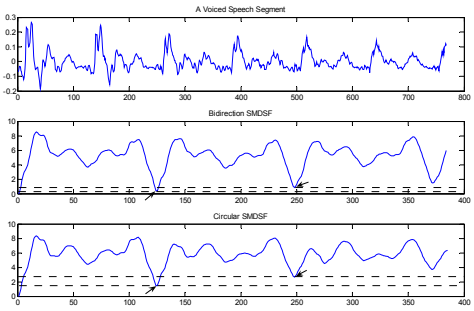

Figure 2: The difference between the bidirectional SMDSF and the circular SMDSF

2.3. Combined SMDSF

As can be seen above, the bidirectional SMDSF and the circular SMDSF have their own characteristics, which are different in the doubling error rate and the halving error rate. It is straightforward to propose that the two functions can complement each other in some sense when they are combined together.

A combined SMDSF is defined as a linear interpolation between the bidirectional SMDSF and the circular SMDSF by the simple form

$$D''_t(\tau) = \alpha D_t(\tau) + (1-\alpha) D'_t(\tau) \quad (15)$$

where $\alpha \in [0, 1]$ is an interpolation parameter.

Table 2 gives the GERs (refer to Section 4 for the definition of GER and the details of the experiment conditions) of pitch estimation using the bidirectional SMDSF, the circular SMDSF, and the combined SMDSF with α=0.3 without any pre-processing and post-processing. The experimental results show that the bidirectional SSDMF has higher doubling error rate and the circular SSDMF has higher halving error rate, while the combined SMDSF has the lowest GER and balanced doubling/halving error rates.

Table 2: GERs and doubling/halving error rates for pitch estimation using the bidirectional SMDSF, the circular SMDSF, and the combined SMDSF

Method	GER (%)	Doubling/Halving (%)
Bidirectional	6.6	4.9/1.7
Circular	4.5	1.9/2.6
Combined	**4.1**	**2.0/2.1**

3. Pitch tracking algorithm

In this section, we describe a pitch tracking algorithm based on the proposed combined SMDSF. The algorithm consists of two main parts: voice activity detection (VAD) and pitch estimation.

From a practical standpoint, we should consider the normalization of SMDSF. There are two methods to normalize the combined SMDSF used in this paper.

The first form of normalization was proposed by Cheveigne [2] and defined as follows

$$\tilde{f}(\tau) = \begin{cases} 1, & \tau = 0 \\ f(\tau)\tau \Big/ \sum_{k=1}^{\tau} f(k), & \tau = 1,..,N-1 \end{cases} \quad (16)$$

while the second one is herein proposed as

$$\tilde{f}(\tau) = f(\tau)N \Big/ \sum_{k=0}^{N-1} f(k), \tau = 0,1,..,N-1 \quad (17)$$

where $f(\cdot)$ means a type of SMDSF.

Experimental results show that the normalized combined SMDSF has more stable estimation values than the original function in voiced/unvoiced speech segmentation, using either Equation (16) or Equation (17). A simple VAD method is described as follows

$$\tilde{f}(\tau) \begin{cases} \le \beta, voiced \\ > \beta, unvoiced \end{cases} \quad (18)$$

where $\tilde{f}(\tau)$ is the normalized combined SMDSF and β is an absolute predefined threshold. The normalized function $\tilde{f}(\tau)$ can be considered as the measurement of the proportion of aperiodic power in certain signal [2]. Thus we can use Equation (18) to detect voiced segments of speech.

After the voice activity detection phase, we use Equation (12) to estimate pitch based on the normalized combined SMDSF. Note that the frame size N must be larger than the maximum value of the pitch period. The frame size used in the bidirectional SMDSF is N and is doubled in the circular SMDSF.

Our pitch tracking algorithm consists of the above two stages without any pre-processing or post-processing. Its time complexity is $O(MN\log_2(N))$ where M is the total number of frames and N is the frame size. The experimental results can show that this proposed algorithm has high performance at a moderate frame rate.

4. Evaluation

4.1. Experiment setup

The algorithm was evaluated on a small database of speech collected at the University of Edinburgh [1]. The Edinburgh database contains the speech of 100 sentences read by one male speaker and one female speaker. The database also contains reference pitch contours derived from simultaneously recorded laryngograph waveforms. The sentences in the database are biased to contain difficult cases for pitch estimation, such as voiced fricatives, nasals, liquids, and glides.

The formal evaluation was made by accumulating errors over all utterances in the database, using the reference pitch contours as ground truth. Note that our estimated pitch contours were not pre-processed (such as by low pass filtering) or post-processed (such as by a smoothing procedure like median filtering or dynamic programming).

