Real-Time Polyphonic Pitch Detection on Acoustic Musical Signals

Thomas A. Goodman†, and Ian G. Batten‡
School of Computer Science, University of Birmingham
Birmingham, United Kingdom
Email: †t.a.goodman@cs.bham.ac.uk, ‡i.g.batten@cs.bham.ac.uk

***Abstract*—This paper presents an algorithm for fundamental frequency detection on polyphonic acoustic musical signals, based on a new 'raking' method over the frequency-domain spectra. The algorithm is evaluated as a classifier, and boasts a good accuracy (83.20%) compared to other such methods, as well as the ability to function effectively in real-time, with a running-speed below 140ms per window evaluated. This proves to be real-time for the use-case, as the latency between an auditory stimulus and its perception by a person has been shown to be longer than this. The algorithm itself runs in linear-time, but is thus slowed by the O(nlog(n)) Fast Fourier Transform during preprocessing. Though the algorithm fails to account for certain edge-cases with overlapping harmonics as well as certain instruments, future work and improvements are also presented, paving the way for further research.**

***Index Terms*—Signal Processing, Acoustics, Algorithms, Music Information Retrieval**

I. Introduction & Related Work

MUSIC has been a fundamental part of human culture for aeons - with the earliest known instruments found to date back c. 36000 years [10]. As such a deep-rooted facet of life, there is a plethora of musical theory as well as mathematics that underpins the field as a whole. With the emergence of computers in the last century, analysis of musical signals is an increasingly growing area of study that is host to a range of challenging problems. One such problem is pitch detection - the process of determining the pitch of a note (or each note in a set of notes).

For real-time pitch detection, the following are requirements [3]:

- Ability to function in real-time
- Minimal latency
- Accuracy in the presence of noise
- Sensitivity to the musical requirements of the performance

These requirements are born of the need to avoid initial pitch-identification errors. Whilst in pre-recorded music, errors can be rectified with further processing of the recording (multiple passes of analysis etc.), there is but a single window for analysis in real-time applications.

Regarding the ability to function in real-time, though this seems obvious or trivial as a requirement, it remains incredibly important to consider the computational complexity of the approach such that the calculated pitches at least appear to be instantaneous for the user.

Importantly there is a definitive lapse between the instant in which a note is perceived by a person and the point at which the person identifies the pitch (or even registers the auditory stimulus). In general, the latency between the auditory stimulus and the display of the calculated pitch should be less than the implicit latency between the start of the stimulus and the point at which the stimulus is registered by the brain. The mean simple reaction time for auditory stimuli has experimentally been shown to be between 140ms and 284ms [13], giving a window in which the processing of the signal can take place - i.e. the $\sim$140ms prior to the registering of the stimulus.

Algorithms which tolerate noisy audio input are particularly desirable [3]. There are many factors that could result in a "noisy" recording - the aforementioned harmonics of the acoustic instruments, imperfect microphone recordings, background noise and more.

A variety of methods for polyphonic pitch detection have been proposed, but it still remains an unsolved problem in the field of Music Information Retrieval (MIR) [1]. Further, there is both a variety of cutting edge ([1], [7], [14], etc.), and historical approaches ([11], [9], [12], [2] etc.) that attempt to tackle the problem.

II. Proposed Approach

The proposed approach is a novel algorithm that 'rakes' through the frequency-domain representation of the signal, extracting fundamental frequencies iteratively until there are no further frequencies to extract.

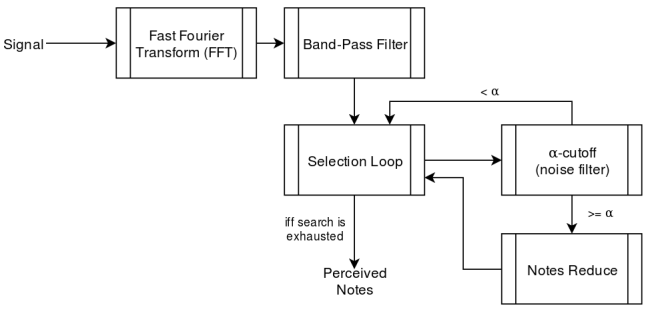

Figure 1. Diagram showing an overview of the proposed approach.

978-1-5386-7568-718$31.00

A. Preprocessing

The preprocessing for the proposed approach is minimal. Initially, the time-domain signal is transformed to the frequency domain by means of a Fast Fourier Transform. Further, a 'band-pass' filter is applied to the signal in the frequency domain, cutting off all frequencies below 60Hz or above 16000Hz, discarding them as noise as the likelihood of such an extreme frequency originating from an acoustic musical instrument is incredibly low.

Though simple, this preprocessing effectively reduces the number of possible fundamentals that the algorithm must consider, increasing the speed at which it can compute which fundamentals were perceived. Moreover, the upper and lower bound of the band-pass filter could be altered to be more optimal based on additional knowledge such as the instruments that are playing. For example, if the instrument perceived was a cello, the upper bound of the filter could be brought significantly lower as the cello is incapable of playing notes even close to that frequency.

