A spectral/temporal method for robust fundamental frequency tracking

Stephen A. Zahorian[a] and Hongbing Hu
Department of Electrical and Computer Engineering, State University of New York at Binghamton, Binghamton, New York 13902, USA

(Received 14 December 2006; revised 2 April 2008; accepted 7 April 2008)

In this paper, a fundamental frequency (F_0) tracking algorithm is presented that is extremely robust for both high quality and telephone speech, at signal to noise ratios ranging from clean speech to very noisy speech. The algorithm is named "YAAPT," for "yet another algorithm for pitch tracking." The algorithm is based on a combination of time domain processing, using the normalized cross correlation, and frequency domain processing. Major steps include processing of the original acoustic signal and a nonlinearly processed version of the signal, the use of a new method for computing a modified autocorrelation function that incorporates information from multiple spectral harmonic peaks, peak picking to select multiple F_0 candidates and associated figures of merit, and extensive use of dynamic programming to find the "best" track among the multiple F_0 candidates. The algorithm was evaluated by using three databases and compared to three other published F_0 tracking algorithms by using both high quality and telephone speech for various noise conditions. For clean speech, the error rates obtained are comparable to those obtained with the best results reported for any other algorithm; for noisy telephone speech, the error rates obtained are lower than those obtained with other methods.
 [DOI: 10.1121/1.2916590]

PACS number(s): 43.72.Ar, 43.72.Dv [DOS] Pages: 4559–4571

I. INTRODUCTION

Numerous studies show the importance of prosody for human speech recognition, but only a few automatic systems actually combine and use fundamental frequency (F_0),[1] with other acoustic features in the recognition process to significantly increase the performance of automatic speech recognition (ASR) systems (Ostendorf and Ross, 1997; Shriberg *et al.*, 1997; Ramana and Srichland, 1996; Wang and Seneff, 2000; Bagshaw *et al.*, 1993). F_0 tracking is especially important for ASR in tonal languages, such as Mandarin speech, for which pitch patterns are phonemically important (Wang and Seneff, 1998; Chang *et al.*, 2000). Other applications for accurate F_0 tracking include devices for speech analysis, transmission, synthesis, speaker recognition, speech articulation training aids for the deaf (Zahorian *et al.*, 1998), and foreign language training. Despite decades of research, automatic F_0 tracking is still not adequate for routine applications in ASR or for scientific speech measurements.

An important consideration for any speech processing algorithm is performance using telephone speech, due to the many applications of ASR in this domain. However, since the fundamental frequency is often weak or missing for telephone speech and the signal is distorted, noisy, and degraded in quality overall, pitch detection for telephone speech is especially difficult (Wang and Seneff, 2000).

A number of pitch detection algorithms have been reported by using time domain and frequency domain methods with varying degrees of accuracy (Talkin, 1995; Liu and Lin, 2001; Boersma and Weenink, 2005; de Cheveigne and Kawahara, 2002; Nakatani and Irino, 2004). Many studies have compared the robustness of pitch tracking for a variety of speech conditions (Rabiner *et al.*, 1976; Mousset *et al.*, 1996; Parsa and Jamieson, 1999). However, robust pitch tracking methods, which can easily be integrated with other speech processing steps in ASR, are not widely available. To make available a public domain algorithm for accurate and robust pitch tracking, the methods presented in this in this paper were developed.

A key component in "yet another algorithm for pitch tracking" (YAAPT) is the normalized cross correlation function (NCCF) as used in the "robust algorithm for pitch tracking" (RAPT) (Talkin, 1995). However, in early pilot testing, the NCCF alone did not reliably give good F_0 tracks, especially for noisy and/or telephone speech. Frequently, the NCCF method alone resulted in gross F_0 errors (especially F_0 doubling for telephone speech) that could easily be spotted by overlaying obtained F_0 tracks with the low frequency part of a spectrogram. YAAPT is the result of efforts to incorporate this observation in a formal algorithm.

In this paper, we describe methods for enhancing and extracting spectrographic information and combining it with F_0 estimates from correlation methods to create a more robust overall F_0 track. Another innovation is to separately compute F_0 candidates from both the original speech signal and a nonlinearly processed version of the signal and then to find the "lowest cost" track among the candidates by using dynamic programming. The basic elements of YAAPT were first given in the work of Kasi and Zahorian (2002) and modifications were described in the work of Zahorian *et al.* (2006). In this paper, we give a comprehensive description of

[a]Author to whom correspondence should be addressed. Tel.: (607) 777-4846. FAX: (607) 777-4464. Electronic mail: zahorian@binghamton.edu.

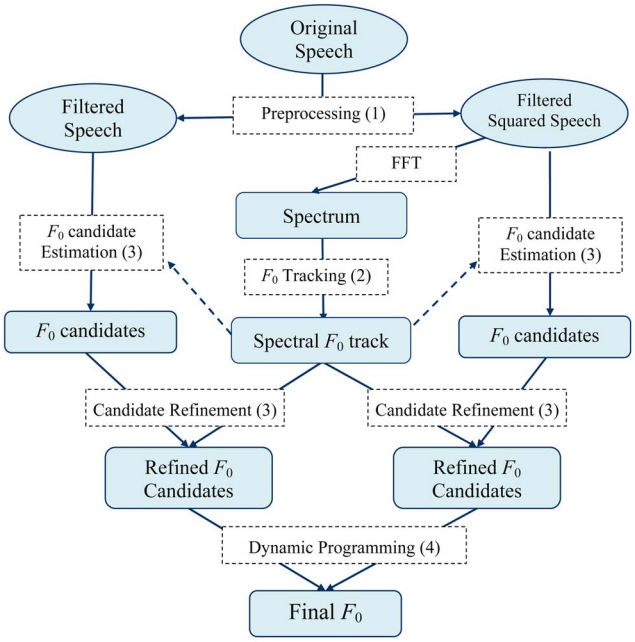

FIG. 1. (Color online) Flow chart of YAAPT. Numbers in parentheses correspond to the steps listed in Sec. II A.

the complete algorithm and extensive formal evaluation results.

II. THE ALGORITHM

A. Algorithm overview

The F_0 tracking algorithm presented in this paper performs F_0 tracking in both the time domain and frequency domain. As summarized in the flow chart in Fig. 1, the algorithm can be loosely divided into four main steps:

(1) Preprocessing: Multiple versions of the signal are created via nonlinear processing (Sec. II B).
(2) F_0 track calculation from the spectrogram of the nonlinearly processed signal: An approximate F_0 track is estimated by using a spectral harmonics correlation (SHC) technique and dynamic programming. The normalized low frequency energy ratio (NLFER) is also computed from the spectrogram as an aid for F_0 tracking (Sec. II C).
(3) F_0 candidate estimation based on the NCCF: Candidates are extracted from both the original and nonlinearly processed signals with further candidate refinement based on the spectral F_0 track estimated in step 2 (Sec. II D).
(4) Final F_0 determination: Dynamic programming is applied to the information from steps 2 and 3 to arrive at a final F_0 track, including voiced/unvoiced decisions (Sec. II E).

The algorithm incorporates several experimentally determined parameters, such as F_0 search ranges, thresholds for peak picking, filter bandwidths, and dynamic programming weights. These parameters are listed in Table I along with values used for experimental results reported in this paper. Similarly, to aid in the explanation of the algorithm and the error measures used for evaluation, primary variables used in this paper are given in Table II. The algorithm is frame based by using overlapping frames with frame lengths and frame spacings as given in Table I.

B. Preprocessing

Preprocessing consists of creating multiple versions of the signal, as shown in the block diagram of Fig. 1. The key idea is to create two versions of the signal: bandpass filtered versions of both the original and nonlinearly processed signals. The bandwidths (50–1500 Hz) and orders (150 points) of the bandpass finite impulse response (FIR) filters were empirically determined by inspection of many signals in time and frequency and also by overall F_0 tracking accuracy. These two signals are then independently processed to obtain F_0 candidates by using the time domain NCCF algorithm, as discussed in Sec. II D.

