A RELIABLE POSTPROCESSOR FOR PITCH DETERMINATION ALGORITHMS

GAO YANG**,*, H. LEICH*, R. BOITE*

*Lab. T.C.T.S., Faculté Polytechnique de Mons, Belgium
**Lernout & Hauspie Speechproducts n.v.,Wemmel, Belgium

RÉSUMÉ

Un problème difficile pour un PDA (Pitch Determination Algorithm) de parole consiste à éviter les grosses erreurs tels que le demi-pitch et le multiple pitch. Cet article propose un nouveau postprocesseur qui peut fiablement et efficacement corriger les grosses erreurs. L'idée principale du postprocesseur se base sur l'utilisation de l'histogramme des dernières valeurs de l'estimation du pitch. Il peut être guaranti que la correcte valeur du pitch est située dans une petite region centrée sur la position du sommet de l'histogramme. Une valeur référence calculée dans la petite region et parfois modifiée avec les valeurs progressives en dehors de la petite region est utilisée pour verifier et corriger l'estimation du pitch actuel. Les résultats obtenus par des tests sur la base de donnée au laboratoire ont démonstré que le PDA avec le postprocesseur proposé est tellement fiable qu' aucune grosse erreur a été découverte.

ABSTRACT

A difficult problem for a PDA (Pitch Determination Algorithm) of speech signal is to avoid gross errors such as halving pitch and multiple pitch. This paper proposes a new postprocessor which can reliablely and efficiently correct the gross errors. The basic principle of the postprocessor is based on utilizing the histogram of the past estimated values of pitch. It can be guaranteed that the correct value of pitch is situated in a small region centring around the peak position of the histogram. A reference value calculated in the small region and sometimes modified with the progressive values outside the small region is used to verify and correct the present estimated pitch. The results obtained by testing the data base in the laboratory showed that the PDA with the proposed postprocessor is so reliable that no gross error was found.

1. INTRODUCTION

There are a large number of methods [1] to estimate the pitch period from original speech signal. As described in [1], a PDA (Pitch Determination Algorithm) can be subdivided into three processing steps: (1) the preprocessor, (2) the basic extractor, and (3) the postprocessor. The task of the preprocessor is data reduction and enhancement in order to facilitate the operation of the basic extractor. The preprocessor is not always necessary, depending on the used algorithm. The basic extractor performs the conversion from the input signal into a series of pitch estimates. The performance of the basic extractor will also influence the quality of the postprocessor whose typical tasks are to correct the gross errors.

Until now none of the existing PDAs operated without errors, even under good recording conditions [1]. Each PDA had its own "favorite" error. Almost any gross error is perceptible.

In order to reduce the gross errors, an efficient PDA has been proposed in [2]. With this PDA the probability of the gross errors such as halving pitch and multiple pitch has been reduced to a very small value which is appropriate for usual applications. In this paper, a reliable postprocessor will be proposed and added to the PDA described in [2] to correct again the occasional errors.

2. AN EFFICIENT PDA

For convenience, the PDA presented in [2] is recalled in this section. In [2] it is used for a fast CELP vocoder in which the precise estimation of pitch is not so necessary because it may be modified in the Long Term Prediction (LTP) analysis process, but the gross errors should be as few as possible.

Let us suppose that the pitch of one segment of speech signal s(n) from n=0 to n=N-1 will be estimated with the sampling frequency equal to f_s=8 kHz. The semi-normalized squared correlation functions are defined as follows. The forward function is

$$R_f(P) = [\ \Theta_f(P)\ ]^2 / E_f(P) \qquad (1)$$

where $\Theta_f(P)$ is the forward correlation function:

$$\Theta_f(P) = \sum_{n=0}^{N-1} s(n)\, s(n+P) \qquad (2)$$

and $E_f(P)$ is the forward energy:

$$E_f(P) = \sum_{n=0}^{N-1} s^2(n+P) \qquad (3)$$

$\Theta_f(P)$ is set to zero if it is negative. Similarly the backward function is

$$R_b(P) = [\,\Theta_b(P)\,]^2 / E_b(P) \qquad (4)$$

where $\Theta_b(P)$ is the backward correlation function :

$$\Theta_b(P) = \sum_{n=0}^{N-1} s(n)\, s(n-P) \qquad (5)$$

and $E_b(P)$ is the backward energy:

$$E_b(P) = \sum_{n=0}^{N-1} s^2(n-P) \qquad (6)$$

$\Theta_b(P)$ is set to zero if it is negative. In the above definitions, the energy terms $E_f(P)$ and $E_b(P)$ are used to normalize the correlation functions and the calculation of the square root is avoided; the energy terms can be calculated by recursion. Suppose the maximal values of the forward and backward functions respectively with (1) and (4):