B. Selection Loop

Algorithm 1 Selection Loop

Input: $\mathcal{F}$, $\mathcal{F}_k$, α, and $notesReduce$
Output: $\mathcal{R}$

$\mathcal{R} \longleftarrow \emptyset$
while $\mathcal{F}_k \neq [\ ]$ **do**
  $\mathcal{C} \longleftarrow \mathcal{F}_k[0]$
  $F_k \setminus \mathcal{C}$
  if $\mathcal{F}(\mathcal{C}) < \alpha\mu$ **then**
    continue
  end if
  $\mathcal{R} \cup \mathcal{C}$
  $\mathcal{F} \longleftarrow notesReduce(\mathcal{F}_k,\ \mathcal{F})$
end while

Inputs:

- $\mathcal{F}_{:c \to Double}$ - a mapping between the pitch chromas present in the signal and their respective total amplitudes
- $\mathcal{F}_{k:[c]}$, - the list of pitch chromas present, ordered by corresponding frequency (increasing)
- $\alpha_{:Double}$ - an expression for the minimum amplitude cutoff (i.e. the highest amplitude at which candidates should be discarded)
- $notesReduce_{:[c] \to (c \to Double) \to (c \to Double)}$ - a function that takes both $\mathcal{F}_k$, and $\mathcal{F}$ and returns a new $\mathcal{F}$ such that the total amplitudes of notes related to the fundamental are appropriately reduced.

Firstly, $\mathcal{R}$ is assigned the empty set, $\emptyset$, and acts as the return value of the algorithm - i.e. the perceived pitches. Then the algorithm loops until $\mathcal{F}_k$ (the list of keys of the mapping $\mathcal{F}$) is empty. This is to avoid mutation of $\mathcal{F}$ itself whilst iterating.

Due to the lack of harmonic 'undertones' in acoustic musical signals, the leftmost encountered harmonic (i.e. the harmonic at index 0 in $\mathcal{F}_k$) can be considered to always be the next candidate as a potential fundamental of the signal. Hence, on lines 3 and 4, the next candidate chroma is popped from $\mathcal{F}_k$, and assigned to $\mathcal{C}$. Depending on whether or not this value is deemed to be a fundamental (further in the algorithm), it will either be added to $\mathcal{R}$ or discarded as appropriate. It is imperative, however, to remove it from $\mathcal{F}_k$ at this point as it should never be considered more than once. This also ensures that F_k will always shrink and eventually the algorithm will terminate.

If the amplitude of the chroma $\mathcal{F}_k$ in $\mathcal{F}$ is below some threshold amplitude, $\alpha\mu$, which is some scalar, α, of the mean amplitude of the remaining chromas in $\mathcal{F}_k$, μ, it is discarded. A good value for α is experimentally determined in Section IV.

Otherwise, if the amplitude of the current chroma falls above the threshold, it is added to the perceived pitches, $\mathcal{R}$, and the amplitudes of all notes in $\mathcal{F}$ are reduced according to the 'Notes Reduce' function. The algorithm then loops again to check the next candidate.

C. 'Notes Reduce' Function

Algorithm 2 'Notes Reduce' Function

Input: $\mathcal{F}$, $\mathcal{F}_k$, $harmonicsFromChroma$, $spline$, $cToF$, and $\mathcal{V}$
Output: $\mathcal{F}$

$curr \longleftarrow \infty$
$points \longleftarrow [\ ]$
$fundamental \longleftarrow \mathcal{F}_k[0]$
$\mathcal{H} \longleftarrow harmonicsFromChroma(fundamental)$
for $\mathcal{F}_i \in \mathcal{F}_k$ **do**
  if $\mathcal{F}(\mathcal{F}_i) < curr \wedge \mathcal{F}_i \in \mathcal{H}$ **then**
    $points \cup \mathcal{F}_i$
    $curr \longleftarrow \mathcal{F}(\mathcal{F}_i)$
  end if
end for
$splineX \longleftarrow points \cup \mathcal{V}$
$splineY \longleftarrow [\mathcal{F}(x) | x \in points] \cup 0$
$splineF \longleftarrow spline(splineX, splineY)$
for $h \in \mathcal{H}$ **do**
  if $h \notin \mathcal{F}_k$ **then**
    continue
  end if
  $\mathcal{F}[h] \longleftarrow \mathcal{F}[h] - splineF(cToF(h))$
end for
return $\mathcal{F}$

Inputs:

- $\mathcal{F}_{:c \to Double}$ - a mapping between the pitch chromas present in the signal and their respective total amplitudes
- $\mathcal{F}_{k:[c]}$ - the list of pitch chromas present, ordered by corresponding frequency (increasing)
- $harmonicsFromChroma_{:chroma \to [chroma]}$ - a function that takes a chroma and returns its harmonics

