Automatic Guitar String Detection by String-Inverse Frequency Estimation

Tobias Geib[1] Maximilian Schmitt[2] and Björn Schuller[3]

Abstract: In this work, we present a novel approach to approximating the fretboard position, i. e., the string and fret combination of guitar and bass recordings, using a feature we call String-Inverse Frequencies (SIFs). These frequencies are obtained from the opposite part of the string pressed down on a fretboard. We then show how they are calculated and proof their usefulness for guitar string detection. Additionally, a database is featured with recordings specifically tailored for this task. Furthermore, we demonstrate a basic approach using SIFs based on FFT spectral analysis and compare it to a basic standard classification process using Mel-Frequency Cepstral Coefficients and Support Vector Machines. The SIF-based approach showed a detection rate of up to $F_1 = 72\,\%$ for both guitar and bass. Finally, we discuss further possibilities regarding SIFs.

Keywords: Guitar String Detection; Music Information Retrieval; Frequency Estimation

1 Introduction

Stringed instruments, such as guitar or bass, experience a range of potential notes which may *overlap* for different strings. As a consequence, it is possible for the string, on which a note is being played, to differ, even if the same frequency is perceived. Each possibility is a combination of the fret and the string, which can also be represented by the frequency of the empty string. In standard music notation, only the pitch and the length of a note are notated. On guitar for instance, a performer would then need to interpret which string was intended by the composer to be played. Consequently, music notation systems dedicated to stringed instruments have been established, such as *tablature*, as described by Wade [Wa10]. In tablature, each line represents a string and the numbers on top of the line represent the fret number at which the string is supposed to be played, thereby removing any ambiguity. As such, tablature is a helpful tool for music education especially considering beginning players.

[1] Chair of Complex and Intelligent Systems, University of Passau, Innstraße 43, 94032 Passau, Germany
tobiasl.geib@gmail.com

[2] Chair of Complex and Intelligent Systems, University of Passau, Innstraße 43, 94032 Passau, Germany
maximilian.schmitt@uni-passau.de

[3] Chair of Complex and Intelligent Systems, University of Passau, Innstraße 43, 94032 Passau, Germany; Department of Computing, Imperial College London, 180 Queens' Gate, London SW7 2AZ, U. K.
schuller@ieee.org

doi:10.18420/in2017_08

Recently, computer automated systems have experienced increased applicability in various fields and consequently the ambition for automatically annotating music has risen as well. Automatic tablature annotation has since been a goal in research, as was presented by Tuohy et al. [TP05], where a system for mapping note sequences to playable tablature was proposed. In contrast to standard notation, tablature requires not only information about what note is being played, but also which specific string produced this note, making it comparatively more challenging. To date, many different approaches have been attempted. A common method is similar to Musical Instrument Recognition, where the timbral characteristics of each string are trained and tested using machine learning. Such an approach was proposed by Maezawa et al. [Ma09], where violin recordings with given standard notation were classified via a Gaussian Mixture Model. In [Ba12], Barbancho et al. proposed a different approach, using a feature called *inharmonicity*, where the divergence from the harmonic overtone series for each string was analysed and used for classification. This concept was combined with machine learning approaches by Abeßer et al. [Ab12], where a multitude of partial features, including inharmonicity, where extracted and then reduced in dimensionality using Linear Discriminant Analysis. After applying a plausibility filter, the classification was then performed by a Support Vector Machine (SVM) using an RBF kernel, achieving an F-measure of up to $F_1 = 90\,\%$ for guitar. Music transcription in general has been a challenge in research especially regarding polyphonic signals [We13]. As such, a study conducted by Marolt et al. [Ma04] showed the usefulness of neural networks for polyphonic transcription of piano music. An additional challenge was tackled by Fiss et al. in [FK11], where Automatic Guitar Transcription was computed in real-time. Another area of research is the transcription of guitar chords. Barbancho et al. [BTI11] used a Hidden Markov Model in order to accurately model chord fingerings.

