# Music disassembly,

Lippay Balazs, Babeș-Bolyai University of Science, Romania, Informatics

# Abscract

In this paper we will explore the building blocks of sound and music, then we will try to reduce actual sound into these atomic parts. We will construct a mathematical formula of identifying played musical notes, including chords. Observe overcomable limitations and the cost of their solutions. Overall walk through a theoretical way of perfect identification that could even lead to live-generatable sheet music in the future.

# Classification

AMS 41-04 Approximations and expansions

ACM I.5.0 Computing Methodologies Pattern recognition

# Introduction

The disassembly of music, the identification of pitch and instant playback capability is a skill that comes with years of hard work and playing experience on an instrument. Expert musicians can identify music just by ear and play it back with a high accuracy rate on their instruments.

What if we had this skill when we first picked up an instrument as a complete beginner? How would it effect our learning curve if we could get instant feedback on our performance?

The obvious answer to the above question is that it would be extremely beneficial, and the with fact that humans can do it, there is a possibility that a machine could be able to as well. Sadly there are very few such software and next to none that use open source methods, and a pattern in them is the specialization on a single instrument.

The aim of this paper is to identify a method, and as a challenge, a non-learning-based one, of live analysis of played music, through a generic non-professional microphone as input; and explore the viability, efficiency and possible use cases of such software.

Below we will explore the the possibility of an algorithm used on machinery fault detection, music theory on pitch, chords and the relation between musical notes, while constructing our mathematical aproach, then tweaking it for increased accuracy.

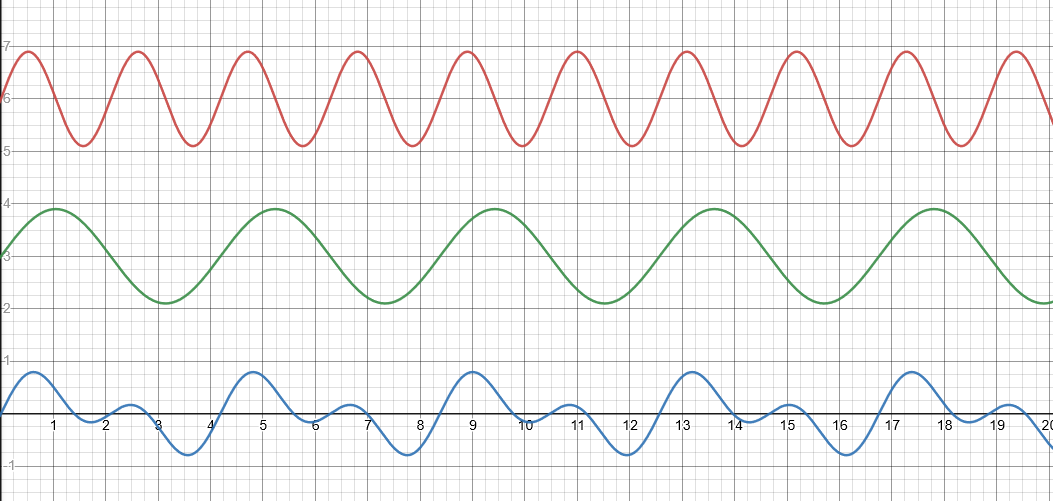
# Methodology description

### The deffinition of sound and music

To start off, we need to define what music is.

Music is sound in patterns. Sound is the vibration of air. The sound that we hear is the summation of all the vibrations around us, be it loud or faint, it is there.

Although it is not human-readable, sound actually follows a pattern, and to identify this pattern, we need to look at how sound is formed from bottom up. The following image shows two different wave patterns, followed by their summation.



If we continue to repeat this operation infinitely many times, for every single frequency, we will get actual sound.

Sound as its base, has every frequency inside it, the only defining factor is their individual power, the distance the wave swings away from the origin.