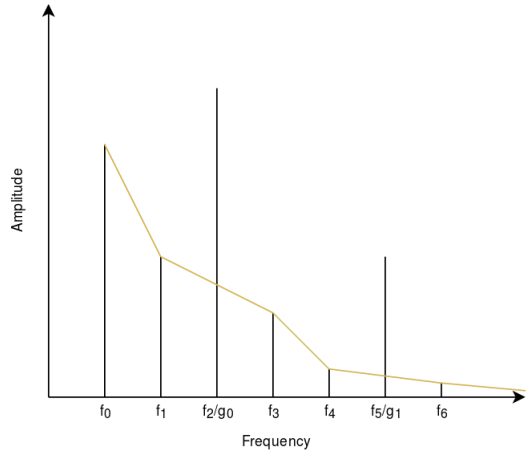

Figure 2. Example spline giving two fundamentals with overlapping harmonics, f_0 and g_0

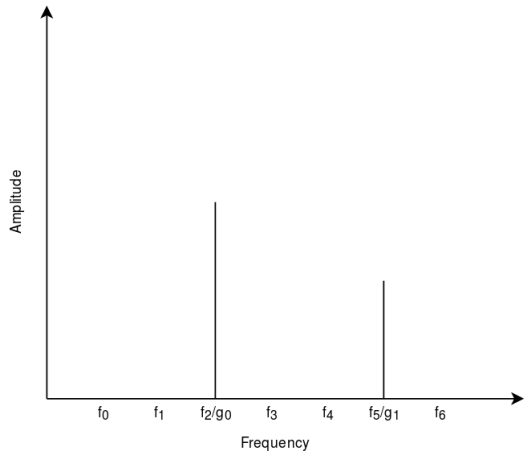

Figure 3. Result of the reduction on the previous spectra (Figure 2)

- $spline_{:[Double]\to[Double]\to(Double\to Double)}$ - A function that takes two equal-length lists of points and returns a corresponding spline function
- $cToF_{:c\to Double}$ - A mapping from chromas to their corresponding frequency
- $\mathcal{V}_{:Double}$ - The upper limit of sampled frequency (eg. $\sim$20000.0Hz for human hearing)

The 'Notes Reduce' function takes the remaining list of uncategorised chromas, $\mathcal{F}_k$, and the mapping $\mathcal{F}$, and uses these to reduce the amplitudes of the harmonics of the fundamental, $F_k[0]$, returning a new mapping $\mathcal{F}$ with updated amplitudes.

On lines 1 to 10 inclusive, *points* is assigned the list containing all of the harmonics of $\mathcal{F}_k[0]$ that have strictly monotonically decreasing amplitudes. This is based off of the assumption that the $(n+1)^{th}$ harmonic of a given fundamental will generally have less energy than the n^{th} harmonic. This allows the algorithm to account for overlapping harmonics, and more importantly, the case in which the harmonic of one note is the fundamental of another - for example, C4 and G5.

The monotonically decreasing points are used on lines 11 to 13 inclusive to fit a spline along the tips of the corresponding peaks, with the spline tapering off from the last point to the point ($\mathcal{V}$, 0) - i.e. the limit of human hearing. A spline with degree 1 (i.e. a spline consisting of linear sections) is both effective and incredibly quick to calculate in this case.

Finally on lines 14 to 20 inclusive, the amplitude of all harmonics of the fundamental are reduced (NB: not just the monotonically decreasing ones) by the amplitude of the spline at that point. The new mapping is then returned and the selection loop continues.

III. Evaluation Method

The problem of polyphonic pitch detection is in essence a classification problem. Given a set of notes, they need to be correctly classified or discarded, with the perfect outcome being that of the frequencies heard, only the perceived pitches are classified (and classified correctly), and the other frequencies are discarded. For each window there are two sets, the heard notes which are all of the frequencies that were recorded (and above some threshold), and the perceived notes, which are the output notes - i.e. the notes that the approach has selected as fundamentals and classified. Moreover each processed note heard falls into four categories,

- True Positive (TP) - A note that was in the heard notes and both correctly retained and classified.
- False Positive (FP) - A note that was in the heard notes and incorrectly retained (regardless of classification).
- True Negative (TN) - A note that was in the heard notes and correctly discarded (i.e. does not appear in the perceived notes)
- False Negative (FN) - A note that was in the heard notes and incorrectly discarded when it should have been retained.

Five different test cases were evaluated. For each, a predetermined musical phrase was recorded into a dynamic microphone, and for each window the 'heard notes' and 'perceived pitches' were recorded (NB: 'heard notes' is synonymous with the input to the classifier, and 'perceived pitches' are the candidates selected as fundamentals). Then, each note in the heard notes was manually categorised as one of the four possibilities (TP, FP, TN or FN). A weighted mean (weighted by the total notes heard) was calculated across 3 repeats, and a number of metrics were calculated for each repeat and averaged across all three. The standard deviation for each averaged metric was also calculated as an indicator for the spread (and further, likely reliability) of the results.

