A High Speed Pitch Extraction Method Based on Peak Detection and AMDF
(Peak 검출과 AMDF에 의한 고속도 음성주기 추출방법)

성 원 용*, 은 종 관**
(Sung, Won Yong and Un, Chong Kwan)

要 約

본 논문에서는 peak 검출과 average magnitude difference function (AMDF)방법을 이용해서 음성의 주기를 고속도로 추출하는 방법이 연구되었다. 먼저 입력 음성을 800Hz로 대역폭을 줄인다음 pitch peak가 될 만한 몇개의 peak을 검출한다. 그 다음 이들 peak들의 값을 갖고 AMDF를 계산해서 이들 값들 중에서 최소의 AMDF 치를 갖는 peak를 원하는 음성주기로 결정을 한다. 이 방법을 사용하여 음성의 주기를 검출하면 타 음성주기 추출방법 보다 훨씬 적은 계산 시간이 소요될 뿐만 아니라 비교적 정확한 결과를 얻을 수 있다.

Abstract

We present a high speed pitch estimation algorithm that is based on peak detection and average magnitude difference function (AMDF). A few pitch candidates are first estimated from the low-pass filtered (800 Hz) speech by a peak detection algorithm. AMDF values of the pitch candidates are then calculated, and the pitch candidate that yields the minimum AMDF value is chosen as the desired pitch period. The new method requires far less computation time than other pitch estimation algorithms, while it yields fairly accurate results.

I. Introduction

It is well known that speech waveform is produced by exciting vocal tract with an excitation that exhibits a noise-like or periodic impulse character. For voiced speech the excitation signal is a quasi-stationary impulses with the fundamental frequency range of 50 to 450 Hz, and for unvoiced speech the excitation signal is essentially random noise.

Pitch extraction is required in a variety of vocoding and speech processing systems. As a result, many different pitch extraction algorithms have been developed. Most of these algorithms require a large number of computations, and thus are difficult to have hardware implementation for real time operation without using a specially designed high speed processor. Accordingly, development of a simple pitch extraction algorithm that yields accurate results has been of interest to many speech researchers.

The pitch extraction algorithms so far developed may be categorized in three different types according to their basic properties: the autocorrelation type, such as those that use

* 準會員, 금성사 중앙연구소
(Gold Star Co., Central Research Laboratory)
** 正會員, 한국과학원 전기 및 전자공학과
(Dept. of Electrical Science, KAIS)
接受日字 : 1980年 4月 1日

autocorrelation function or average magnitude difference function (AMDF) applied to low-pass filtered speech or inverse filtered speech residual signal[1-3]; the time domain pitch extraction type, such as the parallel processing algorithm[4] and the data reduction method[5]; and the spectral analysis type, such as the cepstrum method[6] and the harmonic pitch detector[7]

In this paper, we present a pitch extraction algorithm that is much simpler than the above mentioned approaches. Pitch period is obtained from the result of peak detection and AMDF. Positive and negative peaks are first selected from the 800 Hz low-pass filtered input speech. A few pitch candidate values are selected from the result of peak detection by using a "blanking and run-down" method that is similar to the one used by Gold and Rabiner[4] Then, AMDF is calculated based on these candidate values. Since AMDF needs not be computed at every sampling point, this approach yields significant computational saving. As for voiced/unvoiced decision, it is done based on the information available from the procedure of pitch period estimation.

Following this introduction, we describe the new pitch extraction algorithm in detail in Section II. In Section III, we present and discuss the computer simulation results. In Section IV, we consider hardware implementation of the system. Finally, conclusion is made in Section V.