$$R_f^{max} = \max \{\, R_f(P), \text{ for } P=MIN\_P,...,MAX\_P \,\} \qquad (7)$$

$$R_b^{max} = \max \{\, R_b(P), \text{ for } P=MIN\_P,...,MAX\_P \,\} \qquad (8)$$

where {MIN_P,...,MAX_P} specifies the range of pitch. The initial estimation of pitch as the output of the basic extractor can be obtained by maximizing the following expression in the range of pitch P={MIN_P,...,MAX_P}:

$$\begin{aligned} R(P) &= R_f(P), && \text{if } R_b^{max} < C_0 \cdot R_f^{max} \\ &= R_b(P), && \text{if } R_f^{max} < C_0 \cdot R_b^{max} \\ &= R_f(P) + R_b(P), && \text{else} \end{aligned} \qquad (9)$$

where C_0 is a constant about 0.6. The expression (9) means that we use only the forward function or the backward function for the regions from unvoiced speech to voiced speech, or from voiced speech to unvoiced speech, otherwise the average value of both the forward and backward functions is taken. In general, R_b^{max} is smaller than R_f^{max} for the regions from unvoiced speech to voiced speech; on the contrary R_b^{max} is larger than R_f^{max} for the regions from voiced speech to unvoiced speech. With the output of this extractor, gross errors such as halving pitch are reduced almost to zero since no Hamming window is used in the expressions; so the correction only for errors such as multiple pitch will be much easier.

Suppose P which will be corrected is the optimal output of the basic extractor, C_1 is a constant about 0.75, C_2 is a constant about 1.25, C_3 is a constant about 0.65, and p is a candidate of the corrected pitch. In order to process speech signal in real time, the postprocessor is designed utilizing the pitch estimation (P_0) from the preceding frame:

(a) find the optimal candidate p in the specified range→
Maximize R(p), for $p=C_1 \cdot P/k, C_1 \cdot P/k+1,...,C_2 \cdot P/k$, k=2,3,4

(b) take the correction if two conditions are satisfied→
If $(C_1 \cdot k \cdot P_0 < P < C_2 \cdot k \cdot P_0)$, for k=2,3 or 4,
and $R(p) > C_3 \cdot R(P)$ for the optimal p,
then the correction: P = p

Further simplification of the PDA is possible. The input original signal or a spectrally flattened excitation signal can be lowpass-filtered with a cutoff frequency smaller than 2 kHz, and the sampling rate f_s (generally 8 kHz) is reduced to 4 kHz by a decimation process (2:1). Then, the decimated signal (the output of preprocessor) is processed by the way as the above description. The desired estimation of pitch will be twice the pitch estimation obtained by using the decimated signal. As will be mentioned in the next section another approach can also be used to simplify the PDA. The input original signal can be first cut to the values 1, 0, or -1 with a threshold and then processed. In this way the division operations are avoided.

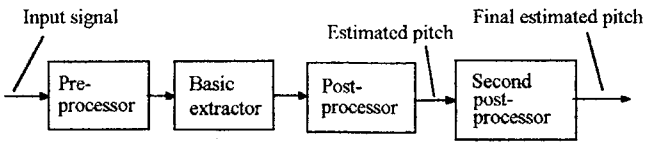
3. A RELIABLE POSTPROCESSOR

The correction of the gross errors by using the postprocessor given in the above section is based on the estimated pitch from the preceding frame. The preceding estimation is not always reliable as a reference to correct the present estimation although the correction decision does not depend on only this reference. If the preceding frame is unvoiced, the present estimation may not be efficiently corrected. In order to reliablely avoid the gross errors, in this section a new postprocessor will be proposed and added to the PDA to correct again the occasional errors. For convenience, this additional postprocessor is here called the "second postprocessor" which in fact can be considered as one part of the postprocessor. Fig. 1 gives the schema for the second postprocessor.

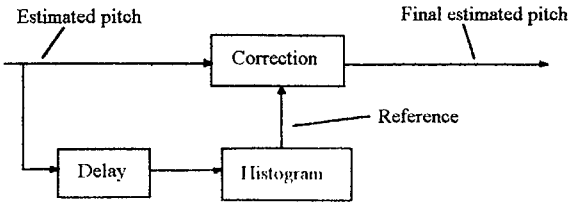
The reliable correction of the gross errors needs a reliable reference value which is close to the correct pitch. Observing the histogram of the estimated pitchs (the output of the first postprocessor) and the original speech signal, we can find that the correct pitch is always situated in a small region centring around the peak position of the histogram. This is due to the fact that the speech pitch period changes slowly in a short interval and the number of the correct estimated pitchs is usually much greater than the number of the false estimated pitchs. That is to say, the peak position of the histogram can be utilized to verify and correct the present estimated pitch.