After sound, we define pitch. Pitch is a constant, powerful waveform of a set frequency mutually agreed upon and named. For example the frequently used reference point, 440 Hz, the A4 note. From A4, we double the frequency to 880 Hz, and we get A5. In between A4 and A5, we split the frequencies into equal intervals, such as A4 to be 0, and A5 to be 12. The points between them are the other notes.



As to why we have 8 letters instead of 12, and the numbering changing on C, that is a question for middle-age European music theory experts.

Mentioning chords, although they are extremely complicated from a music theory perspective, for us, they are the action of playing multiple notes at the same time, thus producing both frequencies simultaneously.

### The mathematical approach

Now that we know what sound is, and that it is a patern, we need to look into how we can decode it. Here is where we use the Fast Fourier Transform algorithm, which is an algorithm that separates a complex polinomial into the multiplication of much simpler atomic polinomials, and as it turns out, it can do much more.

FFT can separate a snapshot of a sound input into its building blocks, the individual waveforms, and by transforming this output, we can observe the power, or loudness, of frequencies. We do this with the following method:

* We assume that T is a vector containing the waveform of a recorded sound interval of length CHUNK with an input rate of RATE.
* We apply Fast Fourier Transformation (FFT) on T, thus we get a vector of complex numbers C.
* We calculate the absolute value of the elements of C, and we get a vector of real numbers P.
* The indexes of P are the separate sound waves, and their values are the loudness of that wave.
* We define M as a vector of the indexes of peaks in P.
* For M we calculate H = | M / CHUNK \* RATE |, which is a vector of the ringing frequencies in the sound chunk.

All in one:

* H = abs(peaks(noise\_gate(X, abs(fft(T)))) / CHUNK \* RATE)

### Implementation

Now that we have a solid mathematical approach, we can implement it, for this I chose python for is popularity and ease of use.

By following the algorithm, after implementation and plotting the whole frequency array, we get the following on a single note:



Although patterns are extremely slightly visible, our peak identification function can not deal with such data, thus a reduction was necessary. For this, division by the bit-rate, and multiplication with the chunk size seems to serve as a usable general audio strength reducer. After this, we get:



And thus, we have a crystal clear pattern, but, as we can observe, we played a note and we see many.

### Instruments

The explanation to the above anomaly is in the nature of instruments. The above graph shows a guitar sound.

With instruments, what gives each and every one of them their unique tone lies in our graph: “resonance and harmonics”.

When an instrument gives off a sound, it does not only generate the corresponding frequency, but resonates upwards on the octaves as well.

Observing such a tendency, we can construct a filter. Taking as example the above graph, if we take the first found peak, let us call it P1, we subtract ~2xP1 from P2, ~1/4xP1 from P3 and P4, maybe 1/8xP1 from P5, we will get roughly a single peak on the correct root frequency. With this if we play multiple notes, even octaves, in a perfect world, we should be able to identify all played notes with perfect accuracy.

# Experimental validation

We will conduct two experiments:

* Accuracy test of our algorithm
* Viability of an instrument filter for perfect note identification

## Acuracy test

Instrument: Baritone Ukulele

### Procedure

Run application

Run tuner

Tune instrument

Do:

Pick note

Play note

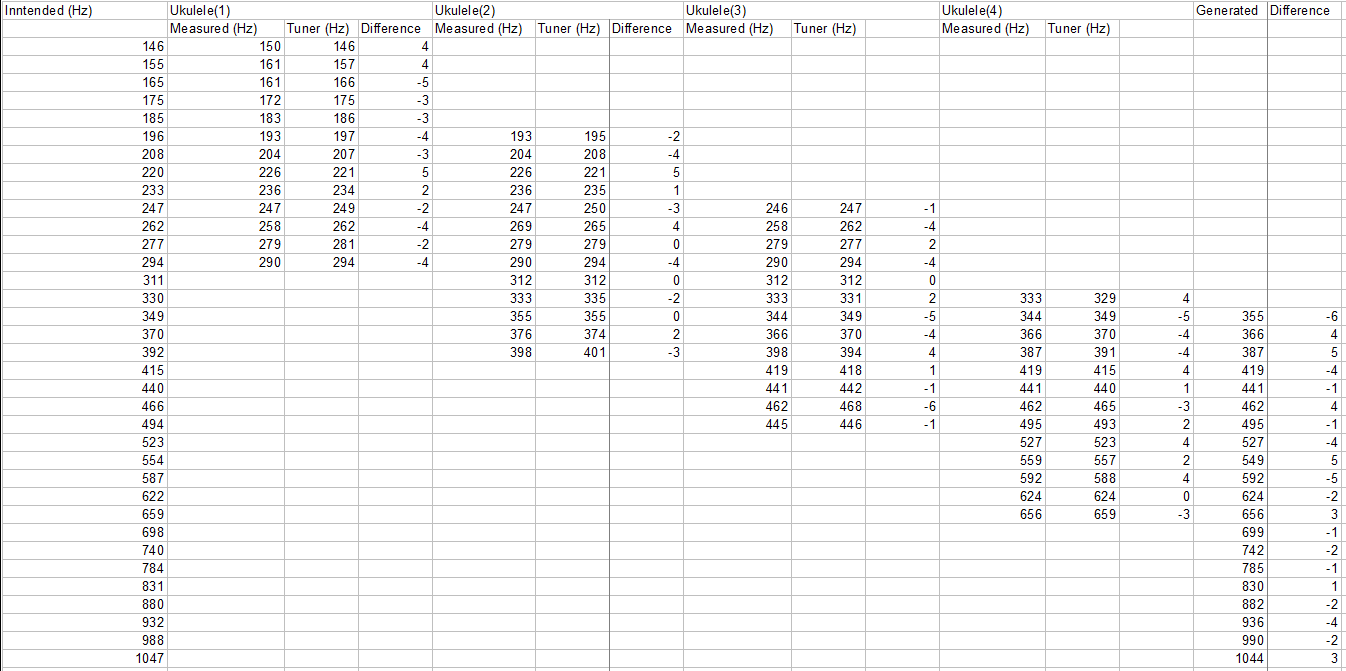
Pick other note

Play other note

Observe is they show the same pattern

Repeat

### Results



### Conclusion

We can accurately determine the played note with an error margin of 6 Hz, which becomes increasingly more significant on the lower end, since notes are closer together. This is due to the output going up to \44kHz, not to around 8kHz where the human hearing caps out. This can be fixed with custom implementions of the FFT function.

## Viability of an instrument filter for perfect note identification

Instrument: Baritone Ukulele

### Procedure

Run application

Run tuner

Tune instrument

Do:

Pick note

Play note

Observe application output

Observe down tuner output

Calculate difference

Repeat

### Results



The above graphs are of the top one played on the thirds string and bottom one on the second string, due to them being made from a different material, their patterns differ

### Conclusion

The experiment resulted in failure when the two string were not showing the same results, thus we state that a simple filter is not enough to get perfect data. It is still possible to create a filter, but it needs to be significantly more complex and will involve logic.

Although the above shown data is perfectly good as input for a learning algorithm, and probably will lead to perfect results, in a purely mathematical way, it is not possible to have a general frequency filter over the whole instrument.

# Results and conclusions

### General experiment results

In conclusion, answering the questions we put up in the introduction. It is entirely possible to disassemble music universally with instant feedback, with a couple limitations.

Identifying played music can be done when playing single notes perfectly, on any instrument.

* lowest sounding frequency is always the root note
* easy identification algorithm

Identifying chords can be done, not counting octaves (since they are masked by instrument resonance), on any instrument

* notes are acquired by filtering out multiplications of the root note by powers of two
* complex identification algorithm

Identifying chords, with octaves, can be done, with pre-built specialized complex filter.