II. The High Speed Pitch Extraction Algorithm

A. General

The overall pitch extraction system is shown in Fig. 1. The input speech is first band-pass filtered (100-800 Hz) to remove high frequency formants. This signal is then fed to a peak detector to obtain major peaks of speech in two analysis blocks (typically 40 msec). Various distances between these peaks become the candidates of the desired pitch period. To determine the true pitch period among the candidate values, AMDF is calculated based on the peak detection results. Normally, in an autocorrelation or AMDF type pitch extractor, correlation is calculated at every sampling instant over at least one analysis block, and peaks are detected for pitch period estimation. However, in our approach pitch candidates are determined first. Therefore, one needs not compute correlation at each sampling point. Rather, AMDF is computed at each location of the pitch candidate, and the largest AMDF null is chosen as the true pitch. The distance between the origin and the largest AMDF null is decided as the true pitch period by the pitch period estimator.

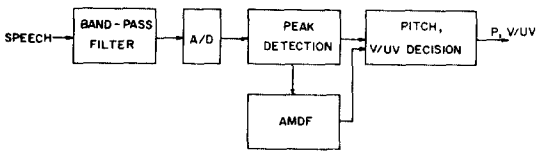

Fig. 1. Block diagram of the pitch extractor.

Table 1. Durations (ms) of step down operation for different ranges of past average pitch period $\bar{P}_n$.

$\bar{P}_n$	D1	D2	D3	D4
2.5 – 4	2.5	0	1.5	16
4 – 7	2.5	1.5	3	13
7 – 12	3	4	5	8
12 – 20	5	7	8	0

Voiced/unvoiced (V/UV) decision is made based on the number of zero crossings, the input speech magnitude, and the null depth of AMDF. All these informations are available from the procedure of pitch period estimation. The decision procedure is similar to the one used by Un and Yang[3].

Finally, the outputs of the pitch period extimation and the V/UV discriminator are fed to a smoother to correct any gross error. We now describe major subsystems in detail.

B. Peak Detector

The peak detector detects positive and

negative dominant peaks (three each) of input speech in two analysis frames. Since there are many spurious peaks, these must be suppressed. For this purpose we use a "blanking and run-down" method that is similar to the one used by Gold and Rabiner[4]. In this method, once a major peak is detected, there is a blanking interval followed by a run-down period. An illustration for a typical blanking and run-down operation for peak detection is shown in Fig. 2. In this figure the solid bars on the time (t)

Fig. 2. Operation of blanking (solid line) and rundown (dotted line) for peak detection.

axis are the peaks preliminarily detected. If a peak exceeds twice the level of the blanking region or the level of the run-down region, it is detected as a major peak, and the operation is restarted. The time lengths of the blanking and run-down regions are proportional to the average pitch period $\bar{P}_n$ determined by

$$\bar{P}_n = \frac{2}{3}\bar{P}_{n-1} + \frac{1}{3}P_{n-1}, \qquad (1)$$

where $\bar{P}_{n-1}$ and P_{n-1} are the average pitch period of (n-1)th frame and the pitch period of (n-1)th frame, respectively. The blanking time τ is given by 0.4 $\bar{P}_n$, and the time constant β of exponential decay in the run-down region is $\bar{P}_n$. Distances between the detected peaks are then measured. These distances become the candidates of the desired pitch period. In addition, we include as another candidate the pitch period of the previous voiced block to prevent any gross error that occurs occasionally at the trailing edge of voiced sound because of peak detection failure. Accordingly, a total of seven candidate pitch periods are chosen. One is selected among these as the true pitch period based on the result of AMDF computation. computation.

C. Computation of AMDF and Determination of Pitch Period

Once candidate pitch periods are determined, we must choose one for the true pitch period. For this purpose we compute AMDF as

$$AMDF(k) = \frac{1}{R}\sum_{i=1}^{N} |s_i - s_{i+k}|,$$

$$k = P_1, P_2, \ldots, P_7 \qquad (2)$$

$$\text{where } R = \sum_{j=1}^{N} |s_j|, \qquad (3)$$

s_i is an input speech sample, $P_1, P_2, \ldots, P_7$ are candidate pitch periods, and N is the number of samples in one analysis block. Note that, in conventional AMDF computation, k ranges from one to N (typically N = 160), and thus N AMDF values must be calculated. However, in our case we need to calculate only seven AMDF values at the locations of detected peaks. Accordingly, computations can be significantly reduced.