1. Nonlinear processing

Nonlinear processing of a signal creates sum and difference frequencies, which can be used to partially restore a missing fundamental. Two types of nonlinear processing, the absolute value of the signal and squared value of the signal, were considered. Since experimental evaluations indicated slightly better F_0 tracking accuracy by using the squared value, the squared value was used for the primary experimental results reported in this paper. The general idea of using nonlinearities such as center clipping to emphasize F_0 has long been known (see the work of Hess, 1983 for an extensive discussion) but appears not to be used in most of the pitch detectors developed since about 1990. For example, the pitch detectors YIN (de Cheveigne and Kawahara, 2002) and DASH (Nakatani and Irino, 2004) do not make use of nonlinearities. Of the seven pitch detectors evaluated by Parsa and Jamieson (1999), only one used a nonlinearity (center clipping). Most previous use of nonlinearities in F_0 detection algorithms was aimed at spectral flattening or reducing formant strength, rather than restoring a missing fundamental (for example, the work of Rabiner and Schafer, 1978).

As shown in the work of Zahorian *et al.* (2006), the fundamental frequency (F_0) reappears by squaring the signal in which the fundamental is either very weak or absent, such as telephone speech. The restoration of the fundamental by using the squaring operation is also illustrated by using spectrograms in Fig. 2. The top panel depicts the spectrogram of a studio quality version of a speech signal, for which the fundamental frequency is clearly apparent. The middle panel shows the spectrogram of the telephone version of the same speech sample, for which the fundamental frequency below 200 Hz is largely missing. In contrast, the fundamental frequency is more clearly apparent in the spectrogram of the nonlinearly processed telephone signal shown in the bottom panel. A bandpass filter (50–1500 Hz) was used after the nonlinearity to reduce the magnitude of the dc component. This same effect was observed for many other examples.

TABLE I. Primary parameters used to configure YAAPT. Value 1 numbers are used to minimize gross errors; value 2 numbers are used to minimize big errors.

Parameter	Meaning	Value 1	Value 2
F_0_min	Minimum F_0 searched (Hz)	60	60
F_0_max	Maximum F_0 searched (Hz)	400	400
Frame_length	Length of each analysis frame (ms)	35	25
Frame_space	Spacing between analysis frames (ms)	10	10
FFT_length	FFT length	8192	8192
BP_low	Low frequency of bandpass filter passband (Hz)	50	50
BP_high	High frequency of bandpass filter passband (Hz)	1500	1500
BP_order	Order of bandpass filter	150	150
Max_cand	Maximum number of F_0 candidates per frame	6	6
NLFER_Thresh1	NLFER boundary for voiced/unvoiced decisions, used in spectral F_0 tracking	0.75	0.75
NLFER_Thresh2	Threshold for definitely unvoiced using NLFER	0.0	0.1
N_H	Number of harmonics in SHC calculation	3	3
WL	SHC window length (Hz)	40	40
SHC_thresh	Threshold for SHC peak picking	0.2	0.2
F_0_mid	F_0 doubling/halving decision threshold (Hz)	150	150
NCCF_Thresh1	Threshold for considering a peak in NCCF	0.25	0.25
NCCF_Thresh2	Threshold for terminating search in NCCF	0.85	0.90
Merit_extra	Merit assigned to extra candidates in reducing F_0 doubling and halving logic	0.4	0.4
Merit_pivot	Merit assigned to unvoiced candidates in definitely unvoiced frames	0.99	0.99
W_1	DP weight factor for V-V transitions	0.15	0.15
W_2	DP weight factor for V-UV or VU-V transitions	0.5	0.5
W_3	DP weight factor for UV-UV transitions	100	0.1
W_4	Overall weight factor for local costs relative to transition costs	0.07	0.9

TABLE II. Variable used in YAAPT on for evaluation of F_0 tracking.

Variable	Meaning
s	Speech signal in a frame
S	Magnitude spectrum of speech signal
n	Time sample index within a frame
t	Time in terms of frame index
f	Frequency in Hz
k	Lag index used in NCCF calculations
i,j	Indices uses used for F_0 candidates within a frame
T	Number of signal frames
SHC	Spectral harmonics correlation
F_0_spec	Spectarl F_0 track, all voiced
F_0_avg	Average of spectral F_0 track
F_0_std	Standard deviation of F_0 computed from spectral F_0 track
NLFER	Normilized low frequency energy ratio
merit	Figure of merit for a F_0 candidate, on a scale of 0 to 1
NCCF	Normilized cross correlation function
K_min	Longest lag evaluated for each frame
K_max	Shortest lag evaluated for each frame
F_0_mean	Arthimetic average over all frames of the highest merit nonzero F_0 candidates for each frame
BP	Back pointer array used in dynamic programming
G_err	Error rate based on large errors in all frame where reference indicates voiced speech
B_err	All large error, including those in G_err+errors of the from UV to V

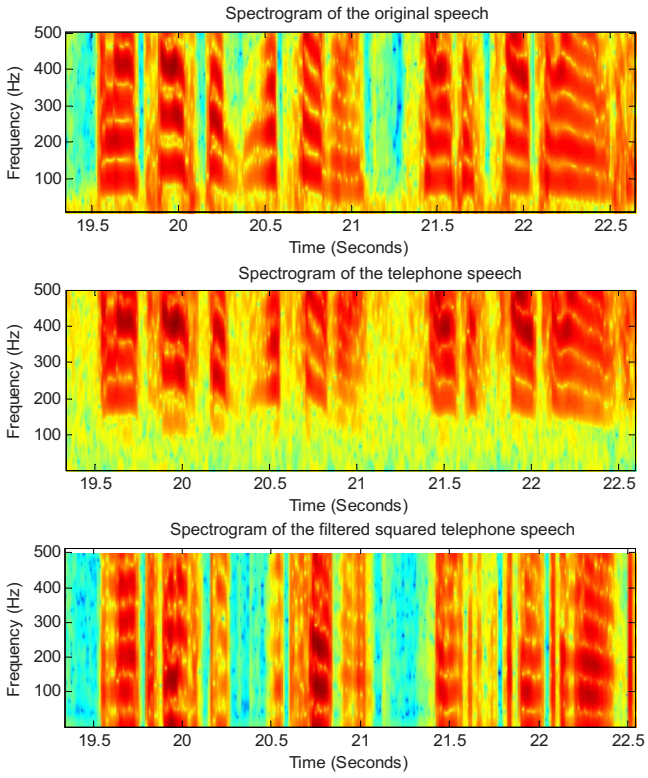

FIG. 2. (Color online) Illustration of the effects of nonlinear processing of the speech signal. The spectrogram of a studio quality speech signal is shown in the top panel, the spectrogram of the telephone version of the signal is shown in the middle panel, and the spectrogram of the squared telephone signal is shown in the bottom panel.

C. Spectrally based F_0 track

One of the key features of YAAPT is the use of spectral information to guide F_0 tracking. Spectral F_0 tracks can be derived by using the spectral peaks which occur at the fundamental frequency and its harmonics. In this paper, it is experimentally shown that the F_0 track obtained from the spectrogram is useful for refining the F_0 candidates estimated from the acoustic waveform, especially in the case of noisy telephone speech. The spectral F_0 track is computed by using the nonlinearly processed speech only.

The initial motivation for exploring the use of spectral F_0 tracks was that the examination of the low frequency parts of spectrograms revealed clear but smoothed F_0 tracks, even for noisy speech. The resolution of the spectral F_0 track depends on the frequency resolution of the spectral analysis, which, in turn, depends on both the frame length and fast Fourier transform (FFT) length used for spectral analysis. For the work reported in this paper, the values of these parameters are listed in Table I. Note that the frame lengths used (25 and 35 ms) are typical of those used in many speech processing applications. The FFT length of 8192 was chosen so that the spectrum was sampled at 2.44 Hz for a sampling rate of 20 kHz, the highest rate used for speech data evaluated in experiments reported in this paper. We hypothesized that this smoothed track could be used to guide the NCCF processing but that the NCCF processing, with a high inherent time resolution of one sampling interval, would give more accurate F_0 estimates. Ultimately, experimental evaluation is needed to check the accuracy of spectral F_0 tracking, versus NCCF-based tracking, versus a combined approach.

1. Spectral harmonics correlation

One way of determining the F_0 from the spectrum is to first locate the spectral peak at the fundamental frequency. This requires that the peak at the fundamental frequency be present and identifiable, which is often not the case, especially for noisy telephone speech. Although the nonlinear processing described in the previous section partially restores the fundamental, additional techniques are needed to obtain an even more noise robust F_0 track. Therefore, a frequency domain autocorrelation type of function, which we call SHC, is used. This method is conceptually similar to the subharmonic summation method (Hermes, 1988) and the discrete logarithmic Fourier transform (Wang and Seneff, 2000), but the details are quite different.