In this contribution, we propose a novel approach to Automatic Guitar String Detection (AGSD) of monophonic signals, in which the frequency of the opposite part of the string is estimated. This frequency is called the String-Inverse Frequency (SIF) f_n^{-1}, which together with the frequency of the played note f_n can be used to classify the string by the unique frequency pair (f_n, f_n^{-1}). The challenge presented by this task is given through both the architecture of the guitar and the nature by which the frequency is produced. The guitar is built in a way to produce a pitch following the chromatic scale within a certain range. As such, it is desirable to have the plucked part of the string ringing as true as possible. However, the opposite part of the string does not in all cases follow any musical measure and is rather unwanted to be ringing. Therefore, the architecture of the guitar does not naturally support this part of the string and its audibility might differ for each guitar. Furthermore, the SIF is produced not by plucking the opposite part of the string, but instead by the action of pressing down the string onto the fretboard, thereby causing both parts to experience excitation simultaneously and equally in size. Both parts then begin to ring, thereby creating a polyphonic audio even though only one string is being played. Accordingly, an additional difficulty is to differentiate which frequency belongs to which part of the string. All of these aspects can compromise practical usability of this approach. AGSD solely based on SIF detection was applied for research purposes, and its applicability in less controlled environments is topic of further discussion. In the following Section 2, we will introduce the concept of SIF. In Section 3, the database employed in our experiments is introduced, followed by a description of the whole AGSD approach and a baseline approach in Section 4. Results are presented in Section 5, before we draw a conclusion in Section 6.

2 String-Inverse Frequencies

In this work, we make use of the physical properties of stringed and fretted instruments. We examine these properties, in order to utilise them in context of SIFs.

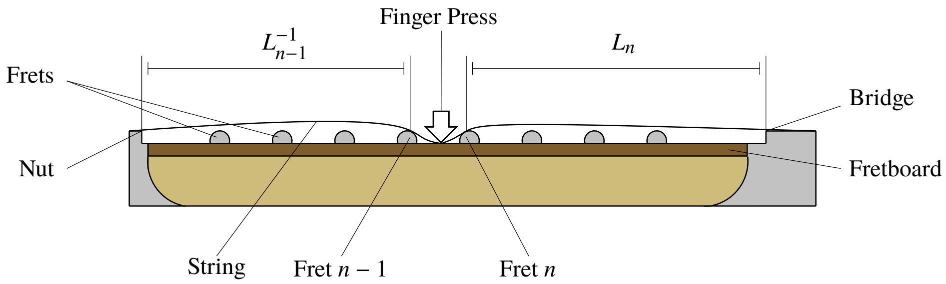

Fig. 1: A schematic model of a guitar neck. The sideview of a guitar during the action of pressing down a string is displayed. It shows the opposite part of the string, which produces the SIF on the left side and the normal part of the string, which produces the Normal Frequency, on the right side. For demonstration purposes, the distance between each fret is displayed as equal and does not represent fret spacing on actual guitars.

2.1 Physical Properties of the Guitar

Starting from the nut, which marks the beginning of the vibrating length of an open string, each fret is spaced chromatically, which will result in an increase of a semi-tone interval per fret. We can therefore determine the frequency of a string, that is played on a given fret, by using the equation for calculating a semi-tone interval based on equal tempered tuning as presented by White et al. [WW14].

Lemma 2.1. Let f be a frequency measured in Hz. The frequency $\tilde{f}$ which is a diatonic semi-tone higher than f can then be described as follows:

$$\tilde{f} = f \cdot 2^{\frac{1}{12}}. \tag{1}$$

As every fret represents a semi-tone leap from the frequency of the preceding fret, we can now calculate the frequency of any given fret on a string as

$$f_n = f_0 \cdot 2^{\frac{n}{12}}, \tag{2}$$

where n represents the number of the fret, and f_0 represents the frequency of the empty string. Another physical property of the strings is the relation between pitch and length of the string. A string of length L and with a tension T produces a pitch of frequency f. As described in [Ha02], the frequency of a string can be determined as

$$f = \frac{v}{2L}, \tag{3}$$

with $v = \sqrt{T/\mu}$ denoting the velocity of propagation of the wave and μ denoting the density of the string material. In our case, only guitars with steel strings were used. Consequently, if L is equal to the length of the scale m, i. e., the maximum length for a guitar string, then the produced pitch will be the frequency of the empty string:

$$f_0 = \frac{v}{2m}. \tag{4}$$

The pitch for two strings with the same tension and density will thus change only if the length differs. The difference in length is then inversely proportional to the change in pitch:

$$f_0 \cdot x = \frac{v}{2m \cdot x^{-1}}. \tag{5}$$

The pitch for any fret can be calculated as shown in equation 2. Consequently, the length of a string played on the n-th fret can now be derived using equation 5:

$$f_n = f_0 \cdot 2^{\frac{n}{12}} = \frac{v}{2m \cdot 2^{-\frac{n}{12}}}. \tag{6}$$

This is an important property for the purpose of this work, since we will use it for calculating the SIFs, as will be shown in the following section.

