YET ANOTHER ALGORITHM FOR PITCH TRACKING

Kavita Kasi and Stephen A. Zahorian

Department of Electrical and Computer Engineering
Old Dominion University
Norfolk, VA 23529, USA.

ABSTRACT

In this paper, we present a pitch detection algorithm that is extremely robust for both high quality and telephone speech. The kernel method for this algorithm is the "NCCF or Normalized Cross Correlation" reported by David Talkin [1]. Major innovations include: processing of the original acoustic signal and a nonlinearly processed version of the signal to partially restore very weak F0 components; intelligent peak picking to select multiple F0 candidates and assign merit factors; and, incorporation of highly robust pitch contours obtained from smoothed versions of low frequency portions of spectrograms. Dynamic programming is used to find the "best" pitch track among all the candidates, using both local and transition costs. We evaluated our algorithm using the Keele pitch extraction reference database as "ground truth" for both "high quality" and "telephone" speech. For both types of speech, the error rates obtained are lower than the lowest reported in the literature.

1. INTRODUCTION

Numerous studies show the importance of prosody for human speech recognition, but only a few automatic systems actually combine and use fundamental frequency (F0) or pitch as it commonly called. Combined with other acoustic features, prosody can be used to significantly increase the performance of automatic speech recognition (ASR) systems [2]. A big stumbling block remains the lack of robust algorithms for F0 tracking. F0 is especially important for ASR in tonal languages such as Mandarin speech, for which pitch patterns are phonemically important [5]. Other applications for accurate F0 tracking include devices for speech analysis, transmission, synthesis; speaker recognition; speech articulation training aids for the deaf ([4], [6]), and foreign language training.

An important consideration for any speech-processing algorithm is performance using telephone speech, due to the many applications of ASR in this domain [3]. However, since the fundamental frequency is often weak or missing for telephone speech, and the signal distorted and noisy and overall degraded in quality, pitch detection for telephone speech is especially difficult [3].

Although many pitch detection algorithms have been reported, using a variety of techniques and with varying degrees of accuracy (see [7], [8] for summary), robust, easy to integrate methods, are still problematic. Therefore we introduce Yet Another Algorithm for Pitch Tracking (YAAPT).

The "kernel" of YAAPT is based on the "Robust Algorithm for Pitch Tracking (*RAPT*)" as discussed in [1]. However both the signal processing and the tracking algorithms are very different. One of the key contributions is the extensive use of spectrographic information to guide the tracking. That is, gross errors in F0 tracking can often be identified, by overlaying pitch tracks with the low frequency part of a spectrogram. In this paper, we describe methods for extracting this spectrographic information, and combining it with pitch estimates from correlation methods, in order to create a robust overall pitch track. Another innovation is to separately compute pitch candidates from both the original speech signal, and a nonlinearly processed version of the signal, and then to find the "lowest cost" track from among the candidates using dynamic programming. In this paper, we give a detailed description of the new innovations and the formal evaluation results.

2. ALGORITHM

The entire F0 tracking algorithm can be divided into five main steps,

1. Preprocessing.
2. F0 candidate selection based on NCCF.
3. Candidate refinement based on spectral information (both local and global).
4. Candidate modifications based on plausibility and continuity constraints.
5. Final path determination using dynamic programming.

2.1 Pre-processing

The first step of preprocessing is to create two versions of the signal, the original and absolute value of the signal. Except for the use of the second signal, which is discussed later, the rest of the preprocessing is fairly conventional. This is, each signal is bandpass filtered (bandpass of 100 Hz to 900 Hz) and center clipped.

The motivation for the nonlinear operation (absolute value) is illustrated in Figure 1. In particular, for a certain telephone sentence, with the acoustic signal shown in the top panel, the low frequency spectrograms are shown in the middle panel (original signal) and bottom panel (absolute value signal). The two curves overlaid on each spectrogram are explained in detail later. This figure illustrates that F0 is much more prominent in the lower panel than the middle panel (also verified by comparison with TIMIT version of same sentence). Similar effects were noted for many other sample signals, some studio quality as well. The strategy adopted was to completely process the signals, compute multiple F0 candidates from each, and then find the 'single' best track as described later.