The spectral harmonics correlation is defined to use multiple harmonics as follows:

$$\mathrm{SHC}(t,f) = \sum_{f'=-\mathrm{WL}/2}^{\mathrm{WL}/2} \prod_{r=1}^{\mathrm{NH}+1} S(t, rf+f'),$$

where $S(t,f)$ is the magnitude spectrum for frame t at frequency f, WL is the spectral window length in frequency, and N_H is the number of harmonics. $\mathrm{SHC}(t,f)$ is then amplitude normalized so that the maximum value is 1.0 for each frame. f is a discrete variable with a spacing dependent on FFT length and sampling rate, as mentioned previously.

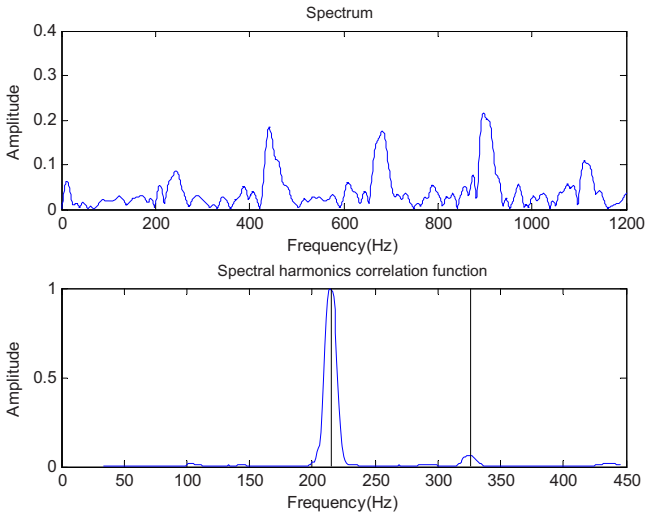

FIG. 3. (Color online) The peaks in the spectral harmonics correlation function. Compared to the small peak at the fundamental frequency of around 220 Hz in the spectrum (top), a very prominent peak is observed in the spectral harmonics correlation function (bottom).

For each frequency f, $\mathrm{SHC}(t,f)$, thus, represents the extent to which the spectrum has high amplitude at integer multiples of that f. The use of a window in frequency, empirically determined to be approximately 40 Hz, makes the calculation less sensitive to noise, while still resulting in prominent peaks for $\mathrm{SHC}(t,f)$ at the fundamental frequency. The calculation is performed only for a limited search range ($F_{0\_}\mathrm{min} \leqslant f \leqslant F_{0\_}\mathrm{max}$, with $F_{0\_}\mathrm{min}$ and $F_{0\_}\mathrm{max}$ values as given in Table I). Experiments were conducted to determine the best value for the number of harmonics. Empirically, it appeared that $N_H=3$ resulted in the most prominent peaks in $\mathrm{SHC}(t,f)$ for voiced speech and, thus, was used for the results given in this paper.

Figure 3 shows the spectrum (top panel) and the spectral harmonics correlation function (bottom panel). Compared to the small peak at the fundamental frequency of around 220 Hz in the spectrum, a very prominent peak is observed in the spectral harmonics correlation function.

2. Normalized low frequency energy ratio

Another primary use of spectral information in YAAPT is as an aid for making voicing decisions. The parameter used is referred to as the NLFER. The sum of spectral samples (the average energy per frame) over the low frequency regions is computed and then divided by the average low frequency energy per frame over the entire signal. In equation form, NLFER is given by

$$\mathrm{NLFER}(t) = \frac{\sum_{f=2\times F_{0\_}\mathrm{min}}^{F_{0\_}\mathrm{max}} S(t,f)}{\frac{1}{T}\sum_{t=1}^{T} \sum_{f=2\times F_{0\_}\mathrm{min}}^{F_{0\_}\mathrm{max}} S(t,f)},$$

where T is the total number of frames, and the frequency range, based on $F_{0\_}\mathrm{min}$ and $F_{0\_}\mathrm{max}$, was empirically chosen to correspond to the expected range of F_0. $S(t,f)$ is the spectrum of the signal for frame t and frequency f. Note that,

with this definition, the average NLFER over all frames of an utterance is 1.0. In general, NLFER is high for voiced frames and low for unvoiced frames and, thus, NLFER is used as information for voiced/unvoiced decision making. In addition, NLFER is used to guide NCCF candidate selection (Sec. II D).

3. Selection of F_0 spectral candidates and spectral F_0 tracking

Beginning with the SHC as described above, F_0 candidates were selected, concatenated, and smoothed by using the following empirically determined method and parameters. Values of the parameters used in experiments throughout this paper are listed in Table I.

(1) The frequency and amplitude of each SHC peak in each frame above threshold SHC_Thresh were selected as spectral F_0 candidates and merits, respectively. For the example shown in Fig. 3, two F_0 candidates were selected. If the merit of the highest merit F_0 candidate is less than SHC_Thresh or if the NLFER is less than NLFER_Thresh1, the frame is considered unvoiced and not considered in the following steps.
(2) To reduce F_0 doubling or halving for voiced frames (a persistent problem with pitch trackers, e.g., the work of Nakatani and Irino, 2004), an additional candidate is inserted at half the frequency of the highest merit candidate if all the candidates are above the F_0 doubling/having decision threshold F_0_mid. Similarly, if all candidates are below F_0_mid, an additional F_0 candidate is inserted at twice the frequency of the highest ranking candidate. The merit of these inserted candidates is set at the midrange value Merit_extra.
(3) All estimated voiced segments are concatenated and viewed as one continuous voiced segment. For each frame in this concatenated segment, one additional F_0 candidate is inserted as the median smoothed (seven point smoothing window) value of the highest merit candidate for each frame. This additional candidate is assigned a merit as Merit_extra.
(4) Dynamic programming, as described in Sec. II E, is used to select the lowest cost path among the candidates. This use of dynamic programming is the same as that used for final F_0 tracking, with the constants as listed in Table I. However, the transition costs involving unvoiced speech segments were relevant, since no unvoiced segments were considered.
(5) The F_0 track is then lengthened to its original length by using linear interpolation to span the sections estimated to be unvoiced from step 1 above.
(6) The result of this whole process is a smoothed F_0 track (F_0_spec) with every frame considered to be voiced. Experiments, reported in a later section, indicate that the spectral F_0 track is quite good but not quite as good as the one obtained by combining the spectral and NCCF tracks introduced in the next section.

D. F_0 candidate estimation from NCCF

F_0 candidates are computed from both the original and the nonlinearly processed signals by using a modified autocorrelation processing in the time domain. The basic idea of correlation based F_0 estimation is that the correlation signal has a peak of large magnitude at a lag corresponding to the period of F_0. This section explains the modified version used for YAAPT: the NCCF (Talkin, 1995), as well as the selection of NCCF F_0 candidates.

1. Normalized cross correlation function

The NCCF is defined as follows:[2] Given a frame of sampled speech $s(n)$, $0 \leq n \leq N-1$,

$$\mathrm{NCCF}(k) = \frac{1}{\sqrt{e_0 e_k}} \sum_{n=0}^{N-K\_\max} s(n)s(n+k),$$

where

$$e_0 = \sum_{n=0}^{n=N-K\_\max} s^2(n), \quad e_k = \sum_{n=k}^{n=k+N-K\_\max} s^2(n),$$

$$K\_\min \leq k \leq K\_\max.$$

In the equation, N is the frame length in samples and K_min and K_max are the lag values needed to accommodate the F_0 search range as described below. As with an autocorrelation, the NCCF is self-normalized for a range of $[-1,1]$ and periodic signals result in NCCF values of 1 at lag values equal to integer multiples of the period. As previously reported by Talkin (1995), the NCCF is better suited for F_0 detection than the "standard" autocorrelation function, as the peaks are better defined and less affected by rapid variations in signal amplitude. The only apparent disadvantage is the increase in computational complexity. Nevertheless, it is still possible for the largest peak to occur at double or half the correct lag value or simply at an "incorrect" value. Thus, the additional processing described below is used.

