GUITARSET: A DATASET FOR GUITAR TRANSCRIPTION

Qingyang Xi[1] **Rachel M. Bittner**[1] **Johan Pauwels**[2]
Xuzhou Ye[1] **Juan P. Bello**[1]
[1] Music and Audio Research Lab, New York University, USA
[2] Center for Digital Music, Queen Mary University of London, UK
tom.xi@nyu.edu

ABSTRACT

The guitar is a popular instrument for a variety of reasons, including its ability to produce polyphonic sound and its musical versatility. The resulting variability of sounds, however, poses significant challenges to automated methods for analyzing guitar recordings. As data driven methods become increasingly popular for difficult problems like guitar transcription, sets of labeled audio data are highly valuable resources. In this paper we present GuitarSet, a dataset that provides high quality guitar recordings alongside rich annotations and metadata. In particular, by recording guitars using a hexaphonic pickup, we are able to not only provide recordings of the individual strings but also to largely automate the expensive annotation process. The dataset contains recordings of a variety of musical excerpts played on an acoustic guitar, along with time-aligned annotations of string and fret positions, chords, beats, downbeats, and playing style. We conclude with an analysis of new challenges presented by this data, and see that it is interesting for a wide variety of tasks in addition to guitar transcription, including performance analysis, beat/downbeat tracking, and chord estimation.

1. INTRODUCTION

Well-annotated audio files are key to MIR research. They are necessary both for evaluating algorithm performance and for developing models. For time-varying musical information such as notes in a polyphonic context, the process of creating accurate annotations can be an especially difficult and slow process. For monophonic audio, there are software tools, such as Tony [12], built to facilitate the manual annotation process by first providing an estimate and allowing the user to manually correct the mistakes. However, there is no equivalent tool for polyphonic audio, and the accuracy of pitch estimation methods on polyphonic audio is significantly worse than for monophonic audio.

Recently, several methods have been developed to address the problem of creating pitch annotations. Most notably, Su and Yang's work [22] provides an efficient way of generating note-level annotations for recordings of polyphonic music by utilizing the midi-keyboard as an annotation interface. An alternative approach was proposed that uses an analysis-synthesis framework to generate annotations by re-synthesizing estimates [18]. However, these methods are insufficient when applied to guitar recordings. In the midi-keyboard approach, it would be very difficult for a keyboard player to replicate note-by-note what a guitarist is playing. The analysis-synthesis approach requires the analysis (i.e. estimate of the correct notes) to be reasonably close to the ground truth in order to generate realistic sounding audio; unfortunately, the existing transcription algorithms perform woefully badly on polyphonic solo guitar recordings. Unsurprisingly, there is no sizable database of guitar recordings with note-level annotations and realistic guitar playing.

In this paper, we present GuitarSet: a sizable dataset of richly annotated, realistic guitar recordings. We describe our data collection and annotation process in detail and introduce our solution for efficiently creating note-level annotations. Our solution relies on the use of an acoustic guitar with a *hexaphonic pickup*, which outputs one channel of audio signal per guitar string; as well as custom annotation tools. This effectively turns polyphonic transcription into monophonic transcription. We conclude with an analysis of new challenges presented by this data, and see that it is interesting for a wide variety of tasks in addition to guitar transcription, including performance analysis, beat/downbeat tracking, and chord estimation. The dataset (audio and annotations) and the code used to generate the annotations are made freely available online. [1]

2. RELATED WORK

A handful of datasets exist for polyphonic instrument transcription. The MAPS dataset [7] contains a large collection of transcribed piano notes, chords, and pieces (using a Disklavier), recorded in different acoustic conditions. Similarly, the UMA-Piano [2] dataset contains all possible combinations of notes at varying dynamics. These datasets have been critical to the development of automated piano transcription methods; Sigtia's deep-learning powered

[1] https://github.com/marl/GuitarSet

piano transcription algorithm [20] and Ewert's algorithm based on non-negative matrix deconvolution [8] are just two of many data driven algorithms that rely on the MAPS dataset. More recently, efforts devoted to historic preservation of player piano rolls also provide new ways of extending transcription datasets for piano music [19].