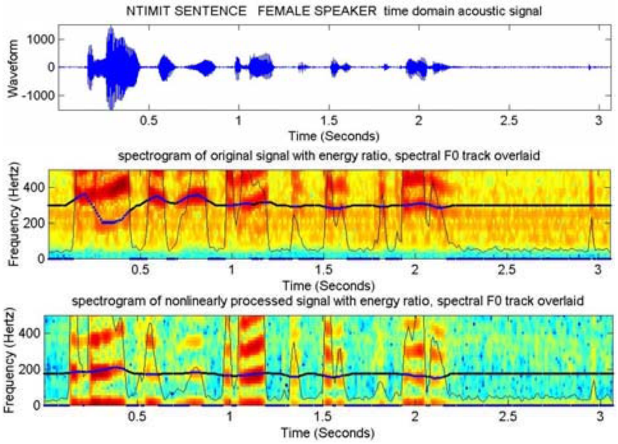

Figure 1: The first panel depicts a time domain NTIMIT sentence, the second panel depicts the spectrogram of the original signal with NLFER and spectral F0 track overlaid, and the third panel depicts the spectrogram of the nonlinearly processed signal with its NFLER and spectral F0 track overlaid.

2.2 F0 candidate estimation from the NCCF

The basic idea of correlation based F0-tracking is that the correlation signal will have a peak of large magnitude at a lag corresponding to the pitch period. If the magnitude of the largest peak is above some threshold (about 0.6), then the frame of speech is usually voiced. A modification to the basic autocorrelation, is the normalized cross correlation function (NCCF) defined as follows:

Given a frame of speech sampled, ***s(n), 0 ≤ n ≤ N-1***

Then:
$$NCCF(k) = \frac{\sum_{n=0}^{N-K} s(n)s(n+k)}{\sqrt{e_0 e_k}}$$

Where $e_k = \sum_{n=k}^{n=k+N-K} s^2(n)$, $\quad 0 \le k \le K\text{-}1 \qquad (1).$

As reported in [1], NCCF is better suited for pitch detection than the 'standard' autocorrelation function as the peaks are more prominent and less affected by the rapid variations in the signal amplitude. Nevertheless, it is still possible for the largest peak to occur at double or half the correct lag value or some other "incorrect" value thus giving rise to small and large errors. Thus, the additional processing described below is used.

Given the possibility of errors in the peaks of the NCCF, for the algorithm presented in this paper, we first attempt to find all large peaks in the NCCF, over some specified search range, and consider these as F0 candidates (typically, Max_cand=3). A final decision as to the actual F0 is made later, based on additional information sources and continuity considerations. In the remainder of this subsection, we describe how F0 candidates are chosen from the peaks and how merits are assigned to each selected peak.

The *intelligent peak-picking* algorithm is a multi-pass method as follows. In the first pass, a point in the NCCF is considered to be a peak, if it is larger than L points (L=2) on either side of the peak, and larger than a very low threshold (typically .3). Thus, it is possible (and likely) that a large number of peaks will be found in this first pass. In the second pass, any first pass peaks, which are closer (in lag) than 2 ms to even larger peaks are eliminated. All peaks remaining at this point are assigned a *merit* value equal to the magnitude of the NCCF at that lag value. In the third pass, each peak which also has a peak at twice the lag value has its merit increased by a small amount (.025), since the presence of the peak at the twice lag value is further evidence that the lower lag value peak is the 'correct' one.

The peaks still retained, with their associated merit values, are the F0 candidates considered for the final F0 track. In those cases where no peaks are found which satisfy the constraints just mentioned, the frame is considered to be unvoiced. Despite the relative robustness of the NCCF and the intelligent peak picking mentioned above, considerable experimental testing indicated that some errors still occurred in a final F0 track obtained from this information alone. Errors were most likely to occur for telephone speech or weakly voiced sections. Therefore, additional information and processing, as described in the next two sections, was used to improve the overall robustness of the algorithm.

