

Enhancement of harmonic content of speech based on a dynamic programming pitch tracking algorithm

Mark R. Every and Philip J.B. Jackson

Centre for Vision, Speech and Signal Processing, Univ. of Surrey, Guildford, U.K.

m.every@surrey.ac.uk, p.jackson@surrey.ac.uk

Abstract

For pitch tracking of a single speaker, a common requirement is to find the optimal path through a set of voiced or voiceless pitch estimates over a sequence of time frames. Dynamic programming (DP) algorithms have been applied before to this problem. Here, the pitch candidates are provided by a multi-channel autocorrelation-based estimator, and DP is extended to pitch tracking of multiple concurrent speakers. We use the resulting pitch information to enhance harmonic content in noisy speech and to obtain separations of target from interfering speech.

1. Introduction

The problem of recognising speech in the presence of stationary noise or concurrent speech from an interfering speaker, from single-channel recordings is addressed, with reference to pitch tracking of one or more speakers. This is the main focus of a speech separation challenge[1]. The emphasis here is on developing methods for estimating the periodicities of one or more speakers, and using this information to enhance the harmonic content of a target speaker, or to suppress interfering speech.

Firstly, a pitch estimator is applied to a windowed segment of speech iteratively, as the window slides along the duration of the utterance. Section 2 describes the multi-band autocorrelation-based pitch estimator. In each time frame, the pitch estimator provides a set of *pitch candidates*. If successful, this set contains a candidate close to the correct pitch for frames in which a speaker is vocalising. To provide robustness when this is not the case, and to find the optimal path through the set of pitch candidates over all time frames, a pitch tracking algorithm is applied. Section 3 describes the dynamic programming (DP) algorithm for pitch tracking. The output indicates whether or not the speaker is voicing in any time frame, and for each voiced utterance, provides a fundamental frequency or f_0 trajectory. In the final stage, this information is used to enhance/suppress the harmonic content of a target/interfering speaker, by application of a filter containing resonances/nodes around the harmonics of f_0 .

Section 4.1 evaluates the pitch tracking performance, section 4.2 describes the procedures and word recognition rates for enhancement of a single speaker in stationary noise, and section 4.3 does the same for enhancement in the two talker case at various target-to-masker ratios (TMRs).

2. Pitch estimation

The time-domain autocorrelation of the waveform has been used previously as the basis for pitch estimation of speech, e.g., in YIN[2] and the software application Praat[3]. Approaches em-

ploying filter banks as front-ends, often aimed at modelling human audition or critical bands, also exist[4, 5]. A multi-band front-end offers an advantage when multiple periodic sources exist concurrently: bands contain different relative contributions of harmonic energy from each source. The periodicity of a particular source tends to dominate certain bands, so by combining periodicity information across bands, it is easier to determine multiple pitch hypotheses for a mixture of concurrent speakers.

The front-end for our pitch estimator is a multi-band filter-bank constructed from 15 second order IIR filters, with characteristic frequencies ranging from 60 to 1000 Hz, and with amplitude responses shown in fig. 1. The filter bank was chosen to provide a reasonable coverage of the frequency range between these limits, and the bandwidths of the filters were chosen empirically and determined by a single constant. There is no reason to believe that this filter bank is ideal, although it led to better pitch estimation results than two auditory-inspired filter banks also tested.

The pitch estimation method used was similar to [3], the main difference was a multi-band front-end used here, and so only salient points are summarised. We denote the output of a filtered band by $x[n]$, and a windowed segment starting at time $n = T$ by $a[n] = x[n + T]w[n]$, where $w[n]$ is a Hann window of length L , $n = 0, \dots, L - 1$. The autocorrelation of $a[n]$ is:

$$r_a[m] = \sum_{n=0}^{L-n-1} a[n+m]a[n] \quad (1)$$

and a similar expression exists for the window function, resulting in $r_w[m]$. The autocorrelation of $x[n]$ is approximately[3]:

$$r_x[m] = \frac{r_a[m]}{r_w[m]} \quad (2)$$

and this is normalised so that $r_x[0] = 1$. Eqn. 2 removes the tapering off of $r_a[m]$ at higher lags arising from the finite sum in eqn. 1. The window length was chosen to be $L = 0.05 f_s$ samples, where f_s is the sample rate. This corresponds to three periods when f_0 is equal to its minimum expected value, set at 60 Hz for speech. The autocorrelation maximum within an expected range of $[f_0^{\min} f_0^{\max}] = [60 \ 320]$ Hz was found:

$$s[m_{\max}] = r_x[m_{\max}] - \alpha \log_2 \left(\frac{f_0^{\min} m_{\max}}{f_s} \right). \quad (3)$$

where the last term above, with α set to 0.01, slightly favours higher frequencies. For a perfectly periodic source, autocorrelation peaks of unit magnitude would exist at lags equal to integer multiples of the period, and so this helps to avoid octave errors by choosing the maximum at the lowest lag. A threshold of 0.4

