

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/261024961>

Real-time guitar string detection for music education software

Conference Paper · July 2013

DOI: 10.1109/WIAMIS.2013.6616120

CITATIONS

12

READS

3,689

3 authors, including:



Christian Dittmar

Friedrich-Alexander-University of Erlangen-Nürnberg

83 PUBLICATIONS 966 CITATIONS

[SEE PROFILE](#)



Jakob Abeßer

Fraunhofer Institute for Digital Media Technology IDMT

95 PUBLICATIONS 932 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Informed Sound Activity Detection in Music Recordings (ISAD) [View project](#)



Music Tagging using Deep Neural Networks [View project](#)

REAL-TIME GUITAR STRING DETECTION FOR MUSIC EDUCATION SOFTWARE

Christian Dittmar, Andreas Männchen and Jakob Abeßer

Fraunhofer Institute for Digital Media Technology
Ilmenau, Germany
{dmr, maenas, abr}@idmt.fraunhofer.de

ABSTRACT

In this paper, we present a novel approach to real-time detection of the string number and fretboard position from polyphonic guitar recordings. Our goal is to assess, if a music student is correctly performing guitar exercises presented via music education software or a remote guitar teacher. We combine a state-of-the-art approach for multi-pitch detection with a subsequent audio feature extraction and classification stage. Performance of the proposed system is evaluated with manually annotated chords recorded using different guitars.

Index Terms—multiple fundamental frequency estimation, string classification, fretboard position, fingering, electric guitar, inharmonicity coefficient, tablature

1. INTRODUCTION

Over the past few years, there have been several new software products that promise to enhance learning a musical instrument by means of assistive software. Acoustic or electric guitars are often the main supported instrument due to their popularity among starters. On the one hand, there are gaming oriented applications such as RockSmith¹, RockProdigy², GuitarBots³, Songs2See⁴, Jamstar⁵, and Bandfuse⁶. They rely on game mechanics inspired by popular music video games such as GuitarHero⁷ or RockBand⁸. That means, the user plays along to a song while playing instructions are presented on screen in sync to music. In contrast to GuitarHero and RockBand, the above mentioned applications allow the user to play real acoustic or electric guitars instead of guitar-shaped plastic controllers. The opposite is done in applications such as FretLight⁹ or GTar¹⁰. Here, the user has to play along to a given song or short exercise on a modified guitar with LEDs lighting up the sequence of strings and frets that are to be played. In the case of FretLight, the playing instructions are usually generated in real-time by a remote guitar teacher. Moreover, there are online services like SoundSlice¹¹ and YouTab¹² that enable users to create and publish so called guitar tabs synced with the corresponding song or video. Guitar tabs are short for tablature notation, which is probably the most popular music notation

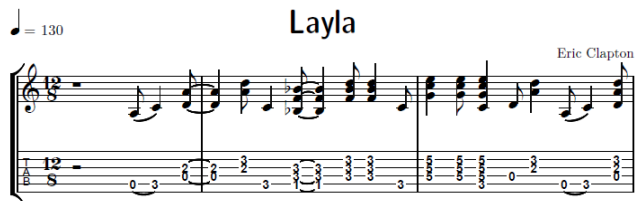


Fig. 1. Score and tablature representation of the first bars from Eric Clapton's Layla. The six horizontal lines in the tablature correspond to the six strings with the tuning E2, A2, D3, G3, B3, E4 (from bottom to top). The numbers correspond to the fret numbers on the strings that are to be played.

format among guitarists. The tablature is based on the geometry of fretted string instruments. It specifies the string and fret number for each note and thus resolves the ambiguity between note pitch and fretboard position. The guitarist can immediately see where he has to place his fingers on the strings and frets (so called fingering). Figure 1 illustrates the correspondence between common music notation and tablature for the guitar. Large amounts of guitar tabs are already available on the internet. However, since most of them are created by amateurs, these tablatures are often incomplete and erroneous. The first application scenario for the method presented here is the semi-automatic extraction of tablatures by means of automatic transcription and string detection from guitar recordings. Secondly, for a remote guitar teacher, it would be helpful to have his playing on a real instrument translated in real-time to corresponding playing instructions for the student. The third application scenario is to assess, if the guitar student is truly following the given playing instructions by means of real-time detection of played notes and strings.