For guitar, the Guitar Playing Techniques dataset [23] contains 6580 clips of single notes along with playing technique annotations. The IDMT-SMT-Audio-Effects dataset [21] contains $\approx$ 20 hours of single guitar notes and chords with varying audio effects. Finally, the IDMT-SMT-Guitar dataset [11] contains several types of guitar data, including single notes, playing techniques, note clusters, and note and chord-level annotations for short excerpts. While each of these datasets are useful, none of them provide note-level annotations of realistic polyphonic guitar pieces, which is a limiting factor in exploring many interesting new research directions.

The absence of a sizable dataset for realistic polyphonic guitar playing is largely due to the difficulty of annotating complex guitar recordings directly. In order to help facilitate analysis of guitar recordings, hexaphonic guitar pickups have become a useful research tool. The idea of using hexaphonic pickups to generate transcriptions was first proposed by O'Grady and Rickard in 2009 [16]. In their method, signals from individual strings are analyzed using supervised non-negative matrix factorization. Hexaphonic pickups have also been used for analysis and resynthesis of monophonic single-note guitar recordings [15], as well as for visualizing guitar performances [1].

We posit that, despite piano and guitar having comparable popularity, research has focused much more heavily on analysis of piano recordings simply because of the availability of data. Online communities that provide guitar tablature such as Ultimate Guitar [2] are very popular, and accurate methods for guitar tablature transcription would have the potential to attract a vibrant community. By creating GuitarSet, and therefore demonstrating an efficient process of creating detailed note level annotations for guitar, we hope to provide the community with better resources for studying guitar transcription and more.

The collection and analysis methods for GuitarSet was designed with the principles described by Su and Yang [22] in mind: (1) **Generality**: We chose well-known progressions in popular styles as the basis of GuitarSet's material, and collect realistic, complex and polyphonic musical phrases. (2) **Efficiency**: The method of creating annotations for GuitarSet is mostly automated, with human experts focusing on correcting onsets, which requires context and expertise. GuitarSet can be easily extended for this reason. (3) **Cost**: The key equipment, the hexaphonic pickup, is very affordable. and (4) **Quality**: In order to preserve nuances in the performance, including intra-note pitch deviations and inter-string onset-time patterns, we craft special tools and provide multiple annotation formats to ensure high quality annotations.

[2] http://www.ultimate-guitar.com/

3. DATA COLLECTION PROCESS

Hexaphonic pickups are magnetic pickups that have individual outputs for each magnet. We ordered a clip-on hexaphonic pickup from ubertar.com, which has 6 individual single coil magnets, and is manually attached to an acoustic guitar. For better pickup signal-to-noise ratio (SNR), nickel wound steel strings are used for the acoustic guitar.

The audio was recorded in a small, soundproof recording studio with minimal reverberation. In addition to the six channels from the hexaphonic pickup, we also record the guitar using a Neumann U87 condenser microphone, placed $\approx$30 cm in front of the 18th fret of the guitar. This results in seven channels of audio overall.

Six experienced guitarists were recruited to record for this database. All six players have more than 10 years of guitar playing experience, and were recruited by the authors. The guitarists were asked to play 30 twelve to sixteen bar excerpts from lead-sheets in a variety of keys, tempos, and musical genres, described in Section 4. During recording, guitarists were provided with a backing track that consisted of a click track, drum set, and bass line, heard through monitoring headphones. For each excerpt, players were asked to comp (play chords), and then to solo over their own comping. The guitarists were allowed to replay excerpts until they were aesthetically satisfied with their performance.

4. DATASET OVERVIEW

We use the JAMS file format [10] to store the rich collection of annotations for this dataset. For each recording, the JAMS file contains annotations for tempo, key, and style (metadata); beats and downbeats (inferred from the click track); instructed chords (from the lead-sheets); performed chords (via automatic estimation); note-level transcription, including string and fret position (via automatic estimation), onsets (via annotation), offsets (via automatic estimation) and pitch contour for each note (via validated automatic estimation). Descriptions of each of these annotation types are detailed in Section 5. Figure 1 gives a visualization of some of the annotations provided for an excerpt of the dataset.