2.2 Calculation of the String-Inverse Frequencies

For the proposed method, an additional property of the guitar is needed. Section 2.1 established the frequency of a string, when the finger is placed at a specific fret, as described in equation 2. However, this only describes the frequency of the string, when the plucking point, i. e., the point where the string is *struck* in order to generate the vibration, is located on the end towards the bridge, which represents the normal way of playing the instrument.

Suppose, one would pluck the string on the opposite end towards the nut. The string would then produce a second frequency. This is the **String-Inverse Frequency** (SIF), since its pitch is inversely related to the **Normal Frequency** (NF). When the pitch of the normal frequency goes up, the SIF goes down. Since the pairs of normal and inverse frequencies are unique for each fret, we can use this property to determine which fret on which string has been played.

In order to make use of the SIF, we first need to accurately determine it. From equation 6 it is known that the distance between the fret and the bridge is $L_n = m \cdot 2^{-\frac{n}{12}}$. Let L_n^{-1} be the distance between the nut and the given fret n, and since m denotes the length of the scale, L_n^{-1} can then be determined by subtracting L_n from the scale length m:

$$\begin{aligned} L_n^{-1} &= m - m \cdot 2^{-\frac{n}{12}} \\ &= m \cdot (1 - 2^{-\frac{n}{12}}). \end{aligned}$$

The resulting pitch of the string with a length of the distance from the nut to the fret, i. e., the SIF $f_n'^{-1}$, can thus be calculated using equation 5:

$$\begin{aligned} f_n'^{-1} &= \frac{v}{2m \cdot (1-2^{-\frac{n}{12}})} \\ &= \frac{f_0}{(1-2^{-\frac{n}{12}})}. \end{aligned} \tag{7}$$

However, since the finger is placed between two frets, the distance that needs to be calculated is one fret lower, i. e., the fret that is closer to the nut than the fret that is meant to be played. Therefore, we account for this by subtracting the fret number n by 1 and we can thus derive the definition for SIFs.

Definition 2.1. Let f_0 be the frequency of the empty string and n be the number of the played fret. The SIF of f_n, called f_n^{-1}, is then defined as:

$$f_n^{-1} = \frac{f_0}{(1-2^{-\frac{(n-1)}{12}})}. \tag{8}$$

Important for AGSD is the uniqueness of the SIFs. Since the pair of both NF and SIF (f_n, f_n^{-1}) is used for classification, it is crucial that no two equal pairs exist.

To proof this, let us assume that there are two strings with frequencies f_n and $\hat{f}_m$, where $f_n = \hat{f}_m$ and $n \neq m$. Suppose $f_0 > \hat{f}_0$. Accordingly, this means $n < m$, which in turn implies $f_n^{-1} > \hat{f}_m^{-1}$, since the string-inverse scale is inverse to the chromatic scale, i. e., a higher fret will produce a lower SIF. The assumption $f_0 < \hat{f}_0$ yields the analog result of $f_n^{-1} < \hat{f}_m^{-1}$, thus showing in every case that $f_n^{-1} \neq \hat{f}_m^{-1}$, provided that $f_n = \hat{f}_m$ and $f_0 \neq \hat{f}_0$. Therefore, every pair (f_n, f_n^{-1}) is unique and the string can be classified by detecting it.

3 Database

In order to ensure optimal evaluation, a database was specifically recorded for the task described in this work. As common guitar recording process does not attribute for SIF detection, there is a special need for the creation of a new database. For the recording, an aspect considered was the acoustical attribute which SIFs inherit, meaning they are negligibly audible in electric recordings. As electric recordings we considered such, which record the direct output of an electric pickup system. The pickup system usually consists of one or two magnetic coils placed right underneath the steel strings, where the only part which interacts within their magnetic field is producing NFs. Therefore, no SIF will be recorded under these conditions. Another electric recording system which provides electric output is often found in acoustic guitars as described by Mariner [Ma77]. In such a system, the vibrations of the guitar body are amplified in order to create an electric output of the sound. Theoretically, this would allow us to audibly detect the SIFs. However, since the guitar body is built in a shape which amplifies NFs, this kind of electric output does not produce an optimal recording for our means, either.