2. Selection of F_0 candidates and merits from NCCF

The following empirically determined procedure was used to create a collection of F_0 candidates and merits from the NCCF peaks:

(1) The spectral F_0 track (F_0_spec) was used to refine the search F_0 range for frame t as follows:

$$F_0\_\mathrm{search}\_\min(t) = \max[F_0\_\mathrm{spec}(t) - 2 \times \mathrm{std}, F_0\_\min],$$

$$F_0\_\mathrm{search}\_\max(t) = \min[F_0\_\mathrm{spec}(t) + 2 \times F_0\_\mathrm{std}, F_{0,}\max],$$

where F_0_std is the standard deviation of F_0 values appearing in the estimated spectral F_0 track.
(2) For each frame, all peaks found over the search range of F_0_search_min(t) to F_0_search_max(t) are located. To be a peak, a NCCF value must be at least NCCF_Thresh1

in amplitude and larger than the two values on either side of the point under consideration. If more than Max_cand/2 peaks are found, only the Max_cand/2 peaks with the highest values of NCCF are retained. Additionally, with searching beginning at a lag value corresponding to F_0_search_max(t) (shortest lag), if a peak is found with NCCF value greater than NCCF_Thresh2, peak searching is terminated. This step was empirically found to reduce F_0 halving instances. This process is repeated for all frames and for both the original and nonlinearly processed versions of the signal and the results combined for each frame. At the end of this step, up to Max_cand, F_0 candidates are found for each frame of the signal.

(3) All peaks found in step 2 are assigned a preliminary merit value equal to the amplitude of the peak. If fewer than Max_cand F_0 candidates are found in step 2, unvoiced candidates (F_0=0) are inserted, each with merit =[1−(merit of the nonzero F_0 candidate with the highest merit for that frame)]. For those frames where no peaks are found in step 2, the frame is preliminarily considered to be unvoiced; all F_0 candidates are set to 0 with merit=Merit_pivot.

(4) The initial merit values from step 3 are modified by using the spectral F_0 track, so as to increase the merits of NCCF F_0 candidates close to the spectral track. First, F_0_avg and F_0_std are computed as the average and standard deviation of F_0 from the spectral F_0 track (F_0_spec). Then, for candidates whose values are less than $5\times F_0$_std from the spectral F_0 value of that frame, the merit is changed as follows:

$$\text{merit}'(t,j) = \text{merit}(t,j) + 0.2 \times [1 - |F_0(t,j) - F_0\_\text{spec}(t)|/F_0\_\text{avg}],$$

where merit′ is the updated merit. For all other candidates, the merit is unchanged (merit′=merit). Note that j is the candidate index and t the frame index.

(5) For all frames with NLFER≤NLFER_Thresh2, the frame is considered to be definitely unvoiced and all F_0 candidates are adjusted to 0 (unvoiced) and have merits set to Merit_pivot. For all frames with NLFER >NLFER_Thresh2, the candidates are inspected to ensure that there is at least one nonzero F_0 estimate as well as an unvoiced candidate (F_0=0). If there initially was no nonzero F_0 candidate, the spectral F_0 is used as a candidate with a merit equal to half of the NLFER amplitude, if NLFER<2 or 1 if NLFER≥2. If there was initially no unvoiced ($F_0\neq 0$) candidate, the lowest merit F_0 candidate is replaced by the F_0=0 candidate, with merit=[1−(merit of the F_0 candidate with the highest meritfor that frame)], as in step 3.

E. Final F_0 determination with dynamic programming

After the processing steps mentioned above, a F_0 candidate matrix and associated merit matrix are created over the interval of a speech utterance. The F_0 candidates and the merits are used to compute transition costs, associated with every pair of F_0 candidates in successive frames, and local costs, for each candidate for each frame. In the remainder of this section, the calculation of these costs is described and the dynamic programming algorithm is summarized.

Three cases are considered for transition costs of successive F_0 candidates as follows:

(1) For each pair of successive voiced candidates (i.e., nonzero F_0 candidates),

$$\text{Cost}_{\text{transition}}(t-1,j{:}t,i) = W_1 \times |F_0(t,i) - F_0(t-1,j)|/F_0\_\text{mean}.$$

F_0_mean is the arithmetic average over all frames of the highest merit nonzero F_0 candidates for each frame. Note that the cost is for transitioning from candidate j in frame $t-1$ to candidate i in frame t.

(2) For each pair of successive candidates, only one of which is voiced,

$$\text{Cost}_{\text{transition}}(t-1,j{:}t,i) = W_2 \times [1 - \text{VCost}(t)],$$

where

$$\text{VCost}(t) = \min[1, |\text{NLFER}(t) - \text{NLFER}(t-1)|].$$

(3) For each pair of successive candidates, both of which are unvoiced,

$$\text{Cost}_{\text{transition}}(t-1,j{:}t,i) = W_3.$$

Values of W_1, W_2, and W_3 used in the experiments are given in Table I. The value of W_3 can be increased to a large value (e.g., 100) to force the dynamic programming routine to select all voiced candidates except for frames considered definitely unvoiced.

The local cost for each F_0 candidate is computed in the straightforward way,

$$\text{Cost}_{\text{local}}(t,i) = W_4 \times [1 - \text{merit}'(t,i)].$$

Thus, F_0 candidates with high merit have low local cost. W_4 is used to control the relative contribution of local costs to transition costs in the overall cost. The dynamic programming is a standard Viterbi decoding method, as described in the works of Rabiner and Juang (1993) and Duda *et al.*, (2000). The program is summarized here for completeness. Initialize:

$$\text{Cost}(1,i) = \text{Cost}_{\text{local}}(1,i), \quad 1 \leq i \leq \text{Max\_cand}.$$

Iterate: for $2 \leq t \leq T$,

for $1 \leq i \leq$ Max_cand,

$$\text{Cost}(t,i) = \text{MIN\_j}\ \{\text{Cost}(t-1,j) \times \text{Cost}_{\text{transition}}(t-1,j{:}t,i) + \text{Cost}_{\text{local}}(t,i)\},$$

$$\text{BP}(t,i) = \text{ARGMIN\_j}\ \{\text{Cost}(t-1,j) \times \text{Cost}_{\text{transition}}(t-1,j{:}t,i) + \text{Cost}_{\text{local}}(t,i)\}.$$

Max_cand and T are as, respectively, defined in Tables I and II. At the completion of the iterations over t, beginning with ARGMIN_i [Cost(T,i)], the BP array is traced back to

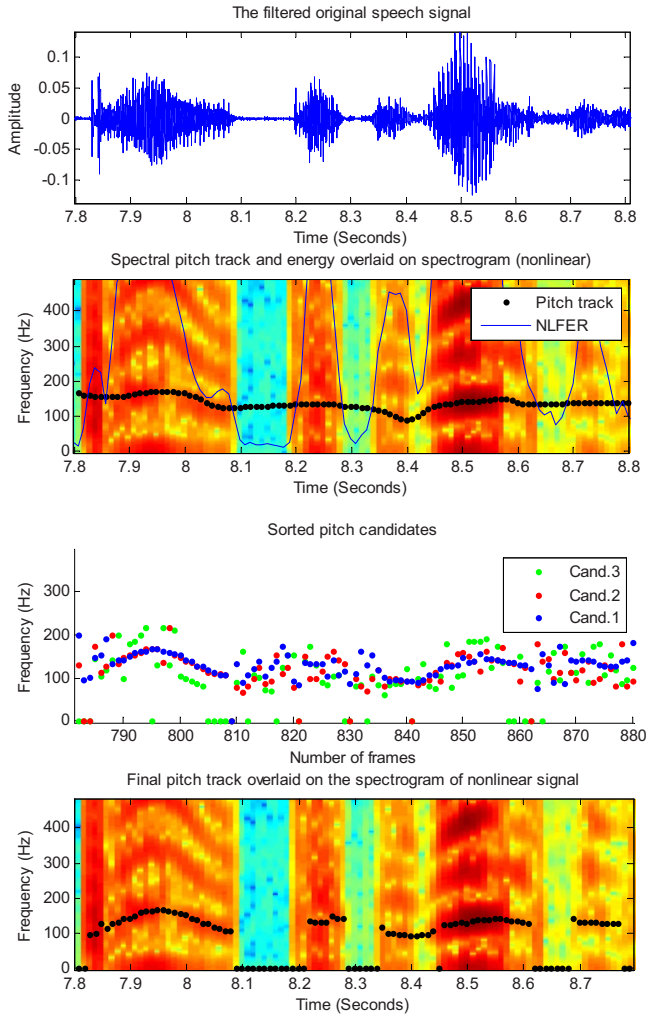

FIG. 4. (Color online) The first panel shows the time domain acoustic signal, the second panel shows the spectrogram of the signal with the low frequency energy ratio and spectral F_0 track overlaid on it, and the third panel shows multiple candidates chosen from the NCCF. The fourth panel shows the final F_0 track.

yield the overall lowest cost F_0 track. An illustration of the overall F_0 tracking algorithm is shown by the four panels in Fig. 4.