In total, each player provided 30.47 minutes of musical material, resulting in just over 3 hours of content in total. Each player was asked to play 30 excerpts, organized as follows: 3 different chord progressions are paired with each of the 5 different genres, all recorded at two different tempi: slow and fast. The three progressions were the 12 bar blues, Autumn Leaves, and Pachelbel's Canon. The five different genres were Rock, Jazz, Funk, Bossa Nova (BN), and Singer-Songwriter (SS). In order to broaden the chord gamut in GuitarSet, key signatures were independently assigned to each of the 30 excerpts.

5. ANNOTATION METHODS

The hexaphonic recordings are analyzed to generate annotations for each string individually, and a complete tran-

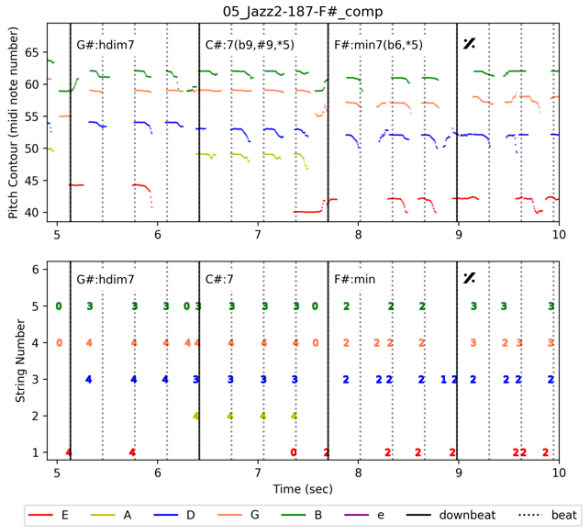

Figure 1. A 5 second excerpt of Jazz comping. Downbeats and beats are indicated with solid and dashed vertical lines respectively. (Top) Played chords and pitch contours, colored by string. (Bottom) Instructed chords (lead sheet) and string/fret positions.

scription of the excerpt is generated by aggregation. For each string, onset/offset time pairs along with continuous pitch tracks are annotated semi-automatically, with manual validation. The validated transcriptions are then used to automatically create derivative annotations, including chords, string and fret number, and more.

We first pre-process the hexaphonic recordings using the KAMIR bleed removal algorithm [17] to reduce noise picked up by the single coil magnets from adjacent strings. We then generate a rough note-level transcription by running pYIN-note [13] over the recording of each string; this rough transcription is used as the starting point for manual validation.

5.1 Note-Level Annotations

We focus our manual annotation efforts on creating note-level annotations, such as the one shown in Figure 1 (Top). Accurately creating or correcting annotations of individual string recordings requires contextual information and musical expertise. For example, an intentionally muted string in a full chord still produces a clear pitch and onset when examining the single muted string in isolation. However, when mixed together with the other more resonant strings, the muted note is completely masked. Because the muted note is neither intended by performer nor heard by listeners, we chose not to annotate it.

In order to address this issue efficiently and maximize automation, we simplify the problem by taking a component approach, and determine the onsets, offsets and pitch tracks sequentially. We first focus on generating high quality onset annotations.By manually validating the onsets, muted notes that shouldn't be included in the annotation and other non-note events are left out of the annotation. Offsets are then automatically estimated, and the resulting note regions are used to facilitate highly accurate pitch track estimations.

5.1.1 Onsets

Given automatically estimated onsets, removing false positive onsets can efficiently be done manually, but accurately adding missed onsets efficiently requires machine assistance. In order to allow annotators to easily add missed onsets, we automatically adjust human-estimated onset times by searching for the most likely spectral flux peak in a local neighborhood. Concretely, for a human estimated onset time $\tilde{a}$, the true onset time a is determined by finding the position for which the windowed onset strength function $G_a(t)$ is maximized.

Let $E(t)$ be the root-mean-squared (RMS) energy calculated at time t, and $N_a(t)$ be the spectral flux novelty function at time t [3]:

$$N_a(t) = \sum_{k=1}^{n/2} H(|X(t,k)| - |X(t-l,k)|) \quad (1)$$

where $H(x) = \frac{x+|x|}{2}$ is the half-wave rectification function and $l = 5.8$ms is a constant lag in time, n is the number of analysis bins, and k is the bin index.

