REAL-TIME IMPLEMENTATION, COMPARISON, AND COMBINATION OF PITCH TRACKING ALGORITHMS

Janina Reuter[1]*, Merikan Koyun*[2]*, Christoph Daniel Schulze*[1]*, Reinhard von Hanxleden*[1]
[1]*Kiel University,* [2]*sonoware GmbH*
stu204139@mail.uni-kiel.de

Abstract: Existing pitch tracking algorithms have proven to achieve error rates of less than 10% when applied to human speech. We take four such algorithms, YIN, RAPT, PEFAC, and TAPS, adapt them to real-time applications where necessary, and present improvements as well as a framework for combining them based on several aspects of their output to improve error rates further. The framework can be used with arbitrary pitch tracking algorithms.
We compare the adapted algorithms separately as well as their combination in our framework in terms of pitch tracking accuracy as well as how well they distinguish voiced and unvoiced signals. Our experiments indicate that PEFAC performs best when it comes to pitch tracking, but our framework performs best overall.

1 Introduction

Pitch is the "auditory perception of tone" [1]. This also applies to human speech, and methods to estimate its pitch have been researched for decades. Pitch tracking has a wide range of applications in different areas, such as communication, phonetics and linguistics, education, and medicine [2].

Current state-of-the-art pitch tracking algorithms have a less than 10% error rate on clean speech [3, 4]. While already quite good, it does leave room for improvement. An approach we follow in this paper is to combine algorithms in an attempt to increase the success rate. Consider three algorithms and assume for the moment that at most one is wrong at any given time. Then, going with the majority vote would yield a perfect success rate. This is, of course, an idealized scenario, but even if we drop our assumption it seems reasonable to expect that a combination of algorithms will increase the success rate.

Related Work In spite of the number of pitch tracking algorithms proposed in the past, only a few publications focused on combining existing algorithms. Yeh et al. [5] observed that different pitch trackers exhibit different types of pitch error. They proposed three approaches for combining an arbitrary number of pitch tracking algorithms in the context of extracting pitch from singing. Our approach differs in that it takes more information into account than only each algorithm's main pitch estimate which adds flexibility in the calculation.

Evaluations of YIN [6], PEFAC [3], and TAPS [4] are included in the respective papers. Furthermore, Jouvet and Laprie [7] and Luengo et al. [8] compared pitch tracking algorithms, including YIN, RAPT, and PEFAC.

Contributions Our main contribution is a real-time approach for combining an arbitrary number of pitch tracking algorithms. Besides main pitch estimates, it also considers continuity of estimates and secondary candidates for analysis of (sub-)harmonics. The system is evaluated on implementations of four existing, but adapted, pitch tracking algorithms.

This paper is based on a bachelor thesis [9] which goes into considerably more detail.

Outline We start with foundations in Section 2, followed by an overview of the four pitch tracking algorithms we shall consider in Section 3. Section 4 describes our approach for how to combine them. All are then evaluated in Section 5. Section 6 concludes the paper.

2 Foundations

An estimation is calculated over a *frame*, which is a sequence of discrete measures at a given rate and across a given time interval. In fact, not the pitch is estimated but, more precisely, the Fundamental Frequency (F0) which is similar most of the time. Algorithms operating in time domain often compare the similarity of the input signal with itself but shifted by a certain time, called *lag*.

The periodic signal in our speech is produced by pulses generated from an opening by the vocal cords. This is called *voiced* speech. A sound is called *unvoiced* if it is instead produced by constrictions in the vocal tract creating turbulent airflow when air is forced through. The estimation whether the current frame corresponds to voiced or unvoiced speech is called *voiced decision*. The F0 of voiced speech can change with every *glottal period* which is the time for one cycle of opening and closing the vocal cords.

A pitch tracking algorithm in our framework receives a frame as input and produces a main candidate, (multiple) secondary candidates, and a voiced decision as output. The main candidate is the usual estimation of an algorithm, while secondary candidates are expected to be connected to Fundamental Frequency (F0) as (sub-)harmonics.