Firstly the True Positive Rate (TPR), False Positive Rate (FPR), True Negative Rate (TNR) and False Negative Rate (FNR) were calculated. As well as this, the Precision (PR), Recall (RE), Specificity (SP), Accuracy (A), and F-Score (F) were determined.

Further, it proves imperative to also analyse the position of the classifier in ROC space (that is, a plot of sensitivity against inverse specificity) [6]. Specificity is an important

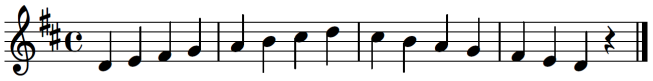

Figure 4. A D Major scale from D4, ascending and descending - the first test case

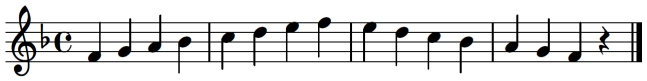

Figure 5. An F Major scale from F4, ascending and descending - the second test case

characteristic of the classifier in this case as it is important to minimise the number of False Positives in the output as it is preferable to lose harmonic content (i.e. due to a False Negative) as opposed to gaining inharmonic content. This is because the resulting notes will more accurately represent the actual notes being played, partly due to overlapping harmonics in the original. Consider, for example, the chord of C7 (C, E, G, B♭) - it is better for a harmonically-related note to be lost (Eg. an output of C, G, B♭) than for a harmonically-unrelated (in relation to the chord) note to be added (Eg. an output of C, E, G, B♭, D) as the harmonic structure of the chord is maintained in the former but not the latter.

The perfect classifier lies at (0, 1) in ROC space (i.e. 100% sensitivity and 100% specificity). In order to derive an overall score that also takes specificity into account, the distance from the classifier to the perfect classifier in ROC space is used. The shorter this distance, the better the classifier is, thus an overall score is calculated for the classifier equal to $\frac{FScore}{distance}$.

The first two test cases are based on monophonic phrases of music. It is to be expected that the classifier will perform well on these, but due to the presence of harmonics, it is unlikely to be perfect. Likely errors include octave errors, where a note is classified as the correct pitch chroma but incorrect pitch height, and errors where aspects of the noise due to an imperfect recording are incorrectly classified as perceived pitches (false positive cases).

The other three cases are based on more complex, polyphonic phrases on a variety of instruments. These are the more challenging tests, and thus, the performance is expected to be much lower than that on the monophonic cases. This increase in difficulty can be attributed both to the instruments being used (piano, guitar, and melodica) as well as the additional challenges that the more complex phrases present, such as overlapping harmonics and fast transitions between notes.

The first test case is an ascending and descending D major scale over one octave on an acoustic flute. Because of its relatively pure sound, it would be expected for the classifier to perform well on the flute.

This test case is essentially identical to that of test case 1, but played on a piano as opposed to a flute. Moreover, the aim is to ascertain that the approach is capable of performing over a variety of instruments (and therefore harmonic patterns). It also serves as a valuable comparison to test case 3 (polyphonic

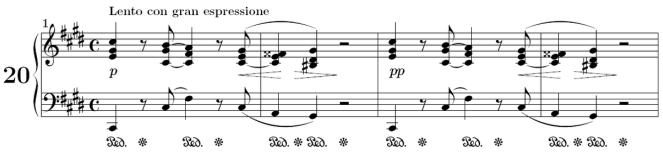

Figure 6. The opening bars of Chopin's Nocturne Lento Con Gran Espressione - the third test case [8]

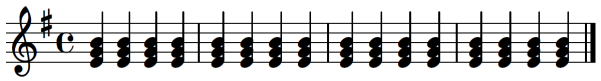

Figure 7. Repeated E minor triads in root position - the fourth test case

piano), showing the difference in performance between the monophonic and polyphonic cases.

A more complex test case, test case 3 has a number of pitfalls and edge cases that a perfect classifier would be able to overcome. Furthermore, the overlap in harmonic content, particularly between the left and right hands of the music, make it incredibly difficult to distinguish between the fundamentals present and their harmonics. As well as this, the introduction of pedal is likely to cause issues with held notes being classified or otherwise interfering with the harmonic content.

Whilst the content of the phrases of the final two cases may appear simple, the resulting harmonic overlap and difficult harmonic structure of the instruments (guitar and melodica respectively) result in a challenging test case overall. It is interesting to compare the capability of the classifier to deal with a variety of instruments, especially ones such as melodica - the tone of which results in a great range of partials that are often not harmonically related to the fundamental itself.

IV. α Determination

α is the scalar value used with the mean amplitude of the heard notes (μ) during each pass of the loop in order to discard harmonics under a certain amplitude. These are regarded as noise and/or harmonics and considered irrelevant to the remaining harmonic structure observed.

It is imperative that an optimal value of α is found in order to obtain the best results from the classifier. Hence, a range of α values were tested, and the best (on average) over the five test cases was used. For each value of α to be tested, an experiment was conducted for each of the five test cases, and the results taken across three runs.