* every result is a played note
* very complex identification algorithm

### Possible use cases

Although decoding complete rock music is most likely out of the realm of possibilities, we can guess with relatively high accuracy what a single instrument is playing at any given time. With this, if we would encode a song, and show the notes on the screen at the correct time, we can tell if a person follows our instructions and plays the notes correctly. With an instrument independent solution, this can be done with only checking if the notes are played or not, and can not accurately determine if extra notes are played alongside it, but focusing on a single instrument, with high effort, this can be done, thus presenting a harder challenge.

With a high effort, specialized algorithm heavily toned for a single instrument. If we pair our pitch detection with a beat detecting algorithm, we could actually produce sheet music from live play.

# Source code

<https://github.com/lippaybalazs/ResearchProject>

# References

[1] The Fast Fourier Transform - Ulrich Oberst - University of Innsbruck - January 2007

<https://www.researchgate.net/publication/220259110_The_Fast_Fourier_Transform>

[2] Bearing vibration detection and analysis using enhanced fast Fourier transform algorithm -

Hsiung-Cheng Lin, Yu-Chen Ye, Bo-Jyun Huang - October 2016

<https://journals.sagepub.com/doi/full/10.1177/1687814016675080>

[3] Audio Analysis using the Discrete Wavelet Transform - George Tzanetakis, Georg Essl, Perry Cook

<https://soundlab.cs.princeton.edu/publications/2001_amta_aadwt.pdf>

[4] Simultaneous Estimation of Chords and Musical Context From Audio - Matthias Mauch; Simon Dixon

<https://ieeexplore.ieee.org/document/5256327>

[5] Vibration parameter estimation methods for ultrasonic measurement systems - Kshetrimayum Milan Singh, Parasuraman Sumathi - July 2015

<https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-smt.2014.0176>

[6] Continuous wavelet transform for non-stationary vibration detection with phase-OTDR - Zengguang Qin, Liang Chen, and Xiaoyi Bao

<https://www.osapublishing.org/oe/fulltext.cfm?uri=oe-20-18-20459&id=240860>

[7] Application of Wavelet Transform and its Advantages Compared to Fourier Transform - M. Sifuzzaman, M.R. Islam and M.Z. Ali - September 27, 2009

<http://inet.vidyasagar.ac.in:8080/jspui/bitstream/123456789/779/2/Art11.pdf>

[8] Separation of adjacent interharmonics using maximum energy retrieving algorithm - Hsiung Cheng Lin - March 2016

<https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-smt.2015.0115>

[9] A Graphical Model for Recognizing Sung Melodies - Christopher Raphael

<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.137.1147&rep=rep1&type=pdf>

[10] AUTOMATED SYNCHRONIZATION OF SCANNED SHEET MUSIC WITH AUDIO RECORDINGS - Frank Kurth, Meinard M̈uller, Christian Fremerey, Yoon-ha Chang, and Michael Clause

<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.205.8799&rep=rep1&type=pdf>

[11] MIREX AUDIO CHORD DETECTION - Maksim Khadkevich, Maurizio Omologo

<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.334.8317&rep=rep1&type=pdf>

[12] Large-Scale Study of Chord Estimation Algorithms Based on Chroma Representation and HMM -Helene Papadopoulos; Geoffroy Peeters

<https://ieeexplore.ieee.org/document/4275055>

[13] Simultaneous estimation of chord progression and downbeats from an audio file - Helene Papadopoulos; Geoffroy Peeters

<https://ieeexplore.ieee.org/document/4517561>

[14] Influences of Signal Processing, Tone Profiles, and Chord Progressions on a Model for Estimating the Musical Key from Audio - Katy Noland; Mark Sandler

<https://ieeexplore.ieee.org/document/6791997>

[15] Pitch Perception and Measurement - Roger Shepard

<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.208.2386&rep=rep1&type=pdf>