3 Four Pitch Tracking Algorithms

This first subsection gives a brief overview of the four pitch tracking algorithms we implemented and highlights the changes we made to them. We then explain our choice of algorithms in the second subsection.

3.1 The Algorithms

YIN (or *Yin and Yang*) [6] operates in the time domain and consists of six steps. Instead of the autocorrelation method (1) the algorithm uses a function that takes the squares of the differences (2) and normalizes this value by the mean values for smaller lags (3), an absolute threshold to reduce octave errors (4), parabolic interpolation for more precise estimates (5), and a local search routine (6).

If any lag satisfies the threshold of step (4) in the algorithm, our implementation [9] considers the frame as *voiced*. In this case the signal and the signal shifted by the corresponding lag correlate well, indicating that the signal is periodic. Secondary candidates for YIN are the smallest lags satisfying the threshold.

RAPT (or *Robust Algorithm for Pitch Tracking*) [1] is an algorithm with a focus on robustness. Similar to YIN, RAPT operates in the time domain, but uses a different correlation function, namely the Normalized Cross Correlation Function (NCCF). To reduce computational complexity, the NCCF is first computed on a low sample rate before switching to a higher sample rate only in the vicinity of maxima found by the first pass. Dynamic programming is then used to select either the best F0 candidate or to characterize the frame as unvoiced.

Our implementation follows this two-pass approach, but does not use dynamic programming in order to achieve the real-time constraint. Hereby, the voicing decision uses a threshold

measuring the periodicity of the signal similar to the implementation for YIN. Again, the secondary candidates are the smallest lags satisfying the threshold.

PEFAC (or *Pitch Estimation Filter with Amplitude Compression*) [3] operates in the logarithmic frequency domain using a matched filter to leverage naturally occurring harmonics in speech. The input frequency spectrum is normalized as a preprocessing step based on the Long-Term Average Spectrum of Speech (LTASS) [10].

We deviate from the original algorithm by basing the voiced decision on the ratio between the maximum filter output value and its average. In addition, we added parabolic interpolation on the main candidate to increase the accuracy of estimates. We use the frequency corresponding to the highest filter values as secondary candidates.

TAPS (or *Temporally Accumulated Peak Spectrum*) [4] operates in the frequency domain as well and utilizes the fact that the fundamental frequency of speech changes much more slowly than the fundamental frequency of background noise. Therefore local maxima in the spectra of consecutive frames are accumulated and fed to an Autocorrelation Function (ACF). The ACF values are averaged in a small neighborhood and the estimate is calculated by a harmonic average of the highest maxima.

Our implementation adds a voicing decision based on the relative height of the maxima. We use frequencies corresponding to the highest local maxima as secondary candidates.

3.2 Why These Particular Algorithms?

When combining algorithms to improve the quality of estimates, it seems reasonable to choose algorithms based on different approaches. Good performance of the individual algorithms is of course a prerequisite for their combination to have a low error rate.

RAPT and YIN represent time domain algorithms and are considered as some of the "best performing" pitch tracking algorithms [11]. Since both are based on a correlation function, they follow the idea of measuring the similarity of the signal with a delayed version of itself. TAPS uses that fact that the human pitch and noise differ in their temporal properties. PEFAC, finally, combines the LTASS and a matched filter on the logarithmic frequency axis to obtain an estimate which is robust to noise.

Since these algorithms rely on different characteristics we expected that the frames where an estimate is false differs from algorithm to algorithm. However, the presented concept is not restricted to these algorithms. RAPT, YIN, TAPS, and PEFAC are presented here as examples of how to combine pitch tracking algorithms.

4 Combining the Algorithms

We propose a system called Candidates Evaluation (CE) which combines the estimates of multiple pitch tracking algorithms, here called candidates, in order to improve estimation quality. CE is based on a reward system, where the frequency with the highest score is chosen as the combined estimate. It was developed to make use of the individual strengths of each algorithm and to be very flexible such that algorithms can be added, exchanged, or removed easily. If a new algorithm is to be introduced, adaptations to the scoring system may improve the overall accuracy.