If the value of α is too high, the false negative rate will increase as the classifier will begin to discard relevant

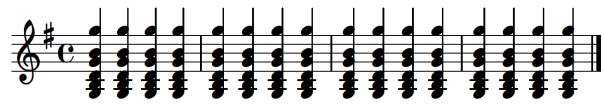

Figure 8. Repeated G minor chords with a tonic root - the fifth test case

candidates. Conversely if the α value is too low, the false positive rate will increase as the classifier will retain and classify candidates that should have been discarded. Moreover, an optimal value of α is essentially a compromise between sensitivity and specificity.

The results for each test case with each value of α can be found in Appendix I.

Following the experimental results, a ROC space graph was plotted for each test case with the classifier for each value of α present. The corresponding distances to (0, 1) were then calculated. See Appendix I for the relevant graphs and tables.

A. Overall Scores

Table I
SCORES PER α VALUE

α Value	Test 1	Test 2	Test 3	Test 4	Test 5	μ
3.00	2.68	6.62	2.05	1.50	3.12	3.19
4.00	4.20	7.64	1.38	1.51	1.52	3.25
4.25	4.56	21.26	2.25	1.53	3.63	6.64
4.50	3.49	21.71	1.48	1.24	2.08	6.00
5.00	2.70	22.71	1.86	1.12	3.06	6.29

NB: The scores shown are (F-Score ÷ Distance)

Given the overall scores for each value of α, a value of 4.25 is chosen for use as it has the highest average score (6.64).

V. EVALUATION

Table II
METRICS PER TEST CASE

	PR	RE	SP	A	F
Test 1	78.09%	93.89%	83.00%	86.89%	84.78%
Test 2	100.00%	97.22%	100.00%	97.85%	98.55%
Test 3	82.46%	75.67%	77.52%	76.36%	78.85%
Test 4	86.31%	82.50%	53.33%	75.83%	83.47%
Test 5	65.08%	100.00%	64.29%	79.07%	78.47%
$\boldsymbol{\mu}$	82.39%	89.86%	75.63%	83.20%	84.82%

The above table shows the scores for each test case with a value of $\alpha = 4.25$. An average for each is taken across the five cases.

As can be seen from the results, the classifier performed extremely well overall on the given test cases. As the test cases used are not the same as those that other papers have used for evaluation however, a comparison between the raw scores is likely to be inconsequential.

A. σ Analysis

In order to ascertain the robustness of the evaluation, it is important to analyse the standard deviations (σ) of the results in Appendix B. A low σ with respect to the mean μ signifies a stable test, and conversely, a high σ with respect to μ suggests that the test is relatively volatile or unstable (i.e. there is significant variation in the results).

A widely-used method for determining whether a given σ is high or low is the use of the coefficient of variation (C_v), which is defined as the ratio between the standard deviation and the mean, that is, $C_v = \frac{\sigma}{\mu}$ [5].

In general, if a given $C_v < 1$, the variation is considered to be relatively low, whereas a $C_v \geq 1$ is indicative of a relatively high variation.

Table III
C_v TOTALS OVER ALL METRICS

$C_v \geq 1$	$1 > C_v \geq 0.9$	$0.9 > C_v$
9	7	209

In the entirety of Appendix B, 225 individual metrics are presented, each with a respective μ and σ. The above table shows the total number of metrics for which the variation (i.e. the C_v) is high (> 1), close to high (≥ 0.9) and low (< 0.9). As can be seen, the majority of the metrics have what can be considered a low variation, whereas only a small proportion have a high or close to high variation. As a result, it can be concluded that the vast majority of the tests can be considered stable.

VI. IMPROVEMENTS & FUTURE WORK

As well as a number of potential improvements to the proposed approach, there is also a host of interesting and thought-provoking further areas that may prove insightful to further research into. It is important to highlight both these improvements as well as further questions in order to incite further probing into these areas.

Firstly, in future work it will be imperative to adapt the implementation of the system to allow for automated testing on much larger sets of data, both to improve the reliability of the results and allow for meaningful comparison to other related works - for example, the MAPS dataset [4] which is used by multiple approaches, including an implementation based on an end-to-end neural network [14]. Moreover, testing should be extended to a wider variety of instruments to ensure that the approach is robust.

Currently the 'Notes Reduce' function only accounts for instruments for which the amplitude of the $(n+1)^{th}$ harmonic has an amplitude that is strictly less than that of the n^{th} harmonic. As this is not the case for all instruments (with one notable counterexample being the trumpet), the current approach certainly makes mistakes with these instruments. Moreover, it would be interesting to implement a tolerance to the 'Notes Reduce' function whereby the harmonics need not be strictly monotonically decreasing but instead within some range (both up or down) from the amplitude of the previous harmonic.

As well as this, it may be possible to analyse patterns in the harmonic spectra of various acoustic instruments and then utilise this knowledge to ascertain what instruments are present in a given signal, or at least deduce a probabilistic model of the likely combinations of instruments. This could further be used to adapt the algorithm on the fly to change

aspects such as the α value and 'Notes Reduce' threshold dependent on the instruments.