Therefore, an acoustic set-up was chosen for the audio-recording, meaning that every guitar used in this database was recorded with a microphone placed in front of the guitar. The exact placement of the microphone was also thoroughly considered. Commonly, when recording a guitar acoustically, the microphone is placed directed at the soundhole, since this allows for best sound projection. However, for the means of this work, a different placement was chosen, since a non-biased recording regarding loudness of either SIFs or NFs was desired. For this reason, the microphone was placed facing approximately at the median length of the fretboard. In this position it is assumed for the microphone to be equally distanced to both parts of the string, when averaged over all playing positions.

The dataset features three different kinds of guitars: an *electric*, an *acoustic*, and a *bass* guitar. All instruments recorded using the same method. An electric guitar was used, even though SIFs are only acoustically detectable, since an acoustical recording of an electric guitar reduces the bias for normal frequencies given through the build of the guitar. This is due to its design, not being intended for acoustic projection. Therefore, the difference in loudness of both frequencies is minimised, which allows for optimal conditions and is thus useful for this work. Furthermore, two different kinds of plucking techniques for each guitar type are featured. One is the standard way of plucking, where upon pressing down at a fret the string is *plucked*. In these recordings, a plectrum was used for plucking. The other technique, which from here on out will be referred to as the *fretted* technique, was recorded by pressing down the string onto the fretboard. However, this action was performed without also plucking the string. This kind of variant was chosen since the only impulse given to produce the SIF is the string, as it is pressed down against the metal frets. As a consequence for no part of the string being plucked, the same energy is given to excite both parts. By not plucking the string the best possible loudness distribution between both parts is achieved, which is why it was chosen for this database. An overview over the number of samples for each instrument and plucking style is given in Table 1.

For each type of guitar and playing style, the recordings include one sample of each, within the scope from fret 3 up to and including 14, deriving a total of 12 samples per string. This scope was chosen, since a lower fret number will produce a higher SIF and as the spacing between each fret increases for each lower fret, the relative pitch between each fret will increase exponentially. Correspondingly, the length of the inverse string will also decrease and as such not only will the sound decrease, but also the sound will ring more faintly compared to SIFs produced by higher fret numbers. Therefore, even though theoretically calculable, the SIF produced by the second fret is practically infeasible to be detected by the given set-up and was thus not included in the recorded material.

Each sample features one isolated note. Additionally, for each type of guitar and plucking technique, the database provides samples of each empty string, in order to provide the tuning of the guitar for each recording session. For our purposes, the tuning is important since the frequency of the empty string f_0 is needed for calculating the SIF, as can be seen in equation 8.

<table>
<tr><th></th><th>Electric Guitar</th><th>Acoustic Guitar</th><th>Bass</th><td rowspan="4">$\sum = 392$</td></tr>
<tr><td>fretted</td><td>72</td><td>72</td><td>48</td></tr>
<tr><td>plucked</td><td>72</td><td>72</td><td>48</td></tr>
<tr><td>Total</td><td>148</td><td>148</td><td>96</td></tr>
</table>

Tab. 1: Number of samples included in the database and its subsets

4 Proposed System

In this section, the system built for AGSD will be discussed. The SIF-based approach, as well as the comparative approach using Mel-Frequency Cepstral Coefficient (MFCC) features is described. Additionally, the approach for pitch detection is included.

4.1 Pitch Detection

As a first step, the Normal Frequency (NF) is detected, since it is crucial for the string detection process. If the measured frequency is too inaccurate, the list of possible string-fret combinations $C = [(n^1,f^1),(n^2,f^2),\dots,(n^k,f^k)]$ will also be incorrect, therefore not yielding useful results when the SIF is calculated.