III. EXPERIMENTAL EVALUATION

A. Database description

In the F_0 estimation evaluation, performance comparison of different algorithms based on the same database are of great importance to allow better comparisons among the algorithms. Fortunately, common databases are freely provided for comparative pitch study by different research laboratories. For these databases, the laryngograph signal and/or a reference pitch are usually provided.

In our evaluation, we used the following three databases to evaluate various aspects of the YAAPT algorithm and to compare it with other algorithms:

(1) The Keele pitch database (DB1): This database consists of ten phonetically balanced sentences spoken by five male and five female English speakers (Plante *et al.*, 1995). Speech signals are studio quality speech sampled at 20 kHz. The total duration of the database is approximately 6 min. The laryngograph and the manually checked reference pitch are also provided in the database. The telephone version of the Keele database, formed by transmitting the studio quality speech signals through telephone lines and resampling at 8 kHz, was also used in experiments reported in this paper.

(2) The fundamental frequency determination algorithm evaluation database (DB2): This database is provided by the University of Edinburgh, UK (Bagshaw *et al.*, 1993). Fifty sentences are spoken by one male and one female English speaker. The total duration of the 100 sentences is about 7 min. The signal was sampled at a 20 kHz rate by using 16-bit quantization. The laryngograph and the manually checked reference pitch are also included.

(3) The Japanese database (DB3): This database consists of 30 utterances by 14 male and 14 female speakers (total of 840 utterances, total durations of 40 min, 16 kHz sampling, and 16-bit quantization). For experiments reported in this paper, 100 utterances were used, with approximately half of male speakers and half of female speakers. For this database, the reference used is the same one used in the works of de Cheveigne and Kawahara (2002) and Nakatani and Irino (2004).

B. Evaluation method

As the ground truth for pitch evaluation, the supplied reference pitches were used. These reference pitches were computed from the laryngograph signal and manually corrected. Although these references should be very accurate, by visual inspection of pitch tracks, they still appeared to have some problems with F_0 halving. Consequently, in previous studies, these references were not always used, but instead an algorithm-specific reference was computed from the laryngograph signal (for example, the work of Nakatani and Irino, 2004). Nevertheless, for experiments reported in this paper, supplied references were used for all results.[3]

To test the robustness of the algorithm, additive background noise was also used in the evaluation. The background noise consisted of two kinds of noise: white noise and babble noise. The signal-to-noise ratio (SNR) in terms of the average power ranges from infinity (that is no additional added noise or "clean") to 0 dB. The average power was calculated only from the frames whose power was more than 1/30 of the entire signal's average power (as per the work of Nakatani and Irino, 2004). Evaluations were made with two kinds of telephone speech: the actual telephone speech available in DB1 and simulated telephone speech for all three databases by using a SRAEN (300–3400 Hz 150th order FIR bandpass) filter.

C. Error measures

Errors for F_0 tracking include major errors [unvoiced (UV) frames incorrectly labeled as voiced (V), V frames incorrectly labeled as UV, and large errors in F_0 accuracy for voiced frames such as F_0 doubling or halving] and smaller errors in F_0 tracking accuracy. Of the many error measures

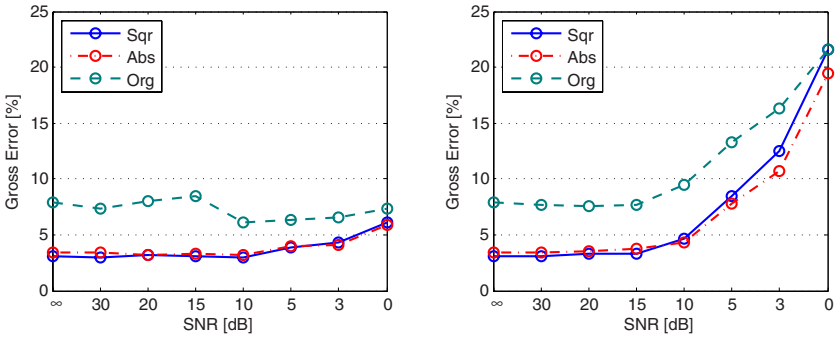

FIG. 5. (Color online) The effect of nonlinear processing for DB1 studio quality speech at various SNR white noises (left) and babble noises (right).

that can be used to quantify F_0 tracking accuracy, we used the following measures to evaluate the tracking method reported in this paper:

(1) Gross error (G_err): This is computed as the percentage of voiced frames, such that the pitch estimate of the tracker significantly deviates (20% is generally used) from the pitch estimate of the reference. The measure is based on all frames for which the reference pitch is voiced, regardless of whether the estimate is voiced or unvoiced. Thus, G_err includes V to UV errors as well as large errors in F_0 values,

$$\mathrm{G\_err} = \frac{1}{\mathrm{NVF}} \sum_{t=1}^{\mathrm{NVF}} \delta[F_0^{\mathrm{ref}}(t), F_0^{\mathrm{est}}(t)], \quad \delta(F_0^{\mathrm{ref}}, F_0^{\mathrm{est}}) = \begin{cases} 1 & |(F_0^{\mathrm{ref}} - F_0^{\mathrm{est}})/F_0^{\mathrm{ref}}| > 0.2 \\ 0 & \text{otherwise,} \end{cases}$$

where F_0^{ref} is reference F_0, F_0^{est} is estimated F_0, and NVF is the number of voiced reference frames.

(2) Big error (B_err): This error is equal to the number of voiced frames with a large error in F_0, plus the number of unvoiced frames erroneously labeled as voiced frames (UV_V_N), divided by the total number of frames T. In equation form,

$$\mathrm{B\_err} = (\mathrm{NVF} \times \mathrm{G\_err} + \mathrm{UV\_V\_N})/T.$$

Both G_err and B_err are expressed as percentages in experiments.

In the following sections of this paper, experimental results are first given to illustrate the effects of nonlinear processing and the performance of various components of YAAPT. These results are followed by a section with experiments and results based on the complete algorithm, including a comparison with three other algorithms (PRAAT, RAPT, and YIN) and a comparison with results reported in the literature using the same databases and the same error measures.

D. The effect of nonlinear processing

As described in Sec. II B, nonlinear processing could be either the absolute value or squared value, or a variety of other nonlinearities (Hess, 1983), to help restore the missing fundamental in the telephone speech. To evaluate the benefits of using this nonlinear processing, we computed the gross errors for three conditions: using the original signal only (no nonlinear processing), using absolute values as the nonlinear processing, and using the squared value as the nonlinear processing.

Figures 5 and 6, respectively, show the gross errors for studio quality speech and for telephone speech for various noise conditions using DB1. Error performance is very similar using either the absolute value or squaring operation. The nonlinear processing is quite beneficial for nearly all conditions tested, except for very high levels of additive babble noise. The most surprising result is that the nonlinear processing improves error performance even for noise-free studio quality speech.[4]

E. Evaluation of individual components of the algorithm

YAAPT computes the F_0 track by using a combination of both spectral and temporal (NCCF) information. The spectral F_0 track is used to determine the F_0 search range for the NCCF calculations and to modify the merits of the temporal F_0 candidates. It could be questioned whether or not both the temporal and spectral tracks are needed and the

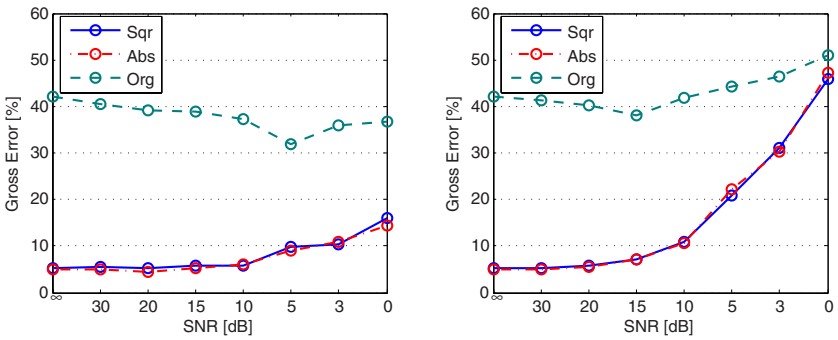

FIG. 6. (Color online) The effect of nonlinear processing for simulated DB1 telephone speech at various SNR white noises (left) and babble noises (right).