The general structure of CE allows for many degrees of freedom. The impact of different algorithms and rewards can be adapted to the application at hand. New reward strategies can be added or existing ones be removed, as can algorithms. Being independent from each other, the

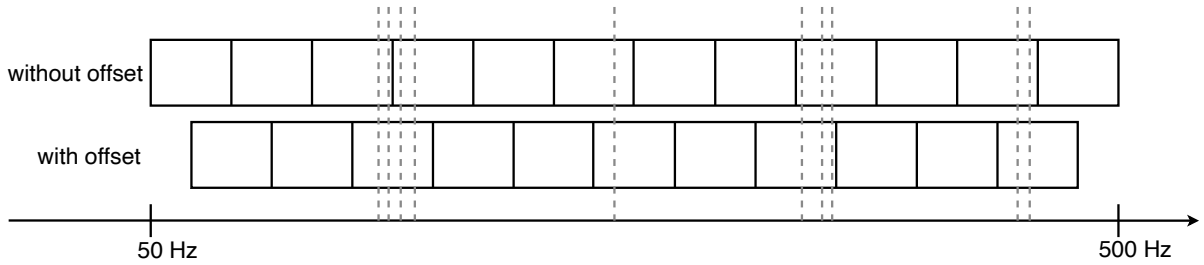

Figure 1 – Distribution of frequency bins with $K = 37.5$, with candidates drawn as dashed lines. Note that for the bin distribution without an offset, a maximum of three candidates end up in any given bin. However, if the bins are arranged with an offset, four candidates are assigned the third bin which seems reasonable.

candidates can be computed concurrently before CE combines them. By the simplicity of CE, only a small overhead beyond the runtime of the used algorithms is required.

4.1 Frequency Bins

Our approach is based on deriving scoring measures based upon the underlying pitch tracking algorithms. Doing so, however, poses a challenge in that settling on a final result is a somewhat fuzzy problem. First, even if the candidates are close, they might not be exactly equal. Second, the candidates might be spread out, and possibly even contradicting. Consider, for example, a situation which presents our scoring system with five candidates: 200 Hz, 300 Hz, 400 Hz, 500 Hz, and 600 Hz. If it is likely for algorithms to produce harmonics of the actual pitch, then 200 Hz seems like a reasonable choice since 400 Hz and 600 Hz are harmonics. However, the presence of 300 Hz and 500 Hz rather point to 100 Hz. Such ambiguities need to be properly resolved.

Our solution is based on partitioning a reasonable target frequency range into one sequence of frequency bins of size $K \in \mathbb{R}_+$ Hz. For human speech, we would expect that range to be about 50–500 Hz, resulting in the situation shown in Figure 1. Each candidate is then mapped to the appropriate bin. However, if several candidates flock around the border between two bins, their would be split even though they are likely to refer to the same frequency. This is why we introduce another sequence of bins with the same size, but at an offset of $K/2$—the first bin thus starts at $50 + K/2$ Hz. This causes candidates closely around $50 + nK$ Hz, $n \in \mathbb{N}$, to all be assigned to the same frequency bin. Of course, the frequency bin size influences how far candidates can deviate and still be considered to refer to the same fundamental frequency.

4.2 Reward System

The final estimate is based on a reward system where each frequency bin gains or loses points based on different information retrieved from candidates assigned to them as well as on the previous estimate. Each bin's final score is the sum of points thus obtained, with the mean of all candidates assigned to the highest scoring bin being chosen as the final estimate. This implies that the size of frequency bins determines how accurate our estimates can be. We will now describe all of the aspects our rewards are based on.

Primary Candidates Each primary candidate produced by an algorithm earns its bin a constant reward. The reward does not have to be constant across algorithms, though—algorithms known to be more reliable might yield higher scores than less reliable algorithms.