For pitch detection, the openSMILE toolkit as established by Eyben et al. [Ey13] was used. openSMILE utilises framewise *Subharmonic Summation* (SHS) to determine the fundamental frequency. The frame size was set to 10 ms, with a minimum pitch of 40 Hz, which approximately marks the lowest possible frequency of a bass guitar. The compression factor for SHS was set to $h = 0.85$, as a slight alteration to [He88]. Since each sound file is isolated, no additional steps for detecting onset or offset were taken. As the algorithm determines the pitch for each time frame, an additional decision process was employed in order to arrive at one final pitch for the entire file. To achieve this, a voting based algorithm was built, in which for each frequency F_t per time frame a vote is given to a candidate frequency F_c, if F_t falls within the semi-tone range of F_c. The semi-tone range of a frequency f is determined by taking the semi-tone above and below f and calculating the middle point between f and each respective semi-tone:

$$\begin{aligned} &\text{Upper Bound:} \quad \left(f + f \cdot 2^{\frac{1}{12}}\right) \cdot \frac{1}{2}, \\ &\text{Lower Bound:} \quad \left(f + \frac{f}{2^{\frac{1}{12}}}\right) \cdot \frac{1}{2}. \end{aligned} \tag{9}$$

If F_t does not fall into the semi-tone range of F_c, F_t is checked for all remaining candidates and if no match is found, F_t is added to the list of candidates. The candidate with the most votes is then chosen, and the pitch is calculated by averaging over each frequency which voted for this candidate.

4.2 String-Inverse Frequency Approach

In order to classify the SIF, the system first determines the tuning $t = [f_0^1, f_0^2, \ldots, f_0^l]$ with l being the number of strings of the respective instrument, by applying the pitch detection algorithm. For the given sample, the NF is then determined. As described in Section 3, it is possible for the SIF to ring louder than the NF and, by extension, to cause the pitch detection to detect the SIF instead of the NF. This is especially likely to occur for notes played around the 12th fret and higher, since it represents the point where both scales cross each other and thereby causing the inverse string to be longer and therefore be louder than the NF, with both frequencies being in the proximity of each other.

When the NF is determined, the list of string-fret combinations $C = [(n^1, f^1), (n^2, f^2), \ldots, (n^k, f^k)]$ can be determined, with n denoting the fret number, f denoting the frequency of the empty string, and k denoting the number of possible combinations. This list is derived by iterating over every tuning entry and calculating the frequency of each fret for the given string. The respective pair is then added to C, if the frequency falls into its semi-tone range as presented in equation 9.

The SIF of each combination in C is then calculated. By using equation 2.2, the list of possible SIF candidates can be derived as $C^{-1} = [f_{n^1}^{-1}, f_{n^2}^{-1}, \ldots f_{n^k}^{-1}]$. For each entry Subharmonic Summation is applied as proposed by Hermes [He88]

$$H(f) = \sum_{n=1}^{N} h_n \cdot P(f \cdot n), \tag{10}$$

with f denoting the frequency to be summed up, N denoting the number of partials considered, h_n denoting the weight for each partial, and $P(i)$ denoting the excitation value in the spectrum at position i.

During the testing phase of this system, we noticed deviations of the SIF from the mathematically accurate frequency. In order to compensate for this effect, a semi-tone scope was applied to the target frequency. For each frequency bin within the scope, the SHS was calculated and then finally averaged to derive the SHS value of the target frequency. By applying this, the list of averaged SHS values of each SIF can be derived as $\hat{C}^{-1} = [H(f_{n^1}^{-1}), H(f_{n^2}^{-1}), \ldots H(f_{n^k}^{-1})]$. Since the SIF and the NF share minimal number of partials, the string class (n,f) can be determined by choosing the string with the highest SHS value of its respective SIF:

$$(n,f) = \arg\max_k (H(f_{n^1}^{-1}), H(f_{n^2}^{-1}), \ldots H(f_{n^k}^{-1})). \tag{11}$$

Having estimated the base-frequency f, the string class s is then retrieved by matching it with the tuning t.

4.3 Baseline MFCC Approach

Since the results presented in this work are based on a specifically recorded database, a comparison with results yielded by other approaches could be misrepresentative. Thus, in order to present a comparable measure, we also built a standard machine learning system similar to the baseline experiment proposed in [Ab12]. The system was trained and tested on the database introduced in Section 3.