The process of selecting a pitch period involves setting two threshold values, TH1 and TH2, as shown in Fig. 3.[1] The pitch period

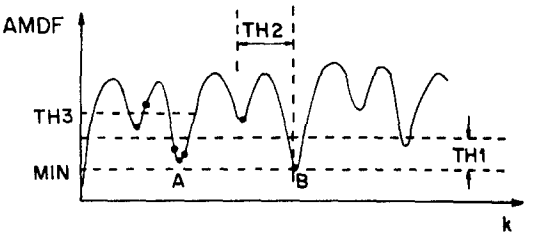

Fig. 3. Null picking of AMDF.

candidate with the minimum AMDF value can be a true pitch period or its multiple. We choose as a pitch period the left most candidate (A in Fig. 3) with the value of local minimum AMDF that falls in the region of MIN and MIN+TH1 and is at least TH2 points apart from the pitch period candidate (B in Fig. 3) having the minimum AMDF value.

The minimum AMDF value is a significant parameter for V/UV decision, because it shows

the periodicity of a speech segment. We set ICLEAR = 1 when the minimum AMDF value is below TH3.[1] Otherwise, we set ICLEAR=0, and the pitch period of the previous voiced block is used for a pitch period if this block is decided voiced.

D. V/UV Decision

The reliability of V/UV decision is important, because a V/UV decision error gives more noticeable degradation of synthetic speech than a pitch error in a vocoder. In this method, V/UV decision is done based on the average magnitude of input signal samples (R of Eq. 3) and the minimum AMDF values. This information is available from the procedure of pitch period estimation. Our V/UV decision procedure is shown in Fig. 4. ICLEAR and V/UV decision result of the previous frame are represented by IPCL and IPVUV, respectively. E1, E2, and E3 are the threshold values of the average magnitude of input signal. They are given by

$$E1 = T1 \cdot RAV \quad (4)$$

$$E2 = T2 \cdot RAV \quad (5)$$

$$E3 = T3 \cdot RAV \quad (6)$$

with $RAV_{new} = 4/5\ RAV + 1/5\ R$ (7)

and $0 < T1 < T2 < T3 < 1$.[2] Here RAV is the moving average of R(Eq. 2). RAV is updated only in the voiced block.

In Fig. 4, there are seven logic paths for a V/UV decision. On path A, if the previous block is voiced, a strong periodicity (ICLEAR = 1) indicates that the present block is a part of the voiced speech. On paths B and C, if the previous block is unvoiced (IPVUV = 0), a strong preiodicity in the present block indicates that the present block is an onset of voiced speech. In this case, a small average magnitude value indicates that only a small portion of the block is voiced. Such a block is decided unvoiced if the average magnitude value is less than a threshold value E2. On paths D and E, an unvoiced decision is made for the present block when the previous and present blocks are both unvoiced and/or weakly periodic. On

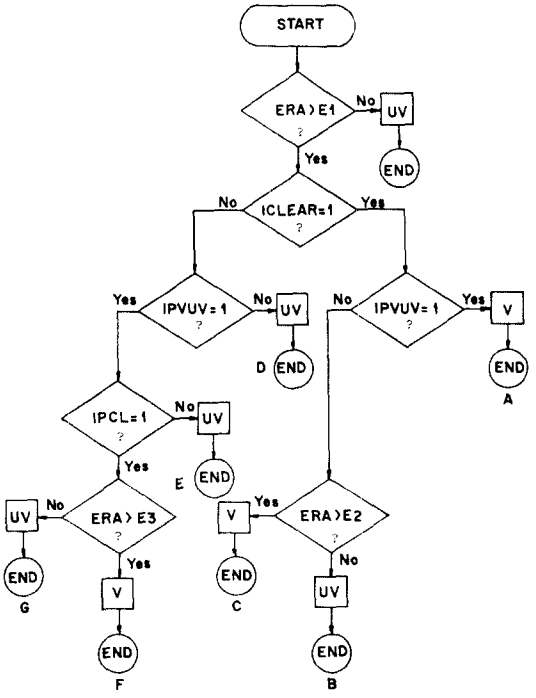

Fig. 4. Flow chart of V/UV decision logic.

paths F and G, we encounter a situation in which the previous block has been classified as voiced because of its high periodicity, while the present block shows weak periodicity. This situation occurs when the present block corresponds to a trailing edge of the voiced speech. A threshold value E3 is used to decide whether a large portion of quasi-periodic voicing exists in the present block.