In order to process the speech signal in real time, a set of past estimated pitchs, PITCH_HIST[i], i=0,1,2,...,I-1, which are obtained from I past voiced frames, are used to calculate the histogram. The pitch values around the peak position of the histogram are used to evaluate the reference value: PITCH_REF. I is a constant about 50 for the frame shift length between 20 ms and 25 ms. The larger the index i is, the more the frame for estimating PITCH_HIST[i] approachs the present frame. So, PITCH_HIST[I-1]=P_0 is the preceding estimated pitch if the preceding frame is voiced. In this procedure it is not very serious to require the exactitude of the

voiced/unvoiced classifier so long as the peak position of the histogram is not sufficiently influenced.



(a) A PDA with a second postprocessor



(b) Second postprocessor

Fig. 1 The schema for the postprocessor

Suppose the peak position of the histogram is P_{opt} and WID equal to a constant about 18 specifies the width of a small region centering on P_{opt} The reference value PITCH_REF is first calculated in the small region by using a way of weighting average:

$$PITCH\_REF = \frac{\sum_{k=1}^{M} i_k \cdot PITCH\_HIST[i_k]}{\sum_{k=1}^{M} i_k} \quad (10)$$

where it is supposed that there exists M values PITCH_HIST[i_k], k=1,2,....M, situated in the small region. We have

$$P_{opt}\text{-}WID/2<PITCH\_HIST[i_k]<P_{opt}+WID/2,\ k=1,2,....M \quad (11)$$

and we make

$$0\le i_0<i_1<....<i_k<....<i_M<I \quad (12)$$

i_M records the maximal index of the past estimated pitchs situated in the small region. In the process for calculating the weighting average of the estimated pitchs in the small region, the larger the index i_k is, the more the weighting coefficient is important. This is reasonable because the more recent estimated pitch is more significant for the reference to correct the present estimated pitch. i_M indicates the index of the most recent estimated pitch used to calculate the reference value PITCH_REF.

In practice, there exists the case that the estimated pitchs PITCH_HIST[i] vary progressively with the increase of the index i and the progressive values PITCH_HIST[i] for $i>i_M$ (if $i_M<I-1$) are also very significant for the reference although they are outside the small region. These estimated pitchs can be used to improve the reference value. The following subroutine is to find the range in which the estimated pitchs PITCH_HIST[i] for $i_M<i\le I_p$ are considered progressive. I_p denotes the superior limit of the index for the progressive estimated pitchs and specifies the progressive range. In the subroutine DIF is a constant about 0.2. If the variation of the next estimated pitch is small, it is considered as a progressive value. The initial value of I_p being made negative is to give a condition: if $I_p>0$ after the operation of this subroutine, there exists the progressive estimated pitchs outside the small region, otherwise no progressive value was found.

```
Ip = -10;            /* Initiation */
if (iM<I-1)          /*if iM=I-1, there is no significance*/
  for (i=iM+1;i<I;i=i+1)
   {
   if (|PITCH_HIST[i]-PITCH_HIST[i-1]|
                <DIF*PITCH_HIST[i-1]) Ip=i;
                /* verify if the value is progressive */
   else break;
   }
```

The following expression calculates the weighting average (PIT_REF) of the progressive estimated pitchs outside the small region:

if ($I_p>0$),

$$PIT\_REF = \frac{\sum_{i=i_M+1}^{I_p} i \cdot PITCH\_HIST[i]}{\sum_{i=i_M+1}^{I_p} i} \quad (13)$$

Then, the reference value PITCH_REF obtained in the small region is improved by the value PIT_REF with a weighting coefficient W_C about 0.4.

if ($I_p>0$),

$$PITCH\_REF \Leftarrow W\_C \cdot PITCH\_REF + (1-W\_C) \cdot PIT\_REF \quad (14)$$

After having the reference value PITCH_REF, the present estimated pitch can be verified and corrected. If the present frame is considered as voiced, we denote VOISE=1, otherwise VOISE=0. Similarly for the preceding frame, if it is voiced, VOISEm=1, otherwise VOISEm=0. As defined in the above section, P_0 indicates the preceding estimated pitch and P the present estimated pitch which can be verified and corrected by the following subroutine. If the present estimated pitch is far from the reference value PITCH_REF or the preceding estimated pitch P_0 when the preceding frame is voiced, the present estimated pitch will be verified and corrected on the condition that the present frame is voiced. The possible candidate of pitch is limited in the neighbourhood of the reference value PITCH_REF. In this neighbourhood the optimal candidate of pitch (Pit) is found by maximizing the function R(·) defined in (9). If the maximum of the function R(·) in the neighbourhood of the reference value PITCH_REF