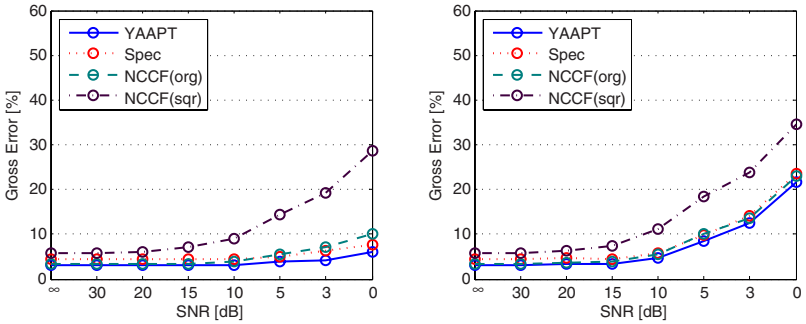

FIG. 7. (Color online) Performance based on individual components of YAAPT for DB1 studio quality speech at various SNR white noises (left) and babble noises (right).

extent to which each of these sources of information contributes to the accuracy of the F_0 tracking. Additionally, it might also be questioned whether or not the nonlinear processing is needed for the time domain F_0 candidates, especially for the case of studio quality speech. Therefore, F_0 tracking was computed by using four different approaches:

(1) using the NCCF candidates from the original signal only, with the final track determined by dynamic programming,
(2) using the NCCF candidates from the squared signal only, with the final track determined by dynamic programming,
(3) using the spectral F_0 track only, and
(4) using the entire YAAPT algorithm, combining both the temporal and spectral information.

Evaluations were conducted for each of these four methods by using both studio quality and telephone speech, and both added white and babble noises. Results are shown in Figs. 7 and 8. The combination of the temporal and spectral tracks results in better performance than using any individual component, illustrating the benefits of using both temporal and spectral information. As shown in Figs. 7 and 8, the gross error results based on the NCCF of the original signal is better than those obtained from the squared signal. For both the studio quality and telephone speech cases, the spectral F_0 tracking obtained by using the squared signal gives a very low gross error. These results, thus, show that the squared signal plays an important role in improving the performance of the entire algorithm for telephone speech.

F. Overall results

The overall evaluation of YAAPT is reported in this section, as well as a comparison with the PRAAT (Boersma and Weenink, 2005), RAPT (Talkin, 1995), and YIN (de Cheveigne and Kawahara, 2002) pitch tracking methods. The autocorrelation method described in the work of Boersma (1993) was used in PRAAT, as opposed to the cross-correlation method, as the autocorrelation option gave better results in pilot experiments. The RAPT tracker used is the MATLAB version of the Talkin algorithm. The RAPT pitch tracker was previously implemented commercially in XWAVES software and is considered to be a robust pitch tracker. More recently, the YIN tracker, which uses a modified version of the autocorrelation method, has been shown to give very high accuracy for pitch tracking for clean speech and music. The DASH and REPS trackers (Nakatani and Irino, 2004) are reported to be the most noise robust trackers developed for telephone speech.[5]

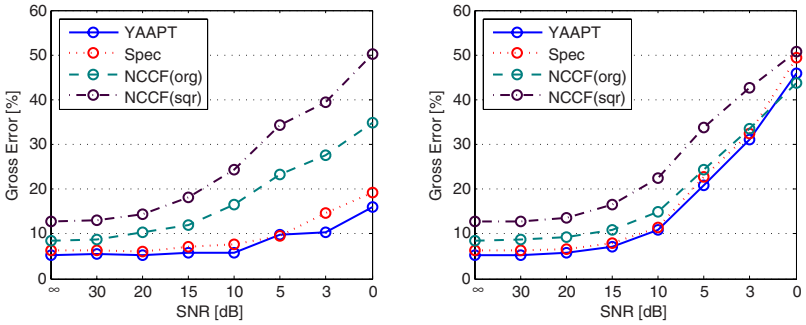

FIG. 8. (Color online) Performance based on individual components of YAAPT for DB1 telephone speech at various SNR white noises (left) and babble noises (right).

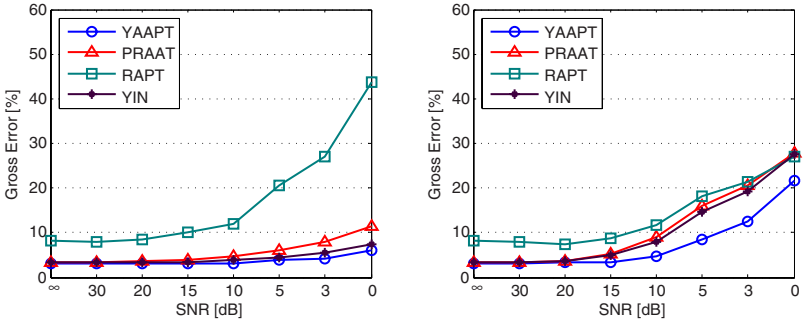

FIG. 9. (Color online) Gross errors for DB1 studio quality speech at various SNR white noises (left) and babble noises (right).

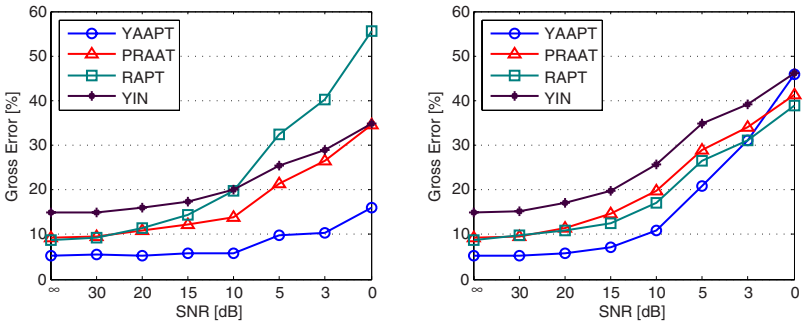

FIG. 10. (Color online) Gross errors for DB1 telephone speech at various SNR white noises (left) and babble noises (right).

1. Gross error results

Figure 9 depicts the gross F_0 errors of the studio quality speech for DB1 in the presence of additive white noise and babble noise, for the YAAPT, PRAAT, RAPT, and YIN pitch trackers. To obtain these results, the parameter values (e.g., Table I, column 1, for YAAPT) were adjusted so that nearly all frames were estimated to be voiced. Similarly, for the three control trackers, parameters were adjusted to minimize gross errors. Note that the gross F_0 errors are based on all large errors (including voiced to unvoiced errors) that a tracker makes for frames that are voiced in the reference. Figure 10 gives results of the telephone speech for the same conditions.

These results show that YAAPT has better gross error performance than the other methods, for all conditions at nearly all SNRs. The performance difference is greatest for telephone speech. The error performance of YAAPT is poor only for telephone speech with very high levels of additive babble noise (SNR$\leqslant$3 dB). It should be noted that this is very noisy speech; in informal listening tests, this speech was nearly unintelligible, with intermittent sections so noisy that the pitch was difficult to discern. Based on an inspection of F_0 candidates and the final F_0 track for YAAPT, it appeared that the final dynamic programming was unable to reliably choose the "correct" candidate for this very noisy condition.

In Table III, gross voicing error values for all three databases are listed for studio quality speech and simulated telephone speech. In this table, as well as other tables, results are given for clean speech, white noise at a 5 dB SNR (W-5), and babble noise at a 5 dB SNR (B-5). For both studio quality and telephone speech, with either no added noise or the W-5 condition, YAAPT has the best performance, sometimes dramatically better. However, for the B-5 telephone condition, YAAPT performance is sometimes worse (depending on database) than that of the other trackers. All four trackers are subject to large increases in error rates as signal degradation increases beyond a certain point.