For any block, if a voiced decision has been made, the V/UV decision logic will output the pitch period P measured previously. On the other hand, if an unvoiced decision has been made, a zero instead of P is output by the V/UV decision logic so that no additional information bit is required to indicate the V/UV decision.

E. Smoothing

Pitch estimations including V/UV decision are smoothed by a smoothing algorithm. The main purpose of smoothing is to eliminate occasional gross errors, such as pitch doubling

or halving, and V/UV decision errors. It was observed that without smoothing gross errors occurred occasionally. However, as a result of smoothing few gross pitch errors have been observed in our extensive testing of male and female speech.

In implementing the smoothing algorithm, it is unavoidable to allow some delay time. The present smoothing algorithm requires one frame of delay time. The algorithm is largely the same as that of Un and Yang[3]. It has been designed to use the running median of pitch period, $\bar{P}_n$ (Eq. 1). The flowchart of pitch smoothing is shown in Fig. 5. In this

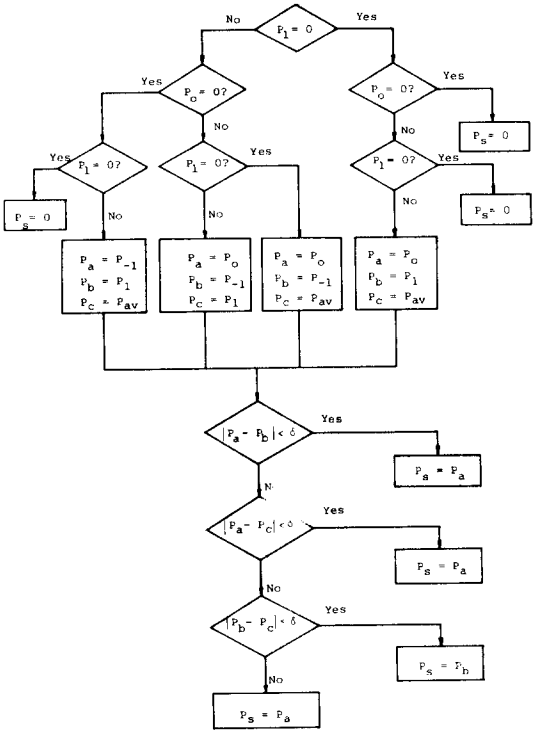

Fig. 5. Flow chart of smoothing algorithm. (Note: For simplicity we use the notations $P_n=P_o$, $P_{n-1}=P_{-1}$ and $P_{n+1}=P_1$).

flowchart, P_n, P_{n-1}, and P_{n+1}, are the pitch estimations of the current, previous, and next frame, respectively. P_c is the result of smoothing algorithm, and this becomes the final pitch estimation for the current block. δ is the threshold value for comparison of the two periods, and is given by

$$\delta = 0.25\bar{P}_n. \tag{8}$$

The smoothing algorithm does the following:

1) If the nth frame is voiced, but the (n-1)th and (n+1)th frames are unvoiced, the nth frame is changed to unvoiced. Similarly, if the nth frame is unvoiced, but the (n-1)th and (n+1)th frames are voiced, the nth frame is changed to voiced and P_c is $(P_{n-1} + P_{n+1})/2$.
2) If P_{n-1}, P_n, and P_{n+1} are all greater than zero, P_n is compared with P_{n-1} and P_{n+1}. If the comparisons exceed a threshold, and P_{n-1} and P_{n+1} are close to each other, P_{n-1} becomes P_c; otherwise, P_n becomes P_c.
3) If P_{n-1} is zero, and P_n and P_{n+1} are greater than zero, P_n is compared with P_{n+1} and $\bar{P}_n$. If the comparisons exceed a threshold and P_{n+1} and $\bar{P}_n$ are close to each other, P_{n+1} becomes P_c; otherwise, P_n becomes P_c.
4) If P_{n+1} is zero, and P_n and P_{n-1} are greater than zero, P_n is compared with P_{n-1} and $\bar{P}_n$. If the comparisons exceed a threshold and P_{n-1} and $\bar{P}_n$ are close to each other, P_{n-1} becomes P_c; otherwise, P_n becomes P_c.
5) If P_n is zero, and P_{n-1} and P_{n+1} are greater than zero, P_{n-1} is compared with P_{n+1} and $\bar{P}_n$. If the comparisons exceed a threshold and P_{n+1} and P_n are close to each other, P_{n+1} becomes P_c; otherwise, P_{n-1} becomes P_c.

III. Computer Simulation and Discussion

The pitch extraction algorithm described has been simulated on a computer, and its performance has been tested extensively with male and female speech samples in different environments. The performance of this algorithm has been compared with the eye detected results and also with the results obtained by the pitch extraction algorithm of Un and

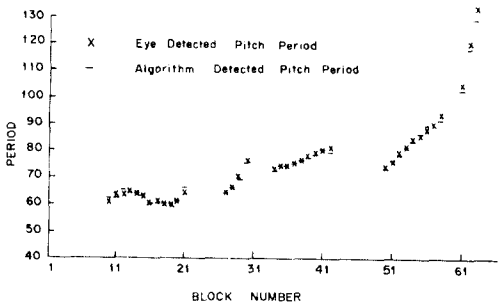

Fig. 6. Comparison of pitch period contours obtained by proposed method and eye detection.

Yang[3]. Fig. 6 shows a comparison of pitch period contour obtained by the present extractor and by eye detection on an interactive display without the smoothing algorithm. It has been observed that the algorithm yields reasonably accurate results. There is no discernible degradation in the quality of synthesized speech by this algorithm compared with that obtained by the eye detection method or the pitch extraction algorithm of Un and Yang.

The present algorithm has some distinct advantages. It can be used for synchronous pitch detection by a slight modification because peaks are detected by a peak detector. Also, it is robust to pitch halving or pitch doubling. Most important of all, computational requirement is very modest compared with the pitch extraction algorithm based on LPC inverse filtering and AMDF or the parallel processing method. This aspect will be discussed later.

The weak point of this method would be that errors can originate from the pattern class minimization procedure by peak detection. It is evident that pitch extraction is not possible if peak detection fails. But this possibility is rather small because the number of pitch candidates is always sufficient.

The present algorithm has been tested with noisy and high-pass filtered speech samples. In this case one can expect a large number of errors because it is difficult to detect peaks from noisy speech. However, only a small number of errors have been observed even in speech with SNR of 10 dB. When it was tested with high-pass filtered speech (250 Hz), pitch halving occurred occasionally. Nevertheless, this is not a serious problem since it can be corrected by the smoothing algorithm.

The threshold values, TH1, TH2, TH3, used for AMDF are 0.1, 11, and 0.9, respectively. Also, the values, T1, T2, and T3, used for V/UV decision are 0.15, 0.3, and 0.35, respectively. These values that give optimum results have been determined experimentally. If the input speech is very noisy, the threshold values for V/UV decision must be increased.

IV. Hardware Implementation Consideration

It would be desirable to implement this algorithm using an inexpensive MOS microprocessor without using an external processing unit. For this purpose the algorithm may be modified slightly to eliminate multiplication operations in the peak detector. That is, the exponential run-down circuit in the peak detector is modified to step down circuit as shown in Fig. 7. Typical durations for each

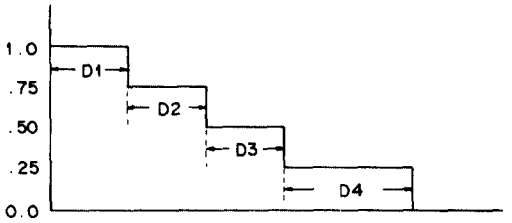

Fig. 7. Step down operation in detection of pitch candidates.

step, D1, D2, D3, and D4 are given in Table 1 according to the past averaged pitch period $\overline{P}_n$ (Eq. 1). D1 is the duration of a blanking region, and D2, D3, and D4 are durations of the step down stages.

