COMPARISON OF F0 ESTIMATION ALGORITHMS FOR MONOPHONIC VOCAL SIGNALS

A PREPRINT

Logan Stillings
Sound and Music Computing
Universitat Pompeu Fabra
Barcelona, Spain
`logan.stillings01@estudiant.upf.edu`

March 29, 2021

ABSTRACT

The task of estimating the fundamental frequency of a monophonic human voice signal is a heavily researched topic in the field of sound and music computing. This type of pitch estimation is a core component of melody extraction systems. In this work I analyze the performance of two monophonic pitch trackers on vocal stems to provide baseline accuracy levels for comparison with melodic estimation algorithms.

***K*eywords** Monophonic f0 estimation · Melodic Estimation

1 Introduction

Estimating the melody from a sound mixture is a heavily researched topic in the field of sound and music computing. In order to define the task of melody estimation from polyphonic signals, one must first define the term "melody". While the concept of melody can be quite subjective, one common definition [1] states that "the melody is the single (monophonic) pitch sequence that a listener might reproduce if asked to whistle or hum a piece of polyphonic music, and that a listener would recognize as being the essence of that music when heard in comparison." This definition fails to account for the fact that different listeners may reproduce different instruments when humming the same song, however, in practice, the melody is a single audio source that pertains to the main instrument or voice in the piece of music. Estimation involves retrieving the fundamental frequency (f0) of the melody. The f0 of a signal refers to the physical property most closely related to the perceptual property of pitch [2], however for the purposes of this work, the terms pitch and f0 will be used interchangeably. Applications of melody estimation include transcription, removing effects from a vocal stem, synthesizing a single voice from unison, converting growling vocals to a clean voice, vocal changes/effects, pitch correction, and remixing. Ideally melody estimation could help audio engineers and singers alike, as well as improve the current research on singing voice extraction.

1.1 Melodic Estimation From Monophonic Signals

Melodic estimation methods are better understood when first examining methods of monophonic f0 estimation. Monophonic signals only contain a single note playing at any given time. One of the oldest and most intuitive methods of monophonic f0 estimation is autocorrelation. The autocorrelation of a discrete signal can be defined as:

$$r_t(\tau) = \sum_{j=t+1}^{t+W} x_j x_{j+\tau} \tag{1}$$

where $r_t(\tau)$ is the autocorrelation function of lag τ calculated at time index t, and W is the integration window size. When given a periodic signal, the autocorrelation function contains peaks at multiples of the period. In practice, the autocorrelation method chooses the highest non-zero-lag peak by searching within a range of lags [3]. However, this method is somewhat rudimentary and is prone to several types of errors. If the lower limit is too close to zero, the algorithm may erroneously choose the zero-lag peak. Conversely, if the higher limit is large enough, it may erroneously choose a higher-order peak. The YIN algorithm [3] was proposed to address these errors by implementing a cumulative mean normalized difference function. YIN was not as prone to amplitude changes and had the added benefit of not requiring an upper frequency bound for calculations. Although YIN provided significant improvements in accuracy of f0 estimation compared to autocorrelation, it outlines a smoothing procedure that does not use the frame-wise estimate. Probabilistic Yin (pYIN) [4] takes YIN and modifies its frame-wise variant using a probabilistic distribution to output multiple pitch candidates with their associated probabilities, while also reducing the loss of useful information before smoothing. pYIN uses Hidden Markov Models to output a monophonic pitch track of the most likely pitch candidates over time for monophonic signals. For many years, pYIN was considered the state of the art in monophonic f0 estimation. In recent years, many state of the art monophonic estimation systems implement deep learning algorithms. Algorithms such as CREPE [5] and SPICE [6] are data driven approaches, meaning that they learn from large volumes of data and are different from knowledge based approaches (like autocorrelation), which use domain specific information to influence processes. Both CREPE and SPICE show significant improvements in f0 estimation accuracy for monophonic signals when compared to pYIN. In this work, I will be comparing the performance of the CREPE and pYIN algorithms on monophonic vocal signals.

1.2 Algorithms

1.2.1 pYIN

pYIN [4], as previously mentioned, is a modification of the YIN algorithm [3] for fundamental frequency (f0) estimation. In the first step of pYIN, f0 candidates and their probabilities are computed using the YIN algorithm. In the second step, Viterbi decoding is used to estimate the most likely f0 sequence and voicing flags.

1.2.2 CREPE

CREPE [5] is a monophonic pitch tracker based on a deep convolutional neural network operating directly on the time-domain waveform input. CREPE is state-of-the-art (as of 2018), outperforming popular pitch trackers such as pYIN and SWIPE.

2 Tests

2.1 Data

MedleyDB [7] is a dataset of annotated, royalty-free multitrack recordings, which was curated specifically for the task of melody extraction. The dataset contains 108 multitracks with individual WAV files for the mix, processed stems, raw audio. The tracks are from a wide mix of genres such as Singer/Songwriter, Classical, Rock, World/Folk, Fusion, Jazz, Pop, Musical Theatre, Rap. In addition to the audio files, each song is provided with metadata information as well as ground truth annotations for the fundamental frequency of the melody. The annotations are stored in a CSV containing timestamp and f0 pairs. This dataset was chosen because of its access to annotations and raw vocal stems. In this work only raw vocal stems were analyzed.