The windowed onset strength function $G_a(t)$ is constructed as follows,

$$G_a(t) = E(t) * N_a(t) * \mathcal{N}(\tilde{a}, \sigma^2) \quad (2)$$

and the onset time is computed as

$$a = \arg\max_t(G_a(t)), \quad (3)$$

where $t \in [\max(a_{prev} + \tau_a, a - 3\sigma), a + 3\sigma]$, $\tau_a = 50$ms and $\sigma = 30$ms. The lower limit on t ensures there are at least τ_a seconds between consecutive onsets. The Gaussian component in $G_a(t)$ ensures the locality of the onset search, favoring proximity with the human estimate. Figure 2 shows an instance of such an adjustment.

5.1.2 Offsets

For all onsets a, the corresponding offset b is estimated automatically, using the following criteria. First the offset novelty $N_b(t)$ is modified slightly from Equation 1:

$$N_b(t) = -\sum_{k=1}^{n/2} H'(|X(t,k)| - |X(t-l,k)|) \quad (4)$$

where $H'(x) = \frac{x-|x|}{2}$ is the negative half wave rectification function and $l = 5.8$ms.

Using the generated offset novelty function, an offset strength function $G_b(t)$ is generated.

$$G_b(t) = \frac{N_b(t) * (\log E(t-l) - \log E(t))}{E(t)} \quad (5)$$

where $t \in [a + \tau_b, a_{next})$ and $\tau_b = 30$ ms. $l = 5.8$ms is the hop length in time of the analysis window.

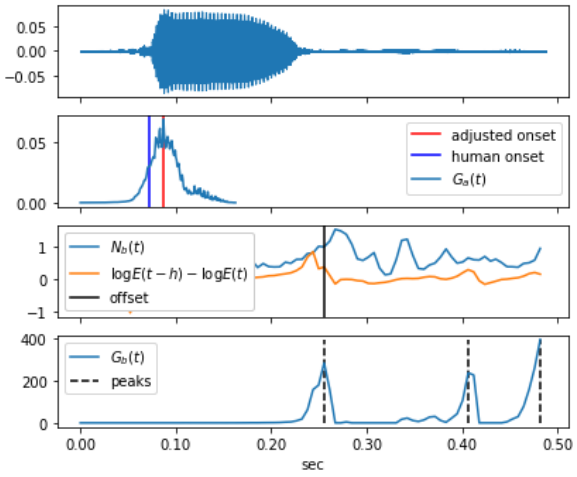

Figure 2. (Top) Waveform of a single note. (Upper Middle) Human estimated onsets, adjusted by examining the onset strength function. (Lower Middle) Offset novelty functions and the detected offset in black. (Bottom) $G_b(t)$ and detected peaks.

The intuition behind the offset strength function $G_b(t)$ is straightforward: log-RMS difference and spectral flux both give peaks for potential offsets, the RMS in the denominator penalizes peaks of $G_b(t)$ that still have significant energy. The peaks of the offset strength function $G_b(t)$ are then thresholded to generate the offset candidates as shown in Figure 2. Offset candidates within 30 ms of the onset a are discarded, and the first offset candidate in time is then chosen from the remaining offsets as b.

5.1.3 Pitch

After onsets and offsets are determined, the pitch tracks of voiced regions are then estimated using pYin. The resulting estimation is then cleaned by the first author in Tony, mostly correcting octave mistakes.

While we annotated the continuous pitch trajectories of each note, the overall center pitches still needs to be inferred. We choose a simple heuristic that averages the pitch track frequencies. For a note with onset at time a and associated pitch track $f(t), t \in [a, b)$; the center pitch of the note p is estimated by taking the average pitch track over a subset of the note region $t \in [a', b')$, where a' and b' are 25% and 50% of the note duration respectively:

$$p = \frac{1}{b' - a'} \sum_{t=a'}^{b'} f(t) \tag{6}$$

We only consider the subset $t \in [a', b')$ to ensure a perceptually relevant average pitch, since the pitch near the onset and offset of a guitar note can sometimes be unstable (e.g. see Figure 1).