2. GOALS & CHALLENGES

As introduced in Section 1, our goal is to estimate the notes and string number n_s from monophonic and polyphonic (monotimbral) recordings of electric guitars. Based on the note pitch and the string number, we can apply knowledge about the instrument tuning in order to derive the fret number n_f . The first challenge is to get a good real-time estimate of the notes being played. Since we focus on the electric guitar, frequencies between 82.40 Hz and 659.25 Hz need to be investigated as potential f_0 -candidates¹³. The second challenge is the selection of suitable audio features that allow to discriminate between different strings. The automatic classification of the played string poses a challenging task since the physical differences

¹³This corresponds to the most common guitar string tuning E2 to E4, and a fret range up to the 12th fret.

¹<http://rocksmith.ubi.com/>

²<http://www.rockprodigy.com/>

³<http://ovelin.com/guitarbots/>

⁴<http://www.songs2see.com/>

⁵<http://jamstar.co/>

⁶<http://bandfuse.com/>

⁷<http://hub.guitarhero.com/>

⁸<http://www.rockband.com/>

⁹<http://www.fretlight.com/>

¹⁰<http://www.incidentgtar.com/>

¹¹<http://www.soundslice.com/>

¹²<http://www.you-tab.com/>

between strings alter the sonic properties of played notes only very slightly.

3. RELATED WORK

An in-depth overview of different approaches for estimation of string and fretboard position from guitar recordings can be found in [1]. Apart from audio analysis, methods that incorporate computer vision techniques, instrument enhancements, and sensors are also discussed there. Here, we solely focus on the publications that are most related to our approach. Already in [2], Barbancho et al. presented an algorithm to estimate the string number from isolated notes in guitar recordings. Their method employs a number of spectral and temporal audio features for each note. In [3], Barbancho et al. presented a system for polyphonic transcription of guitar chords, which also allows to estimate the fingering of the chords. The authors investigated 330 different fingering configuration for the most common three-voiced and four-voiced guitar chords. A Hidden Markov Model (HMM) is used to model all fingering configurations as individual hidden states. Based on an existing multi-pitch estimation algorithm, harmonic salience values are computed for all possible pitch candidates within the pitch range of the guitar. Subsequently, these salience values are used as observations for the HMM. The transitions between different hidden states are furthermore constrained by musicological and physiological models. The authors report a recognition rate of 88 % of correctly identified chord and chord fingering.

In these publications, inharmonicity of the guitar strings [4] constitutes an important feature. Due to dispersive wave propagation within the vibrating guitar string, the overtone frequencies f_k deviate from the ideal integer relationship to the fundamental frequency f_0 . This behavior can be described by:

$$f_k = k f_0 \sqrt{1 + \beta k^2}; \quad k \geq 1 \quad (1)$$

Extraction of the inharmonicity coefficient β from instrument recordings has been covered in the literature with different approaches [5]. In physical modeling of string instruments, inharmonicity is often included into the synthesis models in order to achieve a more natural sound [6, 7].

4. NOVEL APPROACH

Our approach incorporates frame-wise time-frequency transform, onset detection, multi-pitch estimation, feature extraction, and classification. This frame-wise processing enables a real-time string detection. All processing steps are explained in detail in the next sections.

4.1. Time-frequency transform

In order to achieve a sufficiently high frequency resolution, we use frequency reassignment as described in [8]. For each spectral frame, frequency reassignment yields a magnitude spectrum and an instantaneous frequency spectrum that assigns an instantaneous center frequency to each spectral bin. It has the nice property that strong sinusoidal peaks tend to produce stable instantaneous frequencies in the bins surrounding its local maximum. Thus, we can accumulate the magnitudes onto an enhanced magnitude spectrum exhibiting sharper and more pronounced peaks at the location of sinusoidal signal components. Secondly, it can be easily converted to a logarithmically spaced frequency axis by means of linear interpolation. The target frequency axis is logarithmically spaced between 73.41 Hz and 22050 Hz and has a spectral resolution of 288 bins per octave.

4.2. Onset detection

Since our approach is intended to run in real-time, onset detection is performed in a frame-wise manner. For each incoming spectral frame, we compute the relative difference function [9] to the previous frame. This function exhibits clear peaks at spectral changes, such as onsets of new notes or slides from one note to the other. For each peak, we collect spectral frames in order to derive one representative spectrum for multi-pitch estimation (see Section 4.3). Our strategy is to collect 5 frames and take the absolute maximum over time in each frequency bin as an estimate of the overall spectral content. Although we are violating the additivity constraint¹⁴ in this case, our experiments showed that this procedure yields the best results. Moreover, as the multi-pitch estimation is computationally expensive, taking only one representative spectrum per onset into account guarantees real-time applicability.