(Sub-)Harmonics While for TAPS, the secondary candidates contain harmonics of F0, RAPT, YIN, and PEFAC often yield subharmonics of F0. This can be used as a reward factor, although

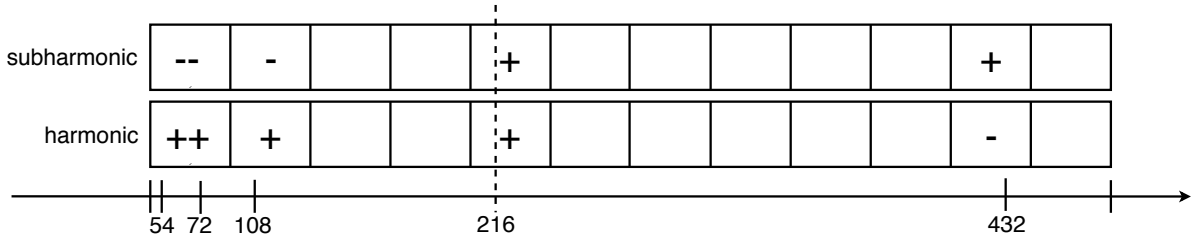

Figure 2 – Secondary candidates scoring example for a candidate at 216 Hz, interpreted as a subharmonic in the upper bins and as a harmonic in the bottom bins. Each plus and minus indicates a bin gaining and losing points, respectively.

it is not known which (sub-)harmonic the secondary candidate refers to.

An example for the scoring by one secondary candidate for the bins without offset is shown in Figure 2. For a subharmonic f_{sub} there exists $n \in \mathbb{N}$ with $nf_{\text{sub}} = \text{F0}$. The other way around, for a harmonic f_{har} there exists $n \in \mathbb{N}$ with $f_{\text{har}} = n\text{F0}$. If an algorithm is expected to produce subharmonics f_{sub}, bins including a frequency $f = nf_{\text{sub}}$ gain points and bins including a frequency $f = 1/n \cdot f_{\text{sub}}$ lose points for any $n \in \mathbb{N}$. Gaining and losing points is switched if instead an algorithm is expected to produce harmonics.

Estimation Continuity Apart from the target frequency range, we have thus far not taken any other properties of human speech into account. One such property is that the difference in fundamental frequencies of adjacent glottal periods is either small or exactly one octave [1]. Since most estimation errors are octave errors and sudden octave jumps rarely occur in reality, we only consider small changes, which we call *continuity*.

We support continuity by adding a constant to the score of bins in the vicinity of the previously chosen bin, since these are more likely to house the next fundamental than bins further away. What "vicinity" means, exactly, depends on the bin size. With 10 Hz bins, two bins in each direction performed well for detecting fast fundamental frequency changes while still being robust against octave errors.

Mean-Based Rewards For any given speaker, the deviation from the mean fundamental is often small. Therefore, mean-based rewarding of bins could be helpful for reduce octave errors, for example. Since pitch tracking generally is not restricted to a single speaker, though, we refrained from taking the mean into account.

One example of the mean being harmful could be a conversation between two people one with a mean fundamental of 70 Hz and one with 210 Hz. The pitch of the second person might then be interpreted as a harmonic of the first and, therefore, produce false estimates.

In settings where only a single speaker is present, though, including mean-based rewards may improve results further.

4.3 Voiced Decisions

Besides voiced decisions of the individual algorithms, our combined voiced decision also takes the previous voiced decision and estimation continuity into account. A voicing score is determined which, if it exceeds a threshold, results in the frame being considered voiced. Taking the previous voiced decision into account reduces fluctuations of the voiced state.

In contrast to the continuity of voiced speech, for unvoiced speech or silence the estimates of neighboring frames are unrelated. Thus, we reduce the voicing score if the current estimate is not in the vicinity of the previous. In our implementation, each voiced estimate produced by the underlying algorithms increases the voiced score by one, as does a previous voiced frame. With four algorithms, a threshold of 3 seemed to work well.