5.2 Derivative Annotations

Given note-level annotations, the lead sheets and the click track, we automatically generate a series of derivative annotations.

5.2.1 String and Fret Position

Since the tuning of the guitar is known at the time of data collection, fret positions can be determined simply by finding the difference in semitones between the annotated pitch and the pitch of the open string. A visualization of these annotations is shown in Figure 1 (Bottom).

5.2.2 Chords

Two different types of chord annotations accompany each of the 180 excerpts. The first type of chord annotation is the chord written in the lead sheet that is provided to the guitar players at the time of data collection. However, in order to better fit the given genre, the players often modified the given chords, hereafter called *instructed chords*. Therefore the *performed chords* are not necessarily the same as the instructed chords. Because the backing track contains a bass line that is aligned to the root and the timing of the instructed chords, the instructed and performed chords vary mostly in chord type, not root. The instructed chords have only four types (major, minor, dominant seventh, half-diminished seventh); specific voicings, extensions and alterations could be freely determined by the players without suggestion bias.

We infer the performed chords by combining information from the lead sheet and the annotated notes. In order to make the comparison between the chords as instructed by the lead sheet and the actual performed chords straightforward, the chord segmentation is determined from the lead sheets. A drawback of this approach is that anticipated or lagging chords changes lead to a slight mismatch between the audio signal and the annotations, which may disturb data-driven methods using this data as a training set. However, we argue that such quantization leads to annotations that are more fit for displaying as sheet music and more consistent than human segmentation, which is subjective in this regard [3]. Furthermore, these cases are expected to be rare because of the aforementioned backing track.

For each chord segment, we first determine if a string is active by verifying whether the total duration of all notes played on that string exceeds 5% of the segment duration. This activity thresholding ensures that notes in adjacent chord segments do not accidentally cause otherwise silent strings to appear active simply because of an offset in chord changes between the lead sheet and recording. Next, the predominant note is determined for all active strings per segment. This is done by taking the MIDI note value with the longest total duration per-string (summed over all note repetitions in the chord segment), resulting in a set of up to six notes per chord segment from which we subsequently derive a chord label.

[3] Informal experiments with symbolic chord recognition software resulted in a far worse segmentation.

The root of the chord is also taken from the lead sheet and the inversion naturally arises from the lowest note in the set. Finally, the chord type is determined from the chroma of the set of notes per string through a decision tree that is part of the open-source MusOO library [4]. See Figure 1 for examples of instructed and played chords.

It is notable that our approach of determining the played chord has several biases; namely, the boundary of each chord and the root of each chord is predetermined. More in depth investigation is needed to determine chord boundaries and roots purely automatically, but this is left for future work.

5.2.3 Beats and Downbeats

Since the data is recorded against a click track, the tempo, beats, downbeats and meter of all the excerpts are known. These annotations are generated for each excerpt automatically given this known metadata.

5.3 Inferred Stroke Information

Another pattern that can be recognized from the annotated data is the inter-string onsets. With the help of the onset adjustment step, the annotation captures minute timing differences across onsets on different strings; and by looking at these onset patterns, one can gain a much better understanding of the picking activity that would be otherwise complex to analyze. Figure 3 shows the onsets per string for a short excerpt. Four different strokes can be clearly identified within this 650 ms excerpt. By examining the relative order of strings in each of the strokes, we can clearly observe that the first and last stroke are downstrokes, and the second and third are up-strokes. Evident from Figure 3, the inter-string onsets are only milliseconds apart during fast strokes, and would be very difficult for humans to manually annotate precisely. This nuanced detail would have been lost if the onset adjustment step were not applied.

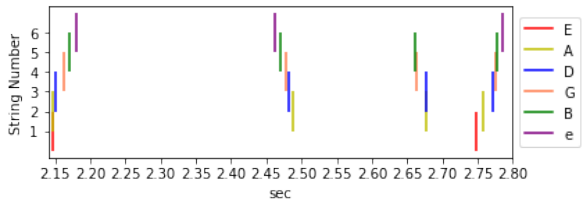

Figure 3. Onsets for each string are shown in different colors.