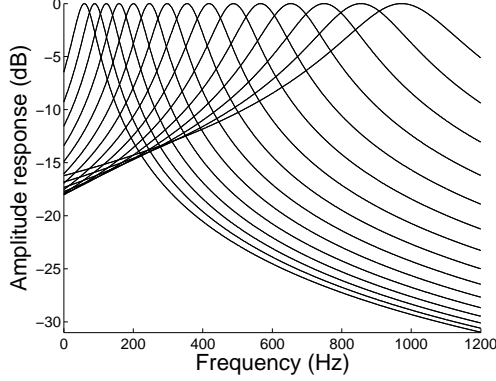


Figure 1: Filter bank front-end of 2nd-order IIR filters.

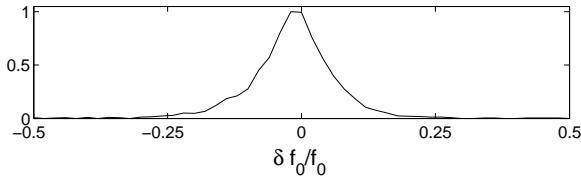


Figure 2: Normalised histogram of the relative change in f_0 between 5 ms frames, obtained from speech utterances in the Keele Pitch Database[6].

was set for $s[m_{\max}]$, below which the band was not considered to be sufficiently periodic to warrant a pitch candidate. This threshold provides robustness to noise and the multiple speaker case, by emphasizing bands for which one periodicity is dominant.

The autocorrelation maxima in all bands were summarised using an intermediate time-frequency map, $E[r, k]$, where r is the frame index and k the frequency bin, before identifying the pitch candidates in each frame. The nearest frequency bin corresponding to lag m_{\max} , $E[r, k]$ is incremented by $s[m_{\max}]$ for each band. The bin width of $E[r, k]$, and hence the resolution of the final pitch candidates, was 0.5 Hz. The pitch candidates, $f'_0[r]$, in each frame were obtained by peak picking from $E[r, k]$. The pitch candidate's strength $s'[r]$ was assigned the value of $E[r, k]$ at the peak.

When performing pitch tracking of two speakers, it was found worthwhile to obtain some additional information from $E[r, k]$. Suppose a given talker, indexed by j , has an average speaking pitch of μ_j , and we assume that the deviation in pitch for this speaker is approximately Gaussian-distributed around this mean with variance σ_j^2 . Then we can formulate a pitch prior for this speaker:

$$p(f_{0,j} | \mu_j, \sigma_j) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{(f_{0,j} - \mu_j)^2}{2\sigma_j^2}\right). \quad (4)$$

The estimates of σ_j and μ_j for the two speakers were obtained by least-squares-error minimisation of the difference:

$$\left| \sum_r E[r, k] - \sum_{j=1}^2 a_p p(f_{0,j} | \mu_j, \sigma_j) \right| \quad (5)$$

where the sum in r is over all time frames, and a_p reflects the relative amplitude of speaker j , which is also estimated in the optimisation. This is returned to in the next section.

3. Pitch tracking

Given a set of pitch candidates in each frame, we now wish to find the optimal path/s through them over time for one/multiple speakers. Physical constraints on speech would render some paths highly unlikely, for example, very large deviations in f_0 between consecutive voiced frames, or rapid switching between voicing and voiceless states. Fig. 2 summarises f_0 data from a set of manually transcribed speech recordings in the Keele Pitch Database[6]. A histogram is shown of the relative change in f_0 between consecutive voiced frames. This is converted to a probabilistic measure of f_0 variation in the tracking algorithm.

A score function is associated with each path through the pitch candidates, and $f'_0 = 0$ is included as a voiceless candidate. Of course, a complete evaluation of every possible path through the data would be computationally infeasible. A dynamic programming (DP) algorithm was used for tracking, which in its simplest form, reduces the computational complexity by iteratively removing suboptimal paths leading up to frame r , and continuing only the remaining tracks to frame $r + 1$. Fig. 3 illustrates some scenarios which the DP algorithm should be robust to, and we discuss the computational cost associated with each of these.