4.3. Multi-pitch estimation

In order to get an estimate of pitch candidates being played, we employ the recently proposed Blind Harmonic Adaptive Decomposition algorithm (BHAD) by Fuentes et al. [10, 11]. Similar to previous approaches such as Specmurt [12] and Shift-Invariant Probabilistic Latent Component Analysis [13], this algorithm models the observed spectrum as a convolution of spectral kernels (representing tonal instruments) with spectral activations (representing pitch candidates) along the logarithmic frequency axis. The usage of logarithmically spaced frequency axes allows the spectral kernels to have a fixed shape, since a change in pitch can be approximated by shifting them upwards or downwards along the frequency axes. In contrast to previous approaches, the BHAD algorithm has the nice property to model the overtone spectrum of each pitch candidate individually as a linear combination of prototypic overtone distributions. In our experiments, this feature turned out to be helpful for detecting multiple occurrences of the same note played in different octaves. This is frequently encountered in common guitar chords played across all 6 strings. Figure 2 shows the pitch candidates BHAD produces with an F-major barre chord¹⁵. This is a real-world example played on a Fender Stratocaster guitar. The magnitude spectrum and pitch candidate probabilities have been scaled to unit maximum for better visibility. The frequency axes are the same in both plots and range from 73 Hz to 3500 Hz. It can clearly be seen, that the spectrum exhibits many strong harmonics. This is due to the fact that each note in the chord produces its own harmonic series. Due to superposition of multiple sinusoids in combination with the typical transfer function of electric guitar pick ups, some of the overtones are stronger than the fundamentals. In the pitch candidates, 6 clearly pronounced peaks are visible, each is marked with its corresponding note name. As can be seen, the pitch class F appears in three different octaves and the C in two octaves. As a reference, the most common fingering chart for this chord is displayed in the lower right corner of Figure 2.

4.4. Feature Extraction

We refer the reader to [1] for a detailed description of the employed audio features. Here, we will focus on the most important ones for the string classification task. The basic procedure is to take each

¹⁴Additivity is imposed by the multi-pitch estimation algorithm used here.

¹⁵The most common fingering for this chord has the index finger pressing all 6 strings on the 1st fret, while middle-, ring- and little finger are put on frets 2 and 3.

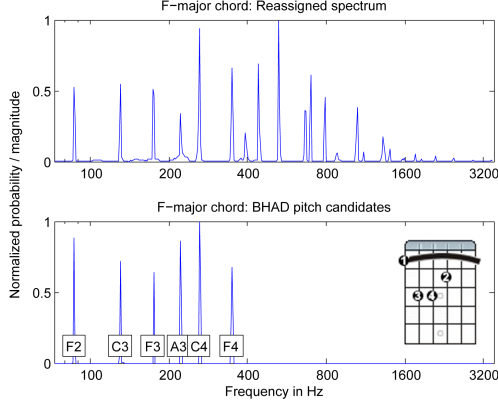


Fig. 2. Example for multi-pitch estimation via BHAD

Feature	Dimensionality
Normalized harmonic frequency deviations	15
Relative harmonic magnitudes	14
Statistics over normalized harmonic frequency deviations	7
Statistics over relative harmonic magnitudes	7
Tristimulus	3
Inharmonicity coefficient β	1
Harmonics magnitude slope	1
Spectral irregularity	1
Odd-to-even harmonics energy ratio	1
Spectral centroid	1

Table 1. Audio features used for the string classification and their dimensionality.

pitch candidate derived from the BHAD algorithm into closer inspection. Since spurious peaks can be found among the pitch candidates, we accept a larger number of candidates than strings available (that is 18 instead of 6). For each candidate, we detect the center frequencies and magnitudes of its harmonics by means of spectral peak picking in the linearly spaced reassigned spectrum. Again, we do not restrict the search to the ideal harmonic frequencies but allow for inharmonicity instead. So for each pitch candidate, we yield the fundamental f_0 as well as the first 14 partials f_k and their respective magnitudes a_0 and a_k . We estimate the inharmonicity coefficient β by minimizing the difference between ideal harmonic frequencies corresponding to Eq. (1) and the observed f_k . This is done in an iterative fashion starting with the first 5 harmonics and an initial $\beta = 0$. After each iteration, more harmonics are taken into consideration and β is updated accordingly. Additionally, we compute various audio features that capture the amplitude and frequency characteristics of the harmonics per pitch candidate as listed in Table 1. In audio feature extraction, descriptors are usually extracted from the complete spectrum. Instead, we only use the discrete frequencies and magnitudes of the harmonics as compact spectral representation.