6. BASELINE EXPERIMENTS

In order to better understand the new challenges posed by this dataset, we evaluate the performance of strong baseline algorithms against our ground truth notes, chords, and beats/downbeats. These experiments are performed without the algorithms seeing any of GuitarSet's data. Detailed results can be found in the GuitarSet repository. [5] All box plots used in this section have box edges showing the first and third quartile, and the whiskers showing 1.5 interquartile range (IQR) away from the box edges.

6.1 Notes

We evaluate the performance of the Deep Salience multiple-f_0 estimation algorithm [4] on GuitarSet's polyphonic rhythmic recordings. Figure 4 shows the results across different splits of the data.

Overall, the model has an accuracy of $\approx 46\%$, and the most common type of error is missed, rather than incorrect, notes. Looking at Figure 4 (Top Left), the results are split by genre, and we see that Jazz is overall the most difficult genre to transcribe (likely due to the more complex chord combinations), while Funk has the highest recall and lowest precision (due to short notes and more unvoiced regions). In Figure 4 (Bottom Left), we see that the audio from the pickup is easier to transcribe than the audio from the microphone, likely because the pickup signal is cleaner.

From Figure 4 (Top Right), we see that the performance varies by player, both in terms of average accuracy and in terms of the variance across all the player's recordings. This suggests that each player's technique or playing style is different enough that algorithm performance differs significantly. Finally, in Figure 4 (Bottom Right), we see the clear trend that the faster the tempo, the more difficult the excerpt is to transcribe.

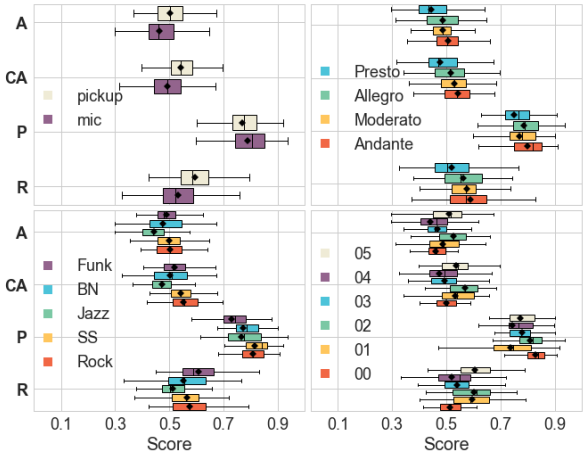

Figure 4. Baseline algorithm multiple-f_0 scores on different splits of GuitarSet. The metrics are `A` (Accuracy), `CA` (Chroma Accuracy), `P` (Precision), and `R` (Recall). (Top Left) Scores split by recording mode. (Top Right) Scores split by excerpt tempo. (Bottom Left) Scores split by genre. (Bottom Right) Scores split by player.

If only the microphone split of the data is considered, we see that the Deep Salience model performs worse on

[4] https://github.com/jpauwels/libMusOO

[5] https://github.com/marl/GuitarSet

GuitarSet than it does on Bach10 and MedleyDB [4]. With the accuracy at only $\approx 43\%$, there are still significant possible performance gains to be had.

6.2 Chords

Next, we evaluate the performance of a state-of-the-art chord recognition baseline [14] against the GuitarSet chord labels. The results, stratified by genre, are show in Figure 5. First, we see that again, some genre's chord labels are easier to estimate than others; in particular, the Rock and Singer Songwriter genres are much easier due to the generally simpler chord types used in those genres compared with the others. Next, we see that there is a large variance in the scores and that there are many outliers. Upon investigating the reason for these outliers, we discovered that some popular guitar textures are not represented in the estimation algorithm's output space. Power chords and octaves, for example, are common guitar textures that are not within the range of typical chord estimation output. While the lead sheet that guides the data collection contains 42 unique chords, the actual detailed chord annotations had a total of 478 unique chord labels (counting all inversions and variations as unique), most of which were small variations of the 42 due to players adding or removing notes.

As shown in Table 1, the overall performance of the baseline chord recognition algorithm on GuitarSet is comparable with the dataset evaluated by Humphrey and Bello [9]. However, as mentioned above, some strata of the dataset are considerably more difficult than the rest.