In fig. 3a, there are clearly two possible paths through each of the two pitch candidates in frame $r - 1$. The higher scoring of each of these pairs is carried through to frame r . If frames $r - 1$ and r contain n_{r-1} and n_r pitch candidates, respectively, then the computational requirement in computing the scores of all possible paths to frame r is $T_r \propto n_{r-1}n_r$. In other words, as each candidate in frame $r - 1$ can connect to any candidate in frame r , $n_{r-1}n_r$ possible permutations exist.

During voicing, a 'correct' pitch candidate may not be detected in some frames. To avoid treating these frames as voiceless, and inflating the number of voicing transitions, the tracking algorithm is allowed to skip small gaps (see fig. 3b), thus increasing the computational requirement to $T_r \propto (n_{r-g-1} + \dots + n_{r-1})n_r$, where g is the maximum allowable gap in frames.

Generalising to the M speaker case, and setting $g = 0$ for the moment, we firstly specify that a path contains a sequence of pitch candidates, $\{f_0^{m'}\}$, for each speaker, $m = 1, \dots, M$, and that at no point along a path should a voiced pitch candidate be assigned to more than one speaker concurrently. A voiceless pitch candidate may be assigned to any number of speakers. Fig. 3c indicates two paths to frame r from $r - 1$, and others exist. Not surprisingly, the number of possible paths increases rapidly with the number of speakers. It can be shown that the computational complexity at frame r increases dramatically to:

$$T_r \propto M Q_{r-1} Q_r \quad (6)$$

where

$$Q_r = \sum_{j=0}^M {}^M C_j \prod_{k=1}^j (n_r - k)$$

and ${}^M C_j = \frac{M!}{j!(M-j)!}$. In eqn. 6, the factor M emerges because the score function must be computed for each speaker. Q_r is the number of possible ways to choose M pitch candidates from n_r candidates (the voiceless candidate may be used more than once), and shown as a function of M and n_r in fig. 4. If gaps are allowed in any speaker's track, the complexity in frame r depends on $\{Q_{r-g-1}, \dots, Q_r\}$. Furthermore, to find the optimal path, the factor corresponding to $Q_{r-1} Q_r$ in eqn. 6 is larger than $\{Q_{r-g-1} + \dots + Q_r\}$, as gaps across consecutive pitch candidates

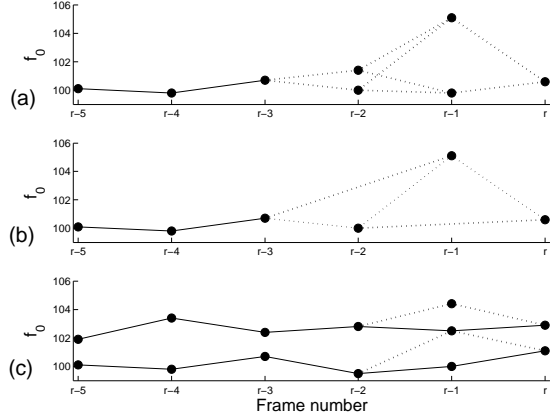


Figure 3: Paths through pitch candidate sequences, when there are (a) multiple pitch candidates per frame, (b) gaps allowed between consecutive candidates, and (c) two speakers.

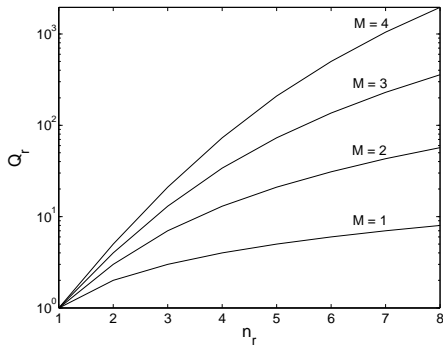


Figure 4: Q_r for M speakers vs. number of pitch candidates n_r .

differ across speaker. Thus, for computational efficiency, the DP algorithm was designed to be suboptimal when both $M > 1$ and $g > 0$, but optimal in all other cases. In practise, with the score function described below, this sometimes resulted in marginally suboptimal tracks being chosen if a gap occurred near the start of a voiced segment, but otherwise no artifacts were observed. A simple check was used to reduce the overall computational cost associated with the scoring function: if the frequency deviation between two consecutive voiced candidates along a path was larger than 15 Hz, the path was considered physically impossible, and so there was no need for the score to be evaluated.