Another area of interest would be looking into the possibility of abusing the properties of stereo recordings in order to improve the algorithm by modelling the sound as a three-dimensional representation in which the instruments (or sections of instruments) can be separated dependent on their position in three-dimensional space.

Other factors that affect the algorithm such as noise levels, microphone type and quality, and room type could also be experimented with to see if there is a noticeable effect on the performance of the algorithm. Moreover, it would prove insightful to ascertain the effects specifically on the optimal α value and 'Notes Reduce' threshold, if any exist.

There are also a few further shortcomings of the current approach in which specific edge-cases are not accounted for. Take, for example, the case in which two or more played notes have a number of overlapping harmonics, but also a number of higher-frequency non-overlapping harmonics. If the algorithm completely removes the overlapping harmonics from the model on the first pass (i.e. on the first fundamental), the higher-frequency harmonics will no longer be monotonically decreasing (and potentially not even within a threshold-based range) from the previous harmonic and will therefore remain untouched by the second pass of the algorithm (i.e. the pass for the second fundamental) and so on. This case results in a number of false positives as the high-frequency non-overlapping harmonics are then likely selected as fundamentals too. One potential solution to this would be to reduce the harmonics proportionally to the fundamentals based on the instrument being played (if reproducible patterns are indeed found in the harmonic spectra) rather than based solely on the spline method that is currently employed.

VII. Conclusion

This paper has outlined an effective method for the pitch analysis of polyphonic (and monophonic) acoustic musical signals. The proposed novel algorithmic approach (II) employs a 'raking' algorithm in the frequency domain (following minimal preprocessing), and if well-implemented, both runs in real-time with respect to the average time taken for humans to react to an auditory stimulus, and exhibits very good performance.

Though there are a number of clear downsides to the approach, they can likely be rectified through further research and the suggested extensions (VI). Despite these, however, the obtained results from the approach are positive, and given more rigorous testing on a communal dataset such as the MAPS dataset, it would certainly be insightful to make quantitative comparisons to the approaches outlined in the related work (I) amongst others. Moreover, by automating testing over larger sets of data, the accuracy and reliability of the results will be increased.

In conclusion, though improvements can be made to the algorithm, the research presented in this report provides a solid basis for further insightful investigation into the area, as well as a different viewpoint on the analysis of musical signals as a whole. Furthermore, it is certainly a step forward in the finding of a solution to this as of yet unsolved problem in computer science.

References

[1] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello. Deep salience representations for f_0 estimation in polyphonic music. In *Proceedings of the 18th International Society for Music Information Retrieval Conference, Suzhou, China*, pages 23–27, 2017.

[2] A. De Cheveigné and H. Kawahara. Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.

[3] P. De La Cuadra, A. S. Master, and C. Sapp. Efficient pitch detection techniques for interactive music. In *ICMC*, 2001.

[4] V. Emiya, R. Badeau, and B. David. Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1643–1654, 2010.

[5] B. Everitt. *The Cambridge dictionary of statistics / B.S. Everitt*. Cambridge University Press Cambridge, U.K. ; New York, 2nd ed. edition, 2002.

[6] T. Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[7] Y. Li and D. Wang. Pitch detection in polyphonic music using instrument tone models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, volume 2, pages II–481–II–484, April 2007.

[8] I. M. Library. Nocturne in c-sharp minor, b.49 (chopin, frédéric). [Online; accessed August 11, 2018].

[9] P. McLeod. Fast, accurate pitch detection tools for music analysis. *Academisch proefschrift, University of Otago. Department of Computer Science*, 2009.

[10] I. Morley. *An Investigation into the Prehistory of Human Musical Capacities and Behaviours, Using Archaeological, Anthropological, Cognitive and Behavioural Evidence*. PhD thesis, 2003.

[11] M. A. Noll. Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum, and a maximum likelihood estimate. In *Symposium on Computer Processing in Communication, ed.*, volume 19, pages 779–797. University of Brooklyn Press, New York, 1969.

[12] X. Rodet and B. Doval. Fundamental frequency estimation using a new harmonic matching method. In *Proceedings of the International Computer Music Conference*, pages 555–555. INTERNATIONAL COMPUTER MUSIC ACCOCIATION, 1991.

[13] J. Shelton and G. P. Kumar. Comparison between auditory and visual simple reaction times. *Neuroscience & Medicine*, 1(1):30–32, 2010.

[14] S. Sigtia, E. Benetos, and S. Dixon. An end-to-end neural network for polyphonic piano music transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(5):927–939, May 2016.

APPENDIX A
ROC SPACE DISTANCES

Appendix A contains the ROC-space plots for each test case (see Section III), for each α value (3.00, 4.00, 4.25, 4.50, and 5.00). These are used to determine a good value for α, which is used in the proposed approach (II) as a minimum amplitude for notes to be considered as candidate $f_0 s$.