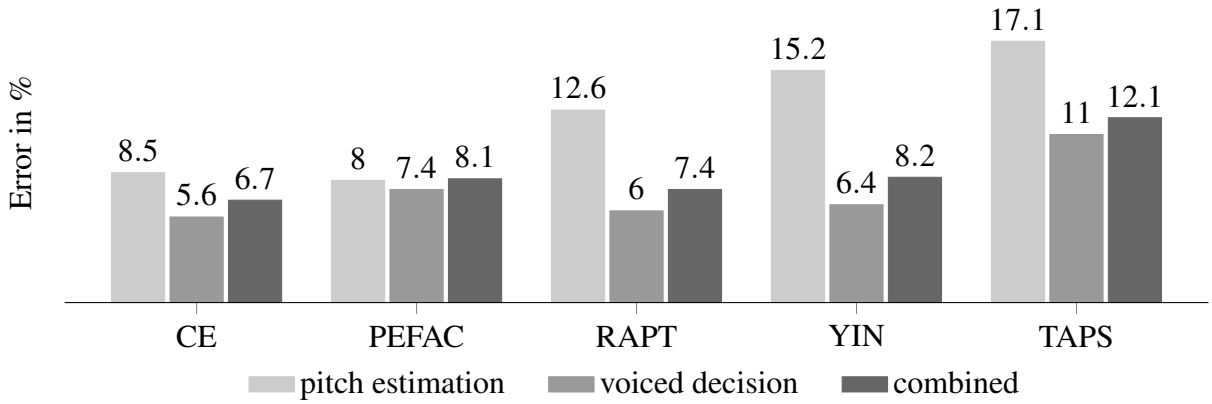

Figure 3 – Estimation error rates for each algorithm.

5 Evaluation

5.1 Experimental Method

We used the Pitch Tracking Database from Graz University of Technology (PTDB-TUG) [11] to evaluate pitch estimation quality. The database consists of 4720 recordings from 10 female and 10 male native English speakers pronouncing the 2342 sentences from the Texas Instruments/Massachusetts Institute of Technology (TIMIT) corpus. Two of these sentences are pronounced by each speaker and the other 2340 by one female and one male speaker each, hence yielding 4720 recordings. The database provides the speech signal, the laryngograph signal (which measures the contact of vibrating vocal folds), and a reference pitch determined by RAPT on the laryngograph signal for each utterance. The calculation on the laryngograph signal is expected to be more precise than for clean speech since the glottal excitation has not been manipulated by the vocal tract yet at this point.

Our evaluation used only 4430 recordings from this database, since we found 290 of the reference pitch files to contain unreasonable reference pitch values [9]. Since the estimation rate of the reference differs from the estimates made by our implementations, the reference pitch was interpolated linearly and then resampled to match our estimation rates. Unvoiced frames are indicated by an estimate of zero by the reference. Our interpolation would thus calculate unreasonable estimates between voiced frames (reference estimate is non-zero) and unvoiced frames (reference estimate is zero). To reduce the resulting pitch estimation error, the reference for such frames is set to unvoiced/zero. An estimate was considered correct if it fell within 5% of the reference pitch, which is similar to previous publications [3].

For each algorithm, we measured the error rates for pitch estimates, voiced estimates, and both combined. The pitch error only considers the frames that are voiced according to the reference. The combined error is defined by the number of frames that suffer from at least one error, divided by the total number of frames.

5.2 Analysis

Figure 3 shows the error rates for each algorithm and our framework. PEFAC performs well in all three categories. However, RAPT, YIN, and TAPS have a higher pitch estimation error rate than voiced decision and combined error rate in common. RAPT has the lowest voiced decision and combined error rates regarding the algorithms, in spite of a higher pitch estimation error rate compared to PEFAC. TAPS has the highest error rate in every category.