The DP algorithm can be viewed as optimal only in the sense of yielding the path with the highest score. The scoring function summarises the nature of the data and the desired output of the system. The scoring function, S_r^j , for speaker j at frame r is:

$$S_r^j = \begin{cases} S_{r'}^j + R^j & ; v \rightarrow v \\ S_{r'}^j - P & ; v \rightarrow u \\ S_{r'}^j - P + \hat{R}^j & ; u \rightarrow v \\ S_{r'}^j & ; u \rightarrow u \end{cases} \quad (7)$$

where $r - g - 1 \leq r' \leq r - 1$ is the frame index of the last pitch candidate on the path preceding r , $v \rightarrow u$ indicates a transition from voiced to voiceless states, P (set empirically to 0.2 times the median of all non-zero pitch candidate strengths, $s'[r]$) is a penalty

for voicing transitions, and R^j is a reward for transitions between voiced states:

$$R^j = s'[r] p_\delta(\delta f) a_j p_{f_0}(f_0[r] | \mu_j, \sigma_j) p_g(r - r') \quad (8)$$

where $\delta f = \frac{f_0[r] - f_0[r']}{f_0[r]}$, $p_\delta(\delta f)$ is the histogram of relative frequency variation in fig. 2, and a_j and $p_{f_0}(f_0[r] | \mu_j, \sigma_j)$ are the estimated amplitudes and pitch priors for each speaker determined in section 2. $p_g(r - r')$ discourages gaps in tracks when it is possible to connect two pitch candidates by a path without gaps:

$$p_g(r - r') = \begin{cases} 1 & ; g = 0 \\ \exp\left(-\left(\frac{r-r'-1}{g}\right)^2\right) & ; g > 0 \end{cases} \quad (9)$$

\hat{R}^j in eqn. 7 is the reward for starting a new voiced segment:

$$\hat{R}^j = s'[r] p_\delta(\delta f) a_j p_{f_0}(f_0[r] | \mu_j, \sigma_j) \quad (10)$$

where δf is the relative frequency difference between f_0 and the last voiced pitch candidate (this encourages continuity in f_0 across consecutive voiced segments). The total score for a path is the sum of the scores over all speakers: $S_r = \sum_{j=1}^M S_r^j$.

We summarise the main characteristics of the scoring function. It favours time continuity of f_0 between frames and between consecutive voiced segments, it tends to separate tracks for the individual speakers into different frequency regions according to the speaker pitch priors, it tends to connect pitch candidates with large strengths, and it seeks to find a path with a minimal number of transitions between voiced and voiceless states.

4. Results

4.1. Pitch tracking

The pitch tracking algorithm was evaluated in the single speaker case against a set of reference pitch tracks obtained from speech and laryngograph data of ten speakers[6], upsampled to 200 Hz. Several different measures of the pitch tracking performance are given in table 1. *Voicing errors (VE)* occur where the reference and estimated pitch track disagree on the state of voicing and are given as a percentage of all frames. Voicing errors consist of *false alarms (FA)* and *false rejects (FR)*, which occur when the reference pitch is voiceless and voiced, respectively. Considering only frames agreed as voiced, *gross f_0 errors* are defined as those where the estimated pitch is closer on a logarithmic scale to half or double the reference pitch than the actual reference pitch. The remaining frames are classified as *matched (M)*, which is reported as a proportion of all voiced frames in the reference. Finally, *fine f_0 error* measures the RMS frequency difference between reference and estimated pitch tracks over all matched frames.

4.2. Single-speaker enhancement in noise

A database of speech data and an associated word recognition task[1] was used to evaluate the effectiveness of the proposed enhancement method for speech in stationary noise at various SNRs. Pitch tracks estimated as above were used to extract harmonic content of voiced segments from the original recording. This was achieved by filtering the original signal using comb-like filters in the spectral domain. Further details of the harmonic filtering approach are provided in [7], but the main points are as follows. A discrete short-time Fourier transform (STFT) of the speech signal

Table 1: Single speaker pitch tracking results.

VE %	FA %	FR %	gross %	M %	fine (Hz)
7.5	7.4	7.9	0.5	91.6	3.0

Table 2: Word recognition rate (%) for a single speaker in noise.

SNR (dB)	clean	6	0	-6	-12
original	98.6	56.7	18.9	11.8	11.7
processed	94.2	72.9	36.7	19.7	11.8

was calculated using a Hann window of length 2048 samples (82 ms at $f_s = 25$ kHz), and a hop size of 256 samples (10.2 ms). In any time frames of the STFT that were considered to be voiced, a binary spectral mask was constructed with ones at integer multiples of f_0 , and at the two neighbouring bins on either side of this, and zeros elsewhere. The binary mask was then multiplied by the corresponding frame of the STFT, in effect retaining only the harmonic content of this speaker. An inverse overlap-add synthesis method then re-synthesised the vocal segments of this speaker from the masked STFT representation.