For this system, we again utilised the openSMILE toolkit introduced in [Ey13] to extract MFCC features, which have been used for music modelling in [Lo00]. For each sample, the functionals of the MFCCs, i. e., moments and further statistics as provided by openSMILE, were computed, resulting in a feature vector of length 360 per audio sample. The extracted features are then passed to the machine learning tool. In this work, we used the machine learning library introduced in [Pe11] with Linear Support Vector Classification (SVC) and applied a stratified 10-fold cross validation training and testing as proposed by [Ko95]. Accordingly, the training set was composed of $\frac{9}{10} \cdot S$ random samples, with S denoting the total number of samples, and the testing set $\frac{1}{10} \cdot S$ random samples, with no sample being represented in both sets at the same time. For training and testing, all samples, belonging to either guitar or bass, were considered, without differentiating between acoustic and electric or the respective play style, in order to ensure large enough training and testing sets. The evaluation was conducted by averaging the results over each fold. In order to avoid overfitting, the SVC was executed for multiple c-values and the results were averaged for each value. The range of the c-values was defined as $c = [10^{-6}, 10^{-5}, \ldots, 10^{1}]$. For each file, a list of probability values p_l is predicted by the SVC. The string s is determined by maximizing p_l as

$$s = \arg\max_l p_l. \tag{12}$$

The pitch detection algorithm was additionally applied, thus retrieving the normal frequency. This process was employed in order to ensure the comparability of both systems. The SIF based approach uses knowledge of the fretboard to determine which strings can produce a given frequency and, thus, the MFCC approach needs to utilise the same knowledge. As described in Section 4.1, the list of possible combinations C is determined. Each string is assigned a *playability* value $p'(s_j)$ with

$$p'(s_j) = \begin{cases} 0, & \text{if } (n_j, f_j) \notin C, \\ 1, & \text{if } (n_j, f_j) \in C. \end{cases} \tag{13}$$

The playability values are each respectively multiplied with the probability values

$$\hat{p}_j = p_j \cdot p'_j, \tag{14}$$

thus making unplayable strings improbable as well. By employing this strategy, the string class is then determined as

$$s = \arg\max_{l} \hat{p}_l. \tag{15}$$

5 Experimental Results and Evaluation

In this section, we present the results yielded from our experiments. Table 2 shows the results in terms of F-measure (F_1) of both approaches as described in Section 4.

Guitar	**SIF** *fretted*	**SIF** *plucked*	**MFCC**	**Chance Level**
Electric Guitar	44 %	**60 %**	59 %	43 %
Acoustic Guitar	35 %	56 %		
Bass Guitar	71 %	58 %	**84 %**	49 %

Tab. 2: Experimental Results (F_1) with Pitch Detection

The results written in bold represent the best result of both approaches for guitar and bass respectively. The average chance level, which is different for each note, is also displayed in order to give additional context to the results yielded. It is important to note that all results yielded from the guitar samples with the *fretted* play style show values below the MFCC baseline at 59 %. The electric guitar dataset barely overpassed the chance level of 43 % with 44 % and the acoustic guitar dataset even presented values underneath the chance level with 35 %. However, *plucked* samples showed clearly higher results, with the electric guitar presenting the highest detection rate for guitar at 60 %. The bass guitar showed higher results for the fretted samples at 71 %, however, the MFCC approach presented the highest F-measure at 84 %.

It is also notable that the fretted samples, with the exception of bass recordings, show significantly lower results than plucked samples, contrary to the assumption in Section 3, that the fretted play style would result in higher detection rates. In order to investigate why, an experiment was conducted, in which the pitch detection stage was skipped and the ground-truth was replaced as NF. The results are shown in Table 3.

Guitar	**SIF** *fretted*	**SIF** *plucked*	**MFCC**	**Chance Level**
Electric Guitar	**72%**	60%	62%	43%
Acoustic Guitar	54%	56%		
Bass Guitar	73%	58%	**85%**	49%

Tab. 3: Experimental Results (F_1) with Ground Truth for Pitch

The results show a distinct increase in detection rate for fretted samples, while showing no improvement for any plucked recordings. This suggests the pitch detection to be responsible for the previously low detection rate. This might be caused by the fretted samples experiencing louder SIF projection, making it thereby harder for pitch detection to be accurate. For plucked samples, the pitch detection worked as intended, thus replacing it with the ground truth did not lead to an increase in the detection rate. The highest result for all guitars is, as supposed in Section 3, the *fretted* **electric** guitar with 72 %. The same effect can be seen for the bass samples. However, they did not surpass the MFCC approach at 85 %. When compared to the SIF approach, the high results for the MFCC approach might indicate that the bass does not experience highly accurate SIF projection. This might be caused by a particularly unfavourable build when considered for SIF detection, suggesting a potential for the SIF approach to achieve even higher results with samples from a more fitting instrument.