2.2 Evaluation

Using the metadata provided with the tracks in the dataset, I determined which tracks contained vocals. I then implement the pYIN and Crepe algorithms on each of raw vocal tracks. The resulting f0 transcription is saved to a text file as timestamp and f0 pairs where the two columns are separated by a space and each row represents the next sequential timestamp.

For evaluating the performance of the two algorithms I implemented mir_eval [8], which is a Python library for evaluating Music Information Retrieval systems. More specifically, I use mir_eval's melody.evaluate method, which compares two melody transcriptions, where the first is treated as the reference and the second as the estimate to be evaluated. The method returns multiple calculated performance metrics, namely Raw Pitch Accuracy, Raw Chroma Accuracy, Voicing Recall, Voicing False Alarm, and Overall Accuracy [9]. The Raw Pitch Accuracy (RPA) measures the percentage of melody frames which the estimated pitch is within a half semitone of a reference. The Raw Chroma Accuracy (RCA) also measures pitch accuracy, but estimated and reference frequencies are mapped to a single octave. Voicing Recall (VR) measures the percentage of frames labeled as voiced (where the f0 is present and not 0) in the reference that are also labeled as voiced by the algorithm. Voicing False Alarm (VFA) measures the percentage of

Table 1: Average Evaluations

Algorithm	Metric	Accuracy(%)
CREPE	Voicing Recall	90.32
CREPE	Voicing False Alarm	23.39
CREPE	Raw Pitch Accuracy	80.61
CREPE	Raw Chroma Accuracy	80.85
CREPE	Overall Accuracy	**80.31**
pYIN	Voicing Recall	89.95
pYIN	Voicing False Alarm	31.30
pYIN	Raw Pitch Accuracy	71.13
pYIN	Raw Chroma Accuracy	71.99
pYIN	Overall Accuracy	**71.21**

frames labeled as un-pitched in the reference that are mistakenly estimated as voiced by the algorithm. Overall Accuracy (OA) measures the percentage of frames that were labeled correctly in terms of pitch and voicing. These metrics are commonly used across most f0 estimation systems.

3 Results

Table 1 contains the average of each evaluations, seperated by algorithm and metric. The Overall Accuracies are highlighted in bold. As you can see, the Overall Accuracy for CREPE is 80.31% and the Overall Accuracy for pYIN is 71.21%. These results support the fact that CREPE was proposed as an improvement to the pYIN algorithm, though this work specifically analyzes monophonic vocal signals. With these metrics, I can use the calculated accuracies of the CREPE algorithm as an expected baseline for a system capable of melodic voice esimation from polyphonic signals.

4 Discussion

While the results do support previous findings that CREPE outperforms pYIN [5], several considerations must also be taken into account for this analysis. The first consideration is that the CREPE algorithm was trained on several different datasets, including MedleyDB. Using the same data for training and testing the algorithm creates a bias in the performance of CREPE and could explain why the accuracy of CREPE is significantly higher than that of pYIN. Secondly, the annotation process for MedleyDB includes manual corrections which do not guarantee a 100% perfect match between the annotation and the audio. Therefore the results can also be affected by some degree of human subjectivity. Lastly, algorithm parameters such as the voicing confidence threshold (50%) chosen for CREPE and the min (65.41 Hz) and max (2093 Hz) frequencies chosen for pYIN may also affect the results. For future work, I recommend finding an additional dataset, other than one used in the training processes of these algorithms, to be used as a test set for evaluation. Ideally this dataset would not contain manual annotations, which are subjective and do not necessarily gaurantee a 100% match between the annotation and audio. And the algorithmic parameters such as voicing confidence threshold and min/max frequencies should be further analyzed to view differences in accuracy.

References

[1] G. E. Poliner, D. P. W. Ellis, F. Ehmann, E. Gómez, S. Streich, , and B. Ong. Melody transcription from music audio: Approaches and evaluation. *IEEE Trans. Audio, Speech, Lang. Processing*, 15:1247–1256, 2007.

[2] B. C. J. Moore. An introduction to the psychology of hearing. *Academic Press*, 2003.

[3] Alain de Cheveigne and Hideki Kawahara. Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, pages 1917–1930, 2002.

[4] M. Mauch and S. Dixon. Pyin: A fundamental frequency estimator using probabilistic threshold distributions. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 659–663, 2014.

[5] J. W. Kim, J. Salamon, P. Li, and J. P. Bello. Crepe: A convolutional representation for pitch estimation. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 161–165, 2018.

[6] Beat Gfeller, Christian Frank, Dominik Roblek, Matt Sharifi, Marco Tagliasacchi, and Mihajlo Velimirovic. Spice: Self-supervised pitch estimation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1118–1128, 2019.

[7] R. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello. Medleydb: A multitrack dataset for annotation-intensive mir research. *International Society for Music Information Retrieval Conference*, 2014.

[8] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis. mir_eval: A transparent implementation of common mir metrics. *Proceedings of the 15th International Conference on Music Information Retrieval*, 2014.

[9] Rachel Bittner and Juan J. Bosch. Generalized metrics for single-f0 estimation evaluation. *20th International Society for Music Information Retrieval Conference*, 2019.