4.2. Experiments and results

Comparisons between the estimated and reference pitch values were made every 12ms. The voicing decision error rate (VDER) was only computed for the fraction of voiced speech misclassified as unvoiced. Additionally, for the fraction of speech correctly identified as voiced, a GER was computed measuring the percentage of comparisons for which the reference and estimated pitch differed by more than 20%.

Note that the GERs in Tables 1 and 2 were computed between all voiced speech reference pitch values and corresponding estimated pitch values. The results of our evaluation are presented in Table 3.

Table 3: VDERs, GERs and doubling/halving error rates for pitch estimation using *Praat* (To Pitch) [9], the first form of normalized combined SMDSF *C* (as defined by Equation 16) and the second form of normalized combined SMDSF *L* (as defined by Equation 17) with β=0.60, 0.65, 0.70

Method		VDER (%)	GER (%)	Doubling/ Halving (%)
Praat: To Pitch		7.2	2.2	1.5/0.7
β=0.60	*C*	8.1	1.8	0.8/1.0
	L	8.4	1.8	0.8/1.0
β=0.65	*C*	**5.8**	**2.2**	1.0/1.2
	L	**6.0**	**2.1**	0.9/1.2
β=0.70	*C*	3.9	2.6	1.1/1.5
	L	4.2	2.5	1.1/1.4

The baseline is the pitch tracking algorithm integrated in *Praat* (Version 4.3.04) [9] which is well-known practical pitch tracking algorithm. The script used in *Praat* is "To Pitch... 0.012 40 500", which means frame shift is 0.012 second and the minimum and maximum values of f_0 are 40Hz and 500Hz. In our algorithm, the frame size used in the bidirectional SMDSF is 25ms and in the circular SMDSF 50ms. The minimum and maximum values of pitch are 2ms and 25ms. The results listed in Table 3 show that two types of normalization function have almost the same error rates at different β values. When $\beta \leq 0.65$, the GERs of our algorithm are lower than *Praat*'s. Larger β value will result in lower GER but higher VDER.

5. Conclusions

In this paper, we proposed two different types of SMDSF: a bidirectional SMDSF and a circular SMDSF. An efficient method to calculate these SMDSFs was also proposed with a small time complexity of $O(N\log_2(N))$. This is very important for real time applications.

The motivation of this paper was to analyze the error characteristics of the two different types of SMDSFs and construct an optimal function that was more accurate for use in pitch tracking. Experimental results show that pitch estimation based on the proposed combined SMDSF achieves the lowest error rate. Especially, it can achieve balanced doubling error rates and halving error rates.

We also described a relatively accurate and efficient pitch tracking algorithm based on the normalized combined SMDSF. Experimental results show that our pitch tracker has lower GER and DER than the baseline *Praat* method when β=0.65. However, the VAD method used in our algorithm was too simple. A more complicated VAD method and pre-/post-processing steps will be considered in our future research.

6. References

[1] P. C. Bagshaw, S. M. Hiller, M. A. Jack. Enhanced pitch tracking and the processing of f0 contours for computer aided intonation teaching. *In Proceedings of the 3rd European Conference on Speech Communication and Technology, volume2:1003-1006, 1993.*

[2] A. D. Cheveigne, H. Kawahara. Yin, a Fundamental Frequency Estimator for Speech and Music. *Journal of the Acoustical Society of America, 111(4):1917-1930, 2002.*

[3] P. Boersma. Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. *Proc. Institute of Phonetic Sciences, 1(17):97-110, 1993.*

[4] M. Ross, H. Shaffer, A. Cohen, et al. Average magnitude difference function pitch extractor. *IEEE Trans on Acoustics, Speech, and Signal Processing, 1974, 22(5):353-362.*

[5] W. Zhang, G. Xu, Y. Wang. Pitch estimation based on circular AMDF. *Proc. of ICASSP, I:341-344, 2002.*

[6] B. Secrest, G. Doddington. An integrated pitch tracking algorithm for speech systems. *Proc. of ICASSP, 1352-1355, 1983.*

[7] S. Sood, A. Krishnamurthy. A robust On-The-Fly Pitch (OTFP) estimation algorithm. *Proc. of ACM Multimedia, 280-283, 2004.*

[8] D. Gerhard. Pitch Extraction and Fundamental Frequency: History and Current Techniques. *Technical Report TR-CS 2003-06, University of Regina Department of Computer Science, 2003*

[9] http://www.fon.hum.uva.nl/praat/praatcon4304_win.zip