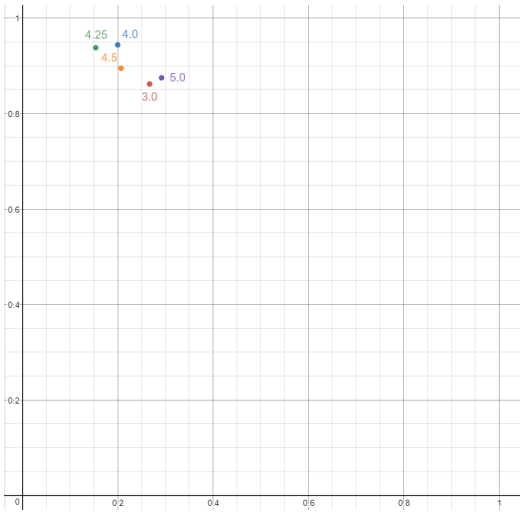

Figure 9. ROC space graph for test case 1

Table IV
TEST CASE 1 - ROC DISTANCES

α Value	Distance to (0,1)
3.00	0.301
4.00	0.208
4.25	0.166
4.50	0.232
5.00	0.318

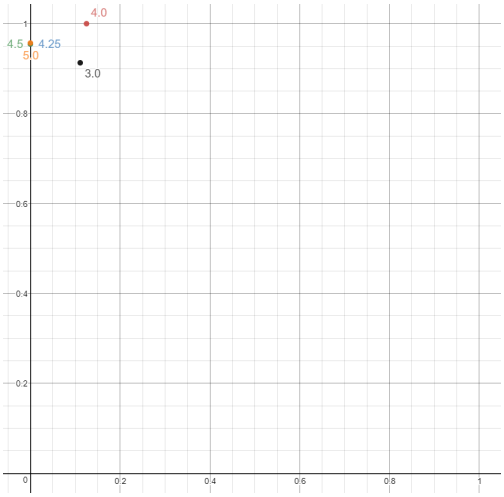

Figure 10. ROC space graph for test case 2

Table V
TEST CASE 2 - ROC DISTANCES

α Value	Distance to (0,1)
3.00	0.141
4.00	0.125
4.25	0.045
4.50	0.045
5.00	0.043

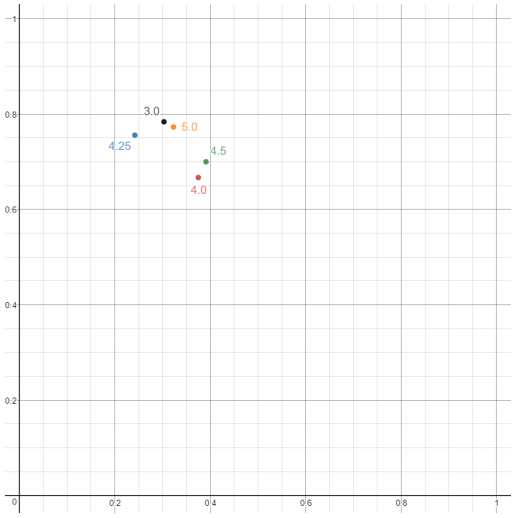

Figure 11. ROC space graph for test case 3

Table VI
TEST CASE 3 - ROC DISTANCES

α Value	Distance to (0,1)
3.00	0.372
4.00	0.502
4.25	0.344
4.50	0.493
5.00	0.395

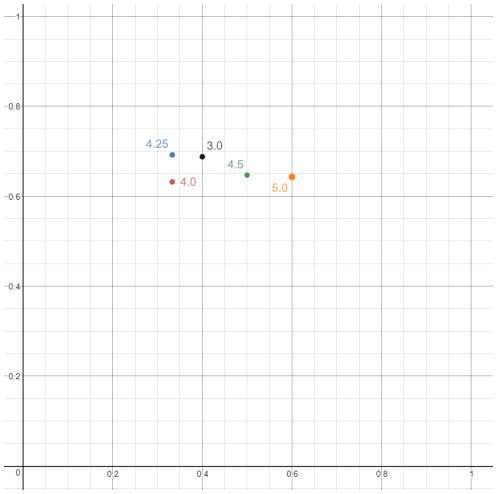

Figure 12. ROC space graph for test case 4

Table VII
TEST CASE 4 - ROC DISTANCES

α Value	Distance to (0,1)
3.00	0.507
4.00	0.496
4.25	0.454
4.50	0.612
5.00	0.698

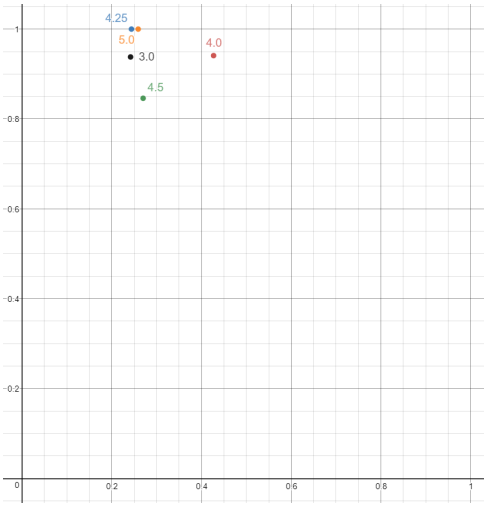

Figure 13. ROC space graph for test case 5

Table VIII
TEST CASE 5 - ROC DISTANCES

α Value	Distance to (0,1)
3.00	0.250
4.00	0.431
4.25	0.244
4.50	0.311
5.00	0.259

APPENDIX B
TEST DATA

Appendix B contains the following metrics, measured for each test case and α value, and averaged across 3 runs,