2. Big error results

For some applications of F_0 tracking, both errors in voicing decisions and large errors in F_0 during voiced sections should be minimized. Thus, big error (B_err), as defined in Sec. III C and which includes both of these types of errors, is the most relevant measure of performance. The big error performance of YAAPT is compared only to that of the RAPT and PRAAT trackers, since the YIN tracker assumes that all frames are voiced. For all trackers, parameter settings were used that are intended to give the best accuracy with respect to big error (e.g., column 2 in Table I parameter values for YAAPT). Big error results, for studio and telephone speech, are shown in Fig. 11 as a function of SNR for added white noise. YAAPT performs better than PRAAT and RAPT for all conditions shown. The minimum big error performance of about 6% for studio quality speech is given by YAAPT. However, since most of the low frequency components are missing, higher big errors are obtained with tele-

TABLE III. Gross errors (%) for studio and simulated telephone speech for various noise conditions.

		Studio			Simulated telephone		
Database	Method	Clean	W-5	B-5	Clean	W-5	B-5
DB1	YAAPT	3.08	3.77	8.48	4.23	6.21	28.66
	PRAAT	3.35	6.91	15.98	9.91	15.72	32.56
	RAPT	8.24	21.33	18.04	9.5	18.21	29.09
	YIN	3.23	4.85	14.74	20.9	25.96	37.4
DB2	YAAPT	3.78	3.81	9.6	4.93	7.8	37.24
	PRAAT	5.96	10.5	19.61	7.81	19.03	32.53
	RAPT	14.08	30.63	23.76	14.63	30.68	30.43
	YIN	3.79	5.36	15.12	13.42	20.13	31.12
DB3	YAAPT	1.16	1.69	4.3	3.63	5.55	21.76
	PRAAT	2.02	3.97	14.75	5.35	11.93	29.7
	RAPT	5.36	12.87	13.78	3.84	13.97	24.88
	YIN	1.38	2.24	12.03	13.83	19.38	32.64

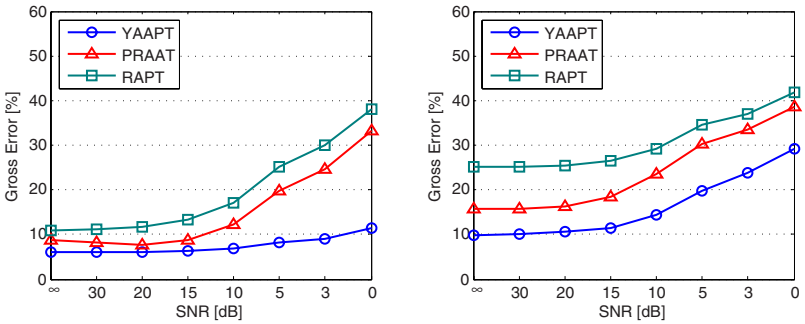

FIG. 11. (Color online) Big error for DB1 studio quality (left) and telephone (right) speech at various SNR white noises.

phone speech. In addition, high noise levels greatly affect the performance of the voiced/unvoiced determination, which, in turn, increase the big error.

A tabular presentation of big error performance is given for YAAPT, PRAAT, and RAPT in Table IV, for studio and simulated telephone speech, for the same noise conditions as used for the gross error results given in Table III. For all cases and all trackers, errors in voicing decisions (UV to V and V to UV) formed the largest portion of the big errors. For these results, YAAPT has the lowest error among the trackers for studio speech but not for the simulated telephone speech. However, as indicated by the results shown in Fig. 11, YAAPT does have the best big error performance for actual telephone speech.

3. Results with telephone speech

To examine results in more detail for real telephone speech, both gross error results and big error results are given in Table V, for the same noise conditions as used in Tables III and IV. YAAPT is compared to PRAAT, RAPT, and YIN for gross errors but to only PRAAT and RAPT for big errors. YAAPT has lower gross and big errors than PRAAT, RAPT, and YIN for the no added noise and W-5 conditions; for big errors in the B-5 condition, YAAPT has similar (poor) performance to PRAAT and RAPT.

G. Comparison of results with other published results

Selected results for gross errors obtained with YAAPT and YIN in this study are tabulated in Table VI along with previously reported results for YIN, DASH, and REPS and for all three databases used in this study. Although test conditions and parameter settings are intended to be identical, clearly, there are differences since the results obtained with YIN in this study and those obtained with YIN in these previous studies are significantly different. There may have been some differences in the reference pitch used, method for simulating telephone speech, methods for adding noise, parameter settings, or even versions of the code used. Nevertheless, the conditions are reasonably close and general comparisons can be made. Overall, the previously reported gross error results for DASH are the lowest. The previously reported gross error rates for YIN are very low for clean studio speech and very high for noisy telephone speech, as compared to the two other trackers.

TABLE IV. Big errors (%) for studio and simulated telephone speech for various noise conditions.

		Studio			Simulated telephone		
Database	Method	Clean	W-5	B-5	Clean	W-5	B-5
DB1	YAAPT	6.09	7.99	22.44	14.07	16.89	44.39
	PRAAT	8.64	19.87	34.9	12.83	20.12	46.9
	RAPT	10.99	25.35	34.28	11.69	21.81	45.6
DB2	YAAPT	7.66	8.41	26.75	8.39	12.72	47.96
	PRAAT	10.43	16.28	37.96	9.98	16.01	45.21
	RAPT	13.54	21.42	35.81	14.44	19.05	39.75
DB3	YAAPT	4.98	7.46	15.05	12.23	16.59	35.54
	PRAAT	8.42	18.9	31.55	9.5	21.99	43.92
	RAPT	11.72	25.26	32.09	11.44	25.04	40.3

TABLE V. Gross and big errors for telephone speech using DB1 for various noise conditions.

		Gross errors (%)			Big errors (%)		
Database	Method	Clean	W-5	B-5	Clean	W-5	B-5
DB1	YAAPT	5.3	8.86	20.93	9.93	19.83	46.22
	PRAAT	9.33	21.5	28.96	15.78	30.38	44.97
	RAPT	8.8	34.29	26.53	25.09	34.3	38.07
	YIN	14.85	25.63	34.95	⋯	⋯	⋯

TABLE VI. Comparison of gross errors for YAAPT, YIN, DASH, and REPS. The "*" indicates the results reported by Nakatani and Irino (2004).

		Studio			Simulated telephone		
Database	Method	Clean	W-5	B-5	Clean	W-5	B-5
DB1	YAAPT	3.08	3.77	8.48	4.23	6.21	28.66
	YIN	3.23	4.85	14.74	20.9	25.96	37.4
	DASH*	2.81	3.32	16.5	3.73	4.15	20.0
	REPS*	2.68	2.98	12.3	6.91	8.49	26.2
	YIN*	2.57	7.22	31.0	7.55	14.6	40.0
DB2	YAAPT	3.78	3.81	9.6	4.93	7.8	37.24
	YIN	3.79	5.36	15.12	13.42	20.13	31.12
	DASH*	0.42	1.34	14.6	0.63	0.97	15.1
	REPS*	0.68	1.05	11.1	2.11	3.25	18.9
	YIN*	1.3	4.38	33.5	5.53	10.3	35
DB3	YAAPT	1.16	1.69	4.3	3.63	5.55	21.76
	YIN	1.38	2.24	12.03	13.83	19.38	32.64
	DASH*	0.3	0.43	8.82	0.73	1.55	14.1
	REPS*	0.26	0.29	4.9	2.11	2.67	12.7
	YIN*	0.44	2.1	28.4	3.27	7.32	34.6

No similar comparisons can be given for big errors, since big error results are not reported for these databases. The focus for the YIN, DASH, and REPS trackers was tracking for the purpose of prosodic modeling, thus eliminating the need for voiced/unvoiced decision making. Consequently, results were only reported for gross errors, the large errors which occur in the clearly voiced (as per the reference) sections of speech.