4.5. String Classification

In order to automatically estimate the string number n_s for each pitch candidate, we concatenate a 51-dimensional feature vector $\{x_i\}$ from the above mentioned features. For feature selection, we use the Inertia Ratio Maximization with Feature Space Projection (IRMFSP) as described in [14]. For classification, we use Support Vector Machines (SVM) [15]. Our SVM configuration uses a linear basis function. The SVM returns probabilities $\{p_i\}$ to assign unknown

Confusion matrix - Chords - IRMFSP-SVM

class (correct)	1	2	3	4	5	6	noNote
1	86.4	0.0	0.0	0.0	0.0	0.0	13.6
2	2.3	97.7	0.0	0.0	0.0	0.0	0.0
3	0.0	2.3	97.7	0.0	0.0	0.0	0.0
4	0.0	0.0	2.3	97.7	0.0	0.0	0.0
5	0.0	0.0	0.0	2.3	95.5	0.0	2.3
6	0.0	0.0	0.0	0.0	4.5	95.5	0.0
noNote	0.0	0.0	0.0	0.0	0.0	9.1	90.9
class (classified)	1	2	3	4	5	6	noNote

Fig. 3. Confusion matrix of the best performing string classification. All values given in percent.

samples to each string class. We estimate the string number \hat{n}_s as $\hat{n}_s = \arg \max_i \{p_i\}$.

4.5.1. Plausibility Filter

As mentioned earlier, most pitches within the relevant frequency range can be produced on multiple fret positions. Based on knowledge about the instrument tuning, we can derive a set of MIDI pitch values that can be played on each string. Consequently, we set the probability values in $\{p_i\}$ to zero for all implausible strings before estimating the string number. We start iterative by assigning the pitch candidate with the highest classification probability to the classified string. Subsequently, we mark this string as occupied so that the remaining pitch candidates have to be placed on the remaining strings. During this procedure, a maximum fret distance of 6 is set as a physiological constraint. If this is violated we swap strings.

5. EVALUATION & RESULTS

We report the performance of our proposed approach on manually annotated ground truth datasets. The classes we want to discern are the 6 different strings, enumerated from 1 to 6. An additional class noNote is introduced to capture spurious pitch candidates.

5.1. Dataset

For the evaluation experiments, we use a ground truth of 712 guitar samples compiled from different datasets. For training, we recorded and annotated notes and chords with a Fender Stratocaster and an Ibanez Power Strat. For testing, we used a subset of the recordings described in [16], that is a Schecter guitar played with a plectrum. It should be noted that testing with samples from a different instrument than the one used for training poses a more difficult task than just splitting samples from one instrument into training and testset. Each set contains barre chords played on 5 to 6 strings. Overall, there are 44 different chords, comprising major and minor chords played on frets 0 to 12.

5.2. Experiments & Results

Multi-pitch Transcription

Our evaluation dataset comprises 2094 single guitar notes. Using the multi-pitch estimation described in Section 4.3, we observed a

recall of 97.23 % and a precision of 73.32 %, yielding an f-measure of 83.59 %.

Estimation of Fretboard Position

Figure 3 shows the confusion matrix we obtained with the best classifier configuration using IRMFSP in combination with linear SVM. The confusion matrix exhibits a clear main diagonal, the mean accuracy is approx. 92%. For almost all strings, a small share of confusion with the neighboring strings is evident. Furthermore, a substantial confusion of the lowest string with the noNote class can be observed. Closer inspection of this class revealed that in some cases it is not safe to start the string assignment during plausibility filtering with the note that yielded the highest $\{p_i\}$. This is in line with the findings in [17], where the correspondence between classification probabilities and classification confidence has been critically investigated.

Estimation of Chord Fingering

Our dataset comprises 44 guitar chords played with different fingerings, 84 % of them were identified correctly. In comparison, Barbancho et al. use 330 different fingerings in [3] and could identify 88 % correctly. It should be noted, that in both works, only incorrect string and fret combinations are penalized. Notes that have not been found by the multi-pitch transcription are ignored.

Latency (Real-time Performance)

The proposed system has an inherent latency of 15 frames after detecting a note onset. Given a hop size of 512 samples and a sampling frequency of 44.1 kHz, the latency is 174 ms.