2.3 Candidate refinement based on spectral information

For all signals examined, the patterns in the spectrogram appeared to clearly show voiced versus unvoiced regions of speech, and clearly showed the approximate F0 contour. Spectrograms were computed for both the original and nonlinearly processed versions of the signal, as mentioned above. In this section we describe the empirically determined methods used to help determine an additional measure to making voiced/unvoiced decisions, and methods used to determine a very smooth pitch contour with extremely few gross errors.

The normalized low-frequency energy ratio (NLFER) is computed to help indicate voiced versus unvoiced regions. The sum of absolute values of spectral samples (the average energy per frame) over the low frequency regions is taken, and then normalized by dividing by the average low frequency energy per frame over the utterance. In equation form NLFER is given by:

$$\text{NFLER} = \frac{\sum_i x(i,j)}{\frac{1}{N}\sum_i\sum_j x(i,j)} \quad \text{where N=total \# of frames, i=frequency index}$$

$$\text{x(i,j)= log magnitude of low frequency regions of spectrogram, } \textit{and} \text{ j=frame index}$$

In general, NLFER is high for voiced regions, and low for unvoiced regions, with a threshold value of approximately 0.65 as a good decision point. Note, however, that final voiced/unvoiced decisions did not use such a hard threshold. The smooth but robust pitch track is obtained using the following steps:

1. The low frequency portion of the spectrogram is smoothed using a mask approximately 60 Hz wide in frequency and 3 frames in time.

2. Simple peak picking is used to determine the first peak in the search range (F0_min: F0_max). If no peak is found, the frame is assumed to be unvoiced.
3. The track found from step 2 is median smoothed with a 3-point median filter.
4. An estimate of the average F0, and standard deviation of F0, is computed using the middle third (voiced frames are sorted in order of frequency) of the voiced frames from step 3.
5. All unvoiced frames in the estimate from step 3, and all those frames that differ significantly from the average F0, are replaced by the average F0 value.
6. The track from step 5 is again median smoothed with a 5-point median filter.
7. Simple heuristics are used to combine the two spectral F0 tracks to determine an overall smooth track.

For voiced frames in the speech, the F0 candidates from the NCCF are tested for "closeness" to the corresponding spectral F0 point. For candidates less than 1.5 F0_min away from the spectral F0 value, the merit is increased by a factor of 1.25, whereas peaks far away from the spectral F0 are reduced in merit according to distance.

3. THE TRACKING ALGORITHM

The end result of the processing steps mentioned above are the F0 candidate matrix, a merit matrix, an NLFER curve (from the original signal), and the spectrographic F0 track. These data are used to obtain local and transition cost matrices, from which the lowest cost pitch track thru all available candidates can be found using dynamic programming. Several processing steps, as outlined below, are used to compute the two cost matrices.

First, the frames are pre-classified according to the NFLER as either definitely unvoiced (NLFER < = .5) or probably voiced (NFLER > .5). Thus, for the definitely unvoiced region (region 1), all F0 candidates are set to 0, and the associated merit is to 0.99. For region 2, which could be either voiced or unvoiced, the basic idea is to make sure that every frame has at least 1 viable pitch estimate, and an unvoiced option, and that merits for both voiced and unvoiced options are assigned to roughly approximate probabilities. For each frame in region 2, the spectral F0 is also included as one candidate, with a merit set at a midpoint. The merit of the unvoiced candidate is set equal to [1 – (merit of best voiced candidate)].

After all of these merits are assigned, the local cost is computed as: $local\cos t = 1 - merit$, using a single matrix operation in MATLAB.

The main points in the computation of transition costs, are as follows ("i" is the present frame index in these equations):

1.For each pair of successive voiced candidates (i.e., non zero F0 candidates) $transition\ \cos t(i) \quad \propto \quad (F0(i) * F0(i-1))^2$
2.For each pair of successive candidates, only one of which is voiced, $transition\ \cos t(i) \quad \propto \quad NFLER(unvoiced\_frame)$.
3.For each pair of successive candidates, both of which are unvoiced, $transition\ \cos t(i) \quad \propto \quad NFLER(i) * NFLER(i-1)$.