AMDF calculation can also be modified to reduce the computation time significantly. The modified AMDF is given by

$$AMDF(K) = \frac{1}{R} \sum_{i=1}^{N/4} |s_{4i} - s_{4i+k}|, \quad (9)$$

$k = P_1, \ldots, P_7$

$$R = \sum_{j=1}^{N/4} |s_{4j}| \quad (10)$$

$$R1 = 4 \cdot R. \quad (11)$$

This modification should not be confused with down sampling as used in the SIFT pitch extraction method[8]. According to our computer simulation results, degradation by this simplification is negligible. This modification is possible because pitch is a long term information of speech and the input signal is low-pass filtered. This modification in AMDF calculation was used previously in hardware implementation of the AMDF pitch extraction method.

Computation time of the proposed algorithm may be calculated using the instruction set of the most popular 8 bit MOS microprocessor such as Z 80[9] which operates with 4 MHz clock. If the low-pass filtered speech is sampled at 8 KHz, the computation time for peak detection is within 4 msec, and AMDF is within 7 msec per 20 msec frame. Therefore, real-time hardware implementation is possible using an inexpensive MOS microprocessor.

V. Conclusion

A new pitch extraction algorithm that is based on peak detection and AMDF has been described. A few pitch candidate values are first selected from positive and negative peak detectors, and then AMDF is calculated based on these candidate valuves. As a result, computation time for AMDF calculation is much reduced and real time operation with an inexpensive MOS microprocessor is possible. Computer simulation and testing of this algorithm in different environments yields good results. Hardware implementation of this algorithm using a MOS microprocessor has been considered. A method of modification of the algorithm for improving the computational speed has also been discussed.

References

1. L.R. Rabiner, "On the Use of Autocorrelation Analysis for Pitch Detection," IEEE Trans. on ASSP, Vol. ASSP-25, pp. 24-33, Feb. 1977.
2. M.J. Ross, et al., "Average Magnitude Difference Function Pitch Extractor," IEEE Trans. on ASSP. Vol. ASSP-22, No. 5, Oct. 1974.
3. C.K. Un and S.C. Yang, "A Pitch Extraction Algorithm Based on LPC Inverse Filtering and AMDF," IEEE Trans. on ASSP, Vol. ASSP-25, No. 6, pp. 565-572, Nov. 1977.
4. B. Gold and L.R. Rabiner, "Parallel Processing Technique for Estimating Pitch Periods of Speech in the Time Domain," J. Acoust. Soc. Amer., Vol. 46, pp. 442-448, Aug. 1969.
5. N.J. Miller, "Pitch Detection by Data Reduction," IEEE Trans. on ASSP, Vol. ASSP-23, No. 1, Feb. 1975.
6. A.M. Noll, "Cepstrum Pitch Determination," J. Acoust. Soc. Amer., Vol. 41, pp. 293-309, Feb. 1967.
7. S. Seneff, "Real Time Harmonic Pitch Detector," IEEE Trans. on ASSP, Vol. ASSP-26, pp. 358-365, Aug. 1978.
8. J.D. Markel, "The SIFT Algorithm for Fundamental Frequency Estimation," IEEE Trans. Audio and Electroacous., Vol. AU-20, No. 5, Dec. 1972.
9. Zilog, Inc., "Z 80-CPU Product Specification," Mar. 1978.

[1] Those values for TH1, TH2 and TH3 are determined by computer simulation. See Section III for typical values of those.

[2] Again, these values are determined by computer simulation. See Section III for typical values.