6. CONCLUSIONS & OUTLOOK

We presented our novel approach for real-time string detection in guitar recordings. Preliminary results show competitive classification performance even with full chords across all 6 strings. In the future, we will extend the training and test scenarios by inclusion of additional datasets. We plan to investigate complementary feature adaption strategies, such as Linear Discriminant Analysis. Furthermore, we want to assess different aggregation strategies of intermediate classification probabilities.

7. ACKNOWLEDGEMENTS

We would like to thank Benoit Fuentes for making his reference implementation of the BHAD algorithm publicly available. In addition, we would like to express our gratitude to Isabel Barbancho et al. and Michael Stein for sharing their dataset of guitar recordings.

8. REFERENCES

- [1] Jakob Abeßer, “Automatic string detection for bass guitar and electric guitar,” in *Proc. of the 9th International Symposium on Computer Music Modeling and Retrieval (CMMR)*, London, UK, 2012.
- [2] Isabel Barbancho, Ana M. Barbancho, Lorenzo J. Tardón, and Simone Sammartino, “Pitch and Played String Estimation in Classic and Acoustic Guitars,” in *Proc. of the 126th Audio Engineering Society (AES) Convention*, Munich, Germany, 2009.
- [3] Isabel Barbancho, Lorenzo J. Tardón, Simone Sammartino, and Ana. M. Barbancho, “Inharmonicity-based method for the automatic generation of guitar tablature,” *IEEE Transactions on Audio, Speech, and Language Processing*, pp. 1857–1868, 2012.
- [4] Neville H. Fletcher and Thomas D. Rossing, *The Physics Of Musical Instruments*, Springer, New York, London, 2 edition, 1998.
- [5] Alexander Galembo and Anders Askenfelt, “Signal representation and estimation of spectral parameters by inharmonic comb filters with application to the piano,” *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 2, pp. 197–203, 1999.
- [6] Hanna Järveläinen, Vesa Välimäki, and Matti Karjalainen, “Audibility of the timbral effects of inharmonicity in stringed instrument tones,” *Acoustics Research Letters Online*, vol. 2, no. 3, pp. 79, 2001.
- [7] Hugo Fastl and Florian Völk, “Inharmonicity of sounds from electric guitars: Physical flaw or musical asset?,” in *Proc. of the 10th International Conference on Music Perception and Cognition (ICMPC)*, Sapporo, Japan, 2008.
- [8] Toshihiko Abe, Takao Kobayashi, and Satoshi Imai, “Robust pitch estimation with harmonics enhancement in noisy environments based on instantaneous frequency,” in *Proc. of the Fourth International Conference on Spoken Language Processing (ICSLP)*, Philadelphia, PA, USA, 1996.
- [9] Anssi P. Klapuri, Antti J. Eronen, and Jaakko T. Astola, “Analysis of the meter of acoustic musical signals,” *IEEE Transactions on Speech and Audio Processing*, pp. 342–355, 2004.
- [10] Benoit Fuentes, Antoine Liutkus, Roland Badeau, and Gaël Richard, “Probabilistic model for main melody extraction using constant-Q transform,” in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, March 2012, pp. 5357–5360.
- [11] Benoit Fuentes, Roland Badeau, and Gaël Richard, “Blind harmonic adaptive decomposition applied to supervised source separation,” in *Proc. of the 20th European Signal Processing Conference (EUSIPCO)*, Bucharest, Romania, August 2012, pp. 2654–2658.
- [12] Shoichiro Saito, Hirokazu Kameoka, Keigo Takahashi, Takuya Nishimoto, and Shigeki Sagayama, “Specmurt Analysis of Polyphonic Music Signals,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 3, pp. 639–650, Feb. 2008.
- [13] Gautham J. Mysore and Paris Smaragdis, “Relative pitch estimation of multiple instruments,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, Taiwan, 2009, pp. 313–316.
- [14] G. Peeters and X. Rodet, “Hierarchical gaussian tree with inertia ratio maximization for the classification of large musical instruments databases,” in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx)*, London, UK, 2003.
- [15] Vladimir N. Vapnik, *Statistical learning theory*, Wiley New York, 1998.
- [16] M. Stein, J. Abeßer, C. Dittmar, and G. Schuller, “Automatic Detection of Audio Effects in Guitar and Bass Recordings,” in *Proc. of the 128th Audio Engineering Society (AES) Convention*, London, UK, 2000.
- [17] Hanna Lukashevich, “Confidence measures in automatic music classification,” in *Proc. of the 36th Annual Conference of the German Classification Society (GfKI)*, Hildesheim, Germany, 2012.