The various proportionality constants mentioned above, plus one additional constant used to adjust the overall ratio of local to dynamic costs, were empirically determined based on an inspection of several hundred sample recordings. The overall algorithm is illustrated by the four panels in Figure 2.

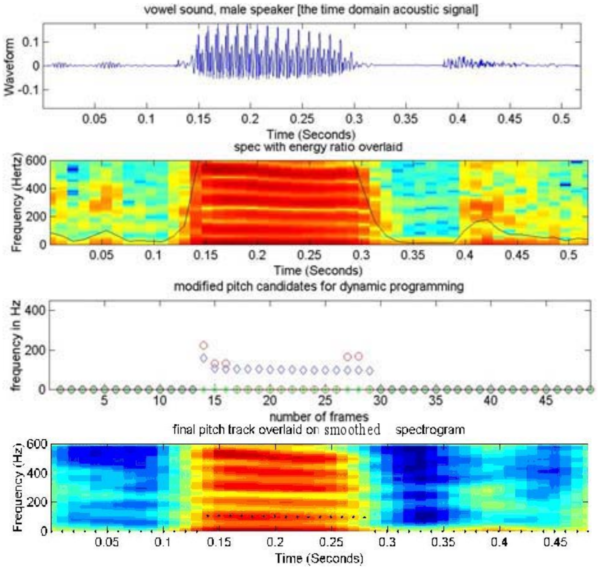

Figure 2: Illustration of the overall pitch-tracking algorithm.

All the algorithms mentioned in this paper were developed as a set of Matlab functions, which are intended to be easily integrated with other software. Key routines include one for computing multiple F0 candidates and merits for each frame, routines for spectral F0 tracking, and a routine for overall tracking. Another routine is under development to determine individual pitch period markers, given the results of the tracking described in this paper. As developed, the routines typically require about twice real time on a 500 MHz PC.

4. EXPERIMENTAL EVALUATION

We evaluated YAAPT using the Keele pitch extraction database [9]. This high quality speech (20 kHz sampling rate) contains 5 male and 5 female speakers, each speaking for about 35 seconds. The telephone version of this data, obtained from the spoken language systems group at MIT, was transmitted through a telephone channel and re-sampled at 8 kHz.

Although the Keele database includes what should be a very reliable control (which we call control$_1$), inspection of this track showed several instances of what appeared to be F0 halving. Therefore, we formed a second control (control$_2$) by simply setting all male pitch values below 70 Hz to 0, and all female pitch values below 110 Hz to 0. Fig. 3 depicts an example of control$_1$ and control$_2$ signals of a typical sentence by a female speaker. Note that the control$_1$ signal appears to have many instances of pitch having, whereas most of these appear to be eliminated in control$_2$.

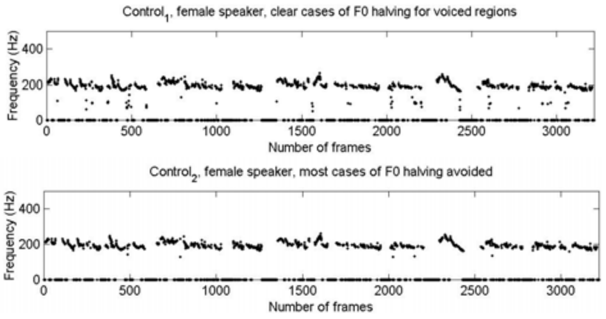

Figure 3: Illustration of F0 halving in control$_1$, mainly eliminated in control$_2$.

Of the many error measures that can be used to quantify F0 tracking accuracy, we used the following four measures to evaluate the tracking method reported in this paper:

1.Gross errors (G_err): This is computed as the percentage of frames such that the pitch estimate of the tracker deviates significantly (20%) from the pitch estimate of the reference. The measure is based only on those frames for which both the reference and tracker indicate voiced frames.

2.Voicing errors (V_err): This is the percentage of frames such that the tracker and the reference disagree in voicing decision.