IV. CONCLUSION

In this paper, a new F_0 tracking algorithm has been developed which combines multiple information sources to enable accurate robust F_0 tracking. The multiple information sources include F_0 candidates selected from the normalized cross correlation of both the original and squared signals and smoothed F_0 tracks obtained from spectral information. Although methods similar to all the individual components of YAAPT have been used to some extent in previous F_0 trackers, these components have been implemented and integrated in a unique fashion in the current algorithm. The resulting information sources are combined by using experimentally determined heuristics and dynamic programming to create a noise robust F_0 tracker. An analysis of errors indicates that YAAPT compares favorably with other reported pitch tracking methods, especially for moderately noisy telephone speech. The entire YAAPT algorithm is available from Zahorian as MATLAB functions.

Except for different settings used to evaluate gross error and big error, all parameter values used in the results reported in this paper were the same for all conditions tested. These conditions span three databases for two languages (English and Japanese), both studio quality and telephone speech, and noise conditions ranging from no added noise to 0 dB SNR with added white and babble noises. Over this wide range of conditions, F_0 tracking accuracy with YAAPT is better, or at least comparable, to the best accuracy achievable with other reported trackers.

From a computational perspective, YAAPT is quite demanding due to the variety of signal processing approaches used and then combined in the complete algorithm. For applications such as prosodic modeling where the voicing decision may not be needed, a very good voiced-only pitch track can be obtained by using the spectral pitch track method described in this paper, with greatly reduced computational overhead and only slight degradation in performance.

ACKNOWLEDGMENTS

This work was partially supported by JWFC 900 and NSF Grant No. BES-9977260. We would like to thank A. de Cheveigne, T. Nakatani, and T. Nearey for access to databases and control F_0 trackers. We also thank the anonymous reviewers for their detailed and helpful comments.

[1]In this paper, we use the terms F_0 and pitch interchangeably, although technically, pitch is a perceptual attribute, whereas F_0 is an acoustic property, generally considered to be the primary cue for pitch.

[2]This implementation of NCCF is slightly different from the one used in the second pass of RAPT, in that RAPT includes a small positive constant inside the radical, to reduce the magnitude of peaks in low amplitude regions of speech. Based on pilot testing, this constant did not improve F_0 tracking accuracy for YAAPT, so it was not used.

[3]Based on experimental testing, the patterns of error results obtained with supplied references and algorithm generated ones are very similar, except that the errors obtained with algorithm generated references are usually 1%–2% lower than those obtained with supplied references. This difference in performance is, thus, significant for clean studio speech but not significant for noisy telephone speech.

[4]It is quite likely that some modifications and changing of parameter values would have resulted in better performance of YAAPT without nonlinear processing, for studio speech. However, the experimental results shown were obtained without changing the algorithm or parameter values, except for changes in the nonlinear signal processing.

[5]The DASH and REPS trackers are proprietary code and were not available for comparison testing.

Bagshaw, P. C., Miller, S. M., and Jack, M. A. (**1993**). "Enhanced pitch tracking and the processing of the F_0 contours for computer aided intonation teaching," in *Proceedings of EUROSPEECH*, Berlin, Germany, 1003–1006. The database used in this paper is also available at http://www.cstr.ed.ac.uk/research/projects/fda (last accessed 4/1/2008).

Boersma, P. (**1993**). "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound," in *Proceedings of the Institute of Phonetic Sciences*, Vol. **17**, 97–110.

Boersma, P., and Weenink, D. (**2005**). "PRAAT: Doing phonetics by computer," version 4.3.14, Institute of Phonetic Sciences, http://www.praat.org (retrieved 5/26/2005).

Chang, E., Zhou, J., Di, S., Huang, C., and Lee, K. F. (**2000**). "Large vocabulary mandarin speech recognition with different approaches in modeling tones," in *Proceedings of the Sixth International Conference of Spoken Language Processing*, Interspeech 2000—ICSLP, Beijing, China, 983–986.

de Cheveigne, A., and Kawahara, H. (**2002**). "YIN, a fundamental frequency estimator for speech and music," J. Acoust. Soc. Am. **111**, 1917–1930.

Duda, R., Hart, P., and Stork, D. (**2000**). *Pattern Classification* (Wiley-Interscience, New York), pp. 128–137.

Hermes, D. J. (**1988**). "Measurement of pitch by subharmonic summation," J. Acoust. Soc. Am. **83**, 257–264.

Hess, W. (**1983**). *Pitch Determination of Speech Signals* (Springer-Verlag, Berlin), pp. 310–355.

Kasi, K., and Zahorian, S. A. (**2002**). "Yet another algorithm for pitch tracking," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, ICASSP, Orlando, Florida, 361–364.

Liu, D. J., and Lin, C. T. (**2001**). "Fundamental frequency estimation based on the joint time-frequency analysis of harmonic spectral structure," IEEE Trans. Speech Audio Process. **9**, 609–621.

Mousset, E., Ainsworth, W. A., and Fonollosa, J. A. R. (**1996**). "A comparison of several recent methods of fundamental frequency and voicing decision estimation," in *Proceedings of the Fourth International Conference on Spoken Language Processing*, ICSLP, Philadelphia, Pennsylvania, 1273–1276.

Nakatani, T., and Irino, T. (**2004**). "Robust and accurate fundamental frequency estimation based on dominant harmonic components," J. Acoust. Soc. Am. **116**, 3690–3700.

Ostendorf, M., and Ross, K. (**1997**). "A multi-level model for recognition of intonation labels," in *Computing Prosody*, edited by Y. Sagisaka, N. Campbell, and N. Higuchi (Springer-Verlag, New York), pp. 291–308.

Parsa, V., and Jamieson, D. G. (**1999**). "A comparison of high precision F_0 extraction algorithms for sustained vowels," J. Speech Lang. Hear. Res. **42**, 112–126.

Plante, F., Meyer, G., and Ainsworth, W. A. (**1995**). "A pitch extraction reference database," in *Proceedings of the Fourth European Conference on Speech Communication and Technology*, EUROSPEECH, Madrid, Spain, 837–840; Information about this database is available at http://www.liv.ac.uk/psychology/hmp/projects/pitch.html (last accessed 4/1/2008).

Rabiner, L., Cheng, M., Rosenberg, A., and McGonegal, C. (**1976**). "A comparative performance study of several pitch detection algorithms," IEEE Trans. Acoust., Speech, Signal Process. **ASSP-24**, 399–418.

Rabiner, L., and Juang, B.-H. (**1993**). *Fundamentals of Speech Recognition* (Prentice-Hall, Englewood Cliffs, NJ), pp. 204–208.

Rabiner, L., and Schafer, R. W. (**1978**). *Digital Processing of Speech Signals* (Prentice-Hall, Englewood Cliffs, NJ), pp. 150–157.

Ramana, R., and Srichand, J. (**1996**). "Word boundary detection using pitch variations," in *Proceedings of the Fourth International Conference on Spoken Language Processing*, ICSLP, Philadelphia, Pennsylvania, 813–816.

Shriberg, E., Bates, R., and Stolcke, A. (**1997**). "A prosody-only decision-tree model for disfluency detection," in *Proceedings of the Fifth European Conference on Speech Communication and Technology*, EUROSPEECH, Rhodes, Greece, 2383–2386.

Talkin, D. (**1995**). "A robust algorithm for pitch tracking (RAPT)," in *Speech Coding and Synthesis*, edited by W. B. Kleijn and K. K. Paliwal (Elsevier Science, New York), pp. 495–518.

Wang, C., and Seneff, S. (**1998**). "A study of tones and tempo in continuous mandarin digit strings and their application in telephone quality speech recognition," in *Proceedings of the Fifth International Conference on Spoken Language Processing*, ICSLP, Sydney, Australia, 635–638.

Wang, C., and Seneff, S. (**2000**). "Robust pitch tracking for prosodic modeling in telephone speech," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, ICASSP, Istanbul, Turkey, 1143–1146.

Zahorian, S. A., Dikshit, P., and Hu, H. (**2006**). "A spectral-temporal method for pitch tracking," in *Proceedings of the Ninth International Conference on Spoken Language Processing*, Interspeech 2006—ICSLP, Pittsburgh, Pennsylvania, 1710–1713.

Zahorian, S. A., Zimmer, A., and Dai, B. (**1998**). "Personal computer software vowel training aid for the hearing impaired," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, ICASSP, Seattle, Washington, Vol. **6**, 3625–3628.