Although the method for extracting harmonic content is an effective method for noise removal during voicing, on its own it does not provide much increase in speech intelligibility, as it makes no attempt to extract voiceless speech content. Hence, a two-fold approach was adopted: harmonic filtering was used when voicing was detected, and a noise removal technique using spectral subtraction[8] was used for all other frames. The spectral subtraction method relies on an estimate of the stationary noise, which was obtained from a 200 ms sample where the RMS amplitude of the original signal was at a minimum, i.e. where it was assumed that both speakers were silent. Table 2 compares the word recognition accuracy for speech within stationary speech-shaped noise at various SNRs, with and without processing. A default HMM-based recogniser using 39 Mel-frequency cepstral coefficients (including delta and acceleration coefficients)[1] was used. Given that the default recogniser[1] was applied to the processed data without any additional training, we would expect the results to be better, had the recogniser been re-trained on processed speech.

4.3. Speech enhancement in the two talker case

The two talker problem consisted of recognising target speech in the presence of an interfering speaker, at various TMRs. The original mixture was separated into two signals, and the RMS energy values of the two separated speaker were used to measure which was louder. It was then assumed at positive/negative TMRs that the target speech would be the louder/softer of the separated signals. Pitch tracks were obtained for each speaker (as in section 3). To enhance the first speaker’s speech, the harmonic content of the second speaker was subtracted from the original mixture. Then a two-fold procedure was applied to the residual: in frames where the first speaker’s voicing was detected, their voice was extracted by harmonic filtering; in all other frames, the residual was assigned entirely to the first speaker. The same procedure was applied to separate the second speaker from the mix. Finally, table 3 gives the word recognition rate using a default recogniser[1], split into three categories, where the target and masker are (i) the same (SS), (ii) of the same gender (SG), (iii) of different genders (DG), and (iv) gives the overall average (AVG).

Table 3: Word recognition rate (%) for the two speaker separation problem at various target-to-masker ratios (dB).

TMR	clean	6	3	0	-3	-6	-9
Original speech							
SS	98.0	62.4	46.2	29.6	18.1	9.7	5.7
SG	99.0	64.3	44.1	33.0	21.0	14.5	7.3
DG	99.2	64.3	46.8	33.5	19.5	11.5	7.5
AVG	98.7	63.6	45.8	31.9	19.4	11.8	6.8
Processed speech							
SS	86.2	39.6	26.0	20.1	13.8	8.6	7.9
SG	87.2	42.5	33.5	25.1	16.8	12.9	10.1
DG	85.5	48.0	40.0	29.3	17.8	16.3	14.5
AVG	86.3	43.3	32.9	24.7	16.0	12.4	10.8

5. Conclusions

The multi-band autocorrelation-based pitch estimation algorithm developed here has shown comparable performance to well-known pitch trackers such as Praat[3], in terms of both voicing errors and average accuracy of the detected f_0 , for the single speaker case. However, it can without any additional modification be applied also to the multiple speaker case. In the noisy speech task, word recognition rates were generally better than those for un-enhanced speech. This was not the case for the two talker task, except at very low TMRs, and additional tests may highlight whether this is due to inadequacies in pitch tracking or in the later enhancement procedure. Improvements could be expected for both tasks after re-training on processed data.

6. References

- [1] M. Cooke and T.-W. Lee, “Speech separation and recognition competition.” [Online] <http://www.dcs.shef.ac.uk/~martin/SpeechSeparationChallenge.htm>, Apr. 2006.
- [2] A. de Cheveigne and H. Kawahara, “YIN, a fundamental frequency estimator for speech and music,” *J. Acoust. Soc. Am.*, vol. 111, no. 4, pp. 1917–1930, 2002.
- [3] P. Boersma, “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, Amsterdam*, vol. 17, pp. 97–110, 1993.
- [4] M. Wu, D. Wang, and G. Brown, “A multipitch tracking algorithm for noisy speech,” *IEEE Trans. Speech and Audio Processing*, vol. 11, no. 3, pp. 229–241, 2003.
- [5] A. Khurshid and S. L. Denham, “A temporal-analysis-based pitch estimation system for noisy speech with a comparative study of performance of recent systems,” *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1112–1124, 2004.
- [6] G. Meyer, “Keele Pitch Database.” [Online] <http://www.liv.ac.uk/Psychology/hmp/projects/pitch.html>, Apr. 2006.
- [7] M. R. Every, *Separation of musical sources and structure from single-channel polyphonic recordings*. PhD thesis, Department of Electronics, University of York, U.K., 2006.
- [8] H.-T. Hu, F.-J. Kuo, and H.-J. Wang, “Supplementary schemes to spectral subtraction for speech enhancement,” *Speech Communication*, vol. 36, no. 3–4, pp. 205–218, 2002.