3.Overall mean square error (NM^2): The overall normalized mean square error (NM^2) for the entire signal is given by:

$$NM^2 = \frac{\sum x(i) - y(i)]^2}{\sum [x(i)]^2} \qquad x(i) = reference\ track, y(i) = computed\ track,$$

$$i = index\ of\ \ present\ voiced\ \ frame$$

4. Voiced region mean square error (NMv^2): This error is computed the same as is NMv^2 except only those frames that are indicated as voiced for both the reference and computed are used in computation.

Results for YAAPT, using the four-error measures are reported below in Table 1. Only G_err error is reported for control$_2$; the other error measures were slightly lower for control$_2$ compared with control$_1$, but not dramatically so. Note that even if control$_2$ is not "correct," these results do indicate that many of the gross errors for control$_1$ are due to the very low F0 values in control$_1$. The only directly comparable results in the literature [3] report a gross error result of 4.25 % for high quality speech, and 4.34% for telephone speech.

Data	**Control$_1$**				**Control$_2$**
	V_err	NM	NMv^2	G_err	G_err
Studio	10.77	19.16	0.48	1.7	0.88
Tele	19.62	35.77	1.29	3.37	3.00

Tabel 1: Error analysis table showing the various error types.

5. SUMMARY

In this paper, a new pitch-tracking algorithm has been developed which combines multiple information sources to enable accurate robust F0 tracking. The multiple information sources include peaks selected from the normalized cross correlation of both the original and rectified signal and smoothed pitch tracks obtained from spectrograms. These multiple information sources are combined using experimentally determined heuristics and dynamic programming. An analysis of errors indicates better performance for both high quality and telephone speech than reported performance for any pitch tracking. The routines mentioned in paper are available from the second author of this paper as MATLAB functions**.**

6. ACKNOWLEDGEMENT

This work was partially supported by NSF grant BES-9977260. Special thanks to George Meyer (Keele University) for supplying the Keele database, and Chao Wang (MIT) for supplying the telephone version of the data.

REFERENCES

[1] D. Talkin, "A Robust Algorithm For Pitch Tracking", in Speech Coding and Synthesis, pp-495-518, 1995.

[2] M. Ostendorf and K. Ross, ``A Multi-Level Model for Recognition of Intonation Labels," in *Computing Prosody*, Y. Sagisaka, N. Campbell and N. Higuchi (Eds.), 291-308, Springer-Verlag, NY: 1997.

[3] Chao Wang and Stephanie Seneff, "Robust Pitch Tracking For Prosodic Modeling In Telephone Speech," ICASSP'00, Turkey.

[4] Pc Bagshaw, SM Miller and MA Jack, "Enhanced Pitch Tracking and the processing of the F0 contours for computer aided intonation teaching", Proc. EUROSPEECH'93, Berlin, pp. 1003-1006.

[5] Eric Chang, Jianlai Zhou, Shou Di, Chao Huang, Kai-Fu Lee, "Large Vocabulary Mandarin Speech Recognition with Different Approaches In Modeling Tones", ICSLP'00, Beijing.

[6] S. A. Zahorian, A. Zimmer, and B. Dai, "Personal Computer Software Vowel Training Aid for the Hearing Impaired", ICASSP'98, pp. VI-3625-3628, Seattle, Washington.

[7] Eric Mousset, William A. Ainsworth, Jose A.R. Fonollosa, "A comparision of several recent methods of fundamental frequency and voicing decision estimation", ICSLP'96, pp. 1273-1276,Philadelphia, .

[8] Rabiner Lawrence R., Cheng, Michael, J. Rosenberg, Aaron E. And McGonegal, Carol A., " A comparative performance Study of several pitch detection algorithms," in IEEE Transaction on Acoustics, Speech and Signal Processing, Vol. ASSP-24, 399-417,No-5,Oct'76.

[9] F.Plante, G.Meyer, and W. A. Ainsworth, "A pitch extraction reference database," in EUROSPEECH'95, Madrid, pp. 837-840.