6 Conclusion

In this contribution, we presented a new feature called String-Inverse Frequency. A database recorded for SIF detection was introduced. A method for calculating SIF was established and its applicability for Automatic Guitar String Detection was shown and evaluated in experimental results yielding a top F-measure of 72 % for guitar and 73 % for bass guitar. A comparative baseline system utilising an MFCC-based SVM classifier was built, while being outperformed for guitar samples by the SIF approach, when accounted for pitch detection. Future research aims to fuse the SIF-based and MFCC-based approaches and to further develop the underlying concept. Moreover, the method will be evaluated on other databases, such as the IDMT-SMT-Guitar database by Kehling et al.[4].

Acknowledgement

The research leading to these results has received funding from the European Union's Seventh Framework Programme through Grant Agreement No. 338164 (ERC StG iHEARu).

[4] https://www.idmt.fraunhofer.de/en/business_units/m2d/smt/guitar.html

References

[Ab12] Abeßer, J.: Automatic string detection for bass guitar and electric guitar. In: International Symposium on Computer Music Modeling and Retrieval. Springer, London, U. K., pp. 333–352, 2012.

[Ba12] Barbancho, I.; Tardon, L. J.; Sammartino, S.; Barbancho, A. M.: Inharmonicity-based method for the automatic generation of guitar tablature. IEEE Transactions on Audio, Speech, and Language Processing, 20(6):1857–1868, 2012.

[BTI11] Barbancho, A. M.and Klapuri, A.; Tardon, L. J.; I., Barbancho: Automatic transcription of guitar chords and fingering from audio. IEEE Transactions on Audio, Speech, and Language Processing, 20(3):915–921, 2011.

[Ey13] Eyben, F.; Weninger, F.; Gross, F.; Schuller, B.: Recent developments in openSMILE, the Munich open-source multimedia feature extractor. In: Proc. ACM Multimedia. Barcelona, Spain, pp. 835–838, 2013.

[FK11] Fiss, X.; Kwasinski, A.: Automatic real-time electric guitar audio transcription. In: Proc. ICASSP. Prague, Czech Republic, pp. 373–376, 2011.

[Ha02] Hall, D. E.: Musical acoustics. Brooks Cole, 2002.

[He88] Hermes, D.: Measurement of pitch by subharmonic summation. The Journal of the Acoustical Society of America, 83(1):257–264, 1988.

[Ko95] Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proc. IJCAI. volume 14, Montreal, QC, Canada, pp. 1137–1145, 1995.

[Lo00] Logan, B.: Mel frequency cepstral coefficients for music modeling. In: Proc. ISMIR. Plymouth, MA, USA, 2000.

[Ma77] Mariner, R. E.: , Acoustic pickups, June 1977. US Patent 4,030,396.

[Ma04] Marolt, M.: A connectionist approach to automatic transcription of polyphonic piano music. IEEE Transactions on Multimedia, 6(3):439–449, 2004.

[Ma09] Maezawa, A.; Itoyama, K.; Takahashi, T.; Ogata, T.; Okuno, H.: Bowed string sequence estimation of a violin based on adaptive audio signal classification and context-dependent error correction. In: Proc. International Symposium on Multimedia. San Diego, CA, USA, pp. 9–16, 2009.

[Pe11] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E.: Scikit-learn: machine learning in Python. The Journal of Machine Learning Research, 12:2825–2830, 2011.

[TP05] Tuohy, D. R.; Potter, W. D.: A genetic algorithm for the automatic generation of playable guitar tablature. In: Proc. ICMC. Barcelona, Spain, pp. 499–502, 2005.

[Wa10] Wade, G.: A concise history of the classic guitar. Mel Bay Publications, 2010.

[We13] Weninger, F.; Kirst, C.; Schuller, B.; Bungartz, H.-J.: A discriminative approach to polyphonic piano note transcription using non-negative matrix factorization. In: Proc. ICASSP. Vancouver, BC, Canada, pp. 6–10, 2013.

[WW14] White, H. E.; White, D. H.: Physics and music: the science of musical sound. Courier Corporation, 2014.