is greater than R(P) multiplied by a coefficient D6 (D6≈0.4), the correction is taken: P=Pit.

```
CORRECT_YES=0;
                /* Initiation, 0 means needless to correct */

if (P<D1*PITCH_REF or P>D2*PITCH_REF)
   CORRECT_YES=1;
   /* D1≈0.7, D2≈1.3; CORRECT_YES= 1: need to correct*/

if (VOISEm=1) and (P>P0+D3 or P<P0-D3)
   CORRECT_YES=1;
    /* D3≈15 for fs=8 kHz and frame shift≈20 ms */

if (VOISE=1 and CORRECT_YES=1)
  {
  P_lim1=D1*PITCH_REF;        * D1≈0.7 *
  P_lim2=D2*PITCH_REF;        /* D2≈1.3 *
  if (PITCH_REF - P_lim1<D4) P_lim1=PITCH_REF-D4;
                                  /* D4≈13 */
  if (PITCH_REF - P_lim1>D5) P_lim1=PITCH_REF-D5;
                                  /* D5≈22 */
  if (P_lim2 - PITCH_REF<D4) P_lim2=PITCH_REF+D4;
  if (P_lim2 - PITCH_REF>D5) P_lim2=PITCH_REF+D5;
  if (P_lim1<MIN_P) P_lim1=MIN_P;
  if (P_lim2>MAX_P) P_lim2=MAX_P;
  /*{P_lim1,P_lim2} specifies the range of the candidate */

   Vmax=0;          /* Initiation */
  for (p=P_lim1; p<=P_lim2; p=p+1)
        if (R(p)>Vmax)
        {
        Vmax=R(p);
        Pit=p;   /*Pit:the pitch candidate for the correction*/
        }

  if (Vmax>D6*R(P)) P=Pit;
   /* D6≈0.4, if Vmax is quite great,the correction is taken. */
  }
```

The main load of the PDA is due to the computation of the function R(p), p=MIN_P,....,MAX_P. With the postprocessor proposed in this section, the requirement to the performance of the function R(p) is relatively lower. It is required only that the real pitch period be situated in the neighbourhood of the peak position of the histogram for the estimated pitchs. This condition is not difficult to satisfy. For some DSP processors, one of the most efficient approachs to simplify the computation load is to first cut the input speech signal to the simple values 1, 0, and -1 with a threshold:

```
MEAN_positive=0;
     /*Initiation of the mean value of the positive signal*/
NUM_positive=0;
     /*Initiation of the number of the positive samples*/
MEAN_negative=0;
     /*Initiation of the mean value of the negative signal*/
NUM_negative=0;
     /*Initiation of the number of the negative samples*/
for (n=0; n<L_max; n=n+1) if (s(n)>0)
          {
          NUM_positive=NUM_positive+1;
          MEAN_positive=MEAN_positive+s(n);
          }
   else {
        NUM_negative=NUM_negative+1;
        MEAN_negative=MEAN_negative+s(n);
        }

MEAN_positive=MEAN_positive/NUM_positive;
                /*the mean value of the positive signal*/
MEAN_negative=MEAN_negative/NUM_negative;
                /*the mean value of the negative signal*/

for (n=0; n<L_max; n=n+1)
   if (s(n)>G*MEAN_positive) s(n)=1;        /*G≈1.2*/
   else if (s(n)<G*MEAN_negtive) s(n)=-1;
        else s(n)=0;
```

In the above subroutine, the mean values of the positive signal and the negative signal are respectively calculated for the speech signal s(n), n=0,1,....,L_max. The thresholds are determined by multiplying the mean values with a coefficient G about 1.2. When the signal exceeds the thresholds, it is set to 1 or -1, otherwise equal to 0. With such a simplified sequential signal, the computation of the function R(·) is much faster (about 5 times) on special DSP processor or on PC (the program is written in assembler) than the direct computation using the original signal. In this way the divisions difficult to implement on some DSP's are avoided.

4 EXPERIMENTAL RESULTS

The performance of the PDA with the proposed postprocessor has been tested by observing speech waveform and using the data base in the laboratory. The date base consists of the sentences of English, French and German, spoken by men and women. The results showed that the PDA is so reliable that no gross error was found.

CONCLUSION

This paper proposes a postprocessor which can reliablely correct the gross errors of estimated pitch from a PDA.

REFFERENCES

[1] WOLFGANG J. HESS, "Pitch and Voicing Determination", Advances in Speech Signal Processing, edited by Sadaoki Furui and M. Mohan Sondhi, 1992.

[2] Gao Yang, Zanellato G. and Leich H., "A fast CELP vocoder with efficient computation of the pitch", on EUSIPCO'92, S6.L-6, pp. 511-514.