Our framework has overall the lowest error rates for voiced decision and combined. However, the pitch estimation error is slightly higher than for PEFAC.

5.3 Discussion

We set out to improve estimates by combining the results of different algorithms, and indeed the data seem to show that CE does improve the voiced estimation and combined error rates. Pitch estimation is more precisely estimated by PEFAC, though. This might be caused by RAPT, YIN, and TAPS having up to twice the error rate of PEFAC, which degrades the quality of the combined estimations. Put another way, however, CE improves the accuracy of three out of four algorithms significantly.

The configuration of the base algorithms whose results feed into CE has to be carefully designed to suit the application. The flexibility of CE allows for arbitrary compositions of input algorithms. One could even use certain algorithms for voicing detection or pitch tracking only, by setting the corresponding weights of the others to zero.

6 Conclusion

We described CE, an algorithm that combines the results of different pitch tracking algorithms to improve accuracy. It takes the continuity of estimates, harmonics and subharmonics, as well as main and secondary candidates into account and uses a scoring system to arrive at a combined result. It is flexible in that it can be used with an arbitrary selection of pitch tracking algorithms.

The evaluation showed an overall solid accuracy of CE. To improve on the result of pitch tracking accuracy, different input algorithms need to be considered that all have similar accuracy, in order to prevent one algorithm from degrading results too much. In our view, CE has the potential to further improve estimates for arbitrary algorithms that already perform reasonably well.

References

[1] TALKIN, D.: *A robust algorithm for pitch tracking (RAPT).* In *Speech Coding and Synthesis*, pp. 495–518. Elsevier Science, 1995.

[2] HESS, W.: *Pitch determination of speech signals: algorithms and devices*, vol. 3 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 1983. doi:10.1007/978-3-642-81926-1.

[3] GONZALEZ, S. and M. BROOKES: *PEFAC – a pitch estimation algorithm robust to high levels of noise*. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(2), pp. 518–530, 2014.

[4] HUANG, F. and T. LEE: *Pitch estimation in noisy speech using accumulated peak spectrum and sparse estimation technique*. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(1), pp. 99–109, 2013. doi:10.1109/TASL.2012.2215589.

[5] YEH, T.-C., M.-J. WU, J.-S. R. JANG, W.-L. CHANG, and I.-B. LIAO: *A hybrid approach to singing pitch extraction based on trend estimation and hidden Markov models*. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 457–460. 2012. doi:10.1109/ICASSP.2012.6287915.

[6] CHEVEIGHNÉ, A. D. and H. KAWAHARA: *YIN, a fundamental frequency estimator for speech and music*. *The Journal of the Acoustical Society of America*, 111(4), pp. 1917–1930, 2002. doi:10.1121/1.1458024.

[7] JOUVET, D. and Y. LAPRIE: *Performance analysis of several pitch detection algorithms on simulated and real noisy speech data*. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pp. 1614–1618. IEEE, 2017.

[8] LUENGO, I., I. SARATXAGA, E. NAVAS, I. HERNÁEZ, J. SANCHEZ, and I. SAINZ: *Evaluation of pitch detection algorithms under real conditions*. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, vol. 4, pp. IV–1057. IEEE, 2007.

[9] REUTER, J.: *Real-Time Pitch Tracking Algorithms in C to Test Model Extraction*. Bachelor thesis, Kiel University, Department of Computer Science, 2019.

[10] BYRNE, D., H. DILLON, K. TRAN, S. ARLINGER, K. WILBRAHAM, R. COX, B. HAGERMAN, R. HETU, J. KEI, C. LUI ET AL.: *An international comparison of long-term average speech spectra*. *The Journal of the Acoustical Society of America*, 96(4), pp. 2108–2120, 1994.

[11] PIRKER, G., M. WOHLMAYR, S. PETRIK, and F. PERNKOPF: *A pitch tracking corpus with evaluation on multipitch tracking scenario*. In *Twelfth Annual Conference of the International Speech Communication Association*. 2011.