- True Positive Rate (TPR)
- False Positive Rate (FPR)
- True Negative Rate (TNR)
- False Negative Rate (FNR)
- Precision (PR)
- Recall (RE)
- Specificity (SP)
- Accuracy (A)
- F-Score (F)

the mean, μ, and standard deviation, σ, are also presented for each.

Because of the sheer volume of them, only the tables for $\alpha = 4.25$ are included. The full

appendix (and thus the remainder of the tables) can be found at http://tomg.io/appendix-b.pdf and https://github.com/TauOmicronMu/tauomicronmu.github.io/blob/master/appendix-b.pdf

A. Test Case 1

Table IX
$\alpha = 4.25$

	TPR	FPR	TNR	FNR
1	35.00%	17.50%	47.50%	0.00%
2	35.85%	5.66%	56.60%	1.89%
3	37.14%	8.57%	48.57%	5.71%
μ	36.00%	10.58%	50.89%	2.53%
σ	1.08%	6.17%	4.98%	2.91%

	PR	RE	SP	A	F
1	66.67%	100.00%	73.08%	82.50%	80.00%
2	86.36%	95.00%	90.91%	92.45%	90.48%
3	81.25%	86.67%	85.00%	85.71%	83.87%
μ	78.09%	93.89%	83.00%	86.89%	84.78%
σ	10.22%	6.74%	9.08%	5.08%	5.30%

B. Test Case 2

Table X
$\alpha = 4.25$

	TPR	FPR	TNR	FNR
1	70.97%	0.00%	22.58%	6.45%
2	70.00%	0.00%	30.00%	0.00%
3	65.52%	0.00%	34.48%	0.00%
μ	68.83%	0.00%	29.02%	2.15%
σ	2.91%	0.00%	6.01%	3.72%

	PR	RE	SP	A	F
1	100.00%	91.67%	100.00%	93.55%	95.65%
2	100.00%	100.00%	100.00%	100.00%	100.00%
3	100.00%	100.00%	100.00%	100.00%	100.00%
μ	100.00%	97.22%	100.00%	97.85%	98.55%
σ	0.00%	4.81%	0.00%	3.72%	2.51%

C. Test Case 3

Table XI
$\alpha = 4.25$

	TPR	FPR	TNR	FNR
1	41.77%	8.86%	32.91%	16.46%
2	48.72%	5.13%	34.62%	11.54%
3	42.11%	14.47%	28.95%	14.47%
μ	44.20%	9.49%	32.16%	14.16%
σ	3.92%	4.70%	2.91%	2.47%

	PR	RE	SP	A	F
1	82.50%	71.74%	78.79%	74.68%	76.74%
2	90.48%	80.85%	87.10%	83.33%	85.39%
3	74.42%	74.42%	66.67%	71.05%	74.42%
μ	82.46%	75.67%	77.52%	76.36%	78.85%
σ	8.03%	4.68%	10.27%	6.31%	5.78%

D. Test Case 4

Table XII
$\alpha = 4.25$

	TPR	FPR	TNR	FNR
1	75.00%	12.50%	12.50%	0.00%
2	48.00%	8.00%	12.00%	32.00%
3	70.00%	10.00%	10.00%	10.00%
μ	64.33%	10.17%	11.50%	14.00%
σ	14.36%	2.25%	1.32%	16.37%

	PR	RE	SP	A	F
1	85.71%	100.00%	50.00%	87.50%	92.31%
2	85.71%	60.00%	60.00%	60.00%	70.59%
3	87.50%	87.50%	50.00%	80.00%	87.50%
μ	86.31%	82.50%	53.33%	75.83%	83.47%
σ	1.03%	20.46%	5.77%	14.22%	11.41%

E. Test Case 5

Table XIII
$\alpha = 4.25$

	TPR	FPR	TNR	FNR
1	27.03%	22.97%	50.00%	0.00%
2	43.48%	23.91%	32.61%	0.00%
3	52.27%	15.91%	31.82%	0.00%
μ	40.93%	20.93%	38.14%	0.00%
σ	12.81%	4.38%	10.28%	0.00%

	PR	RE	SP	A	F
1	54.05%	100.00%	68.52%	77.03%	70.18%
2	64.52%	100.00%	57.69%	76.09%	78.43%
3	76.67%	100.00%	66.67%	84.09%	86.79%
μ	65.08%	100.00%	64.29%	79.07%	78.47%
σ	11.32%	0.00%	5.79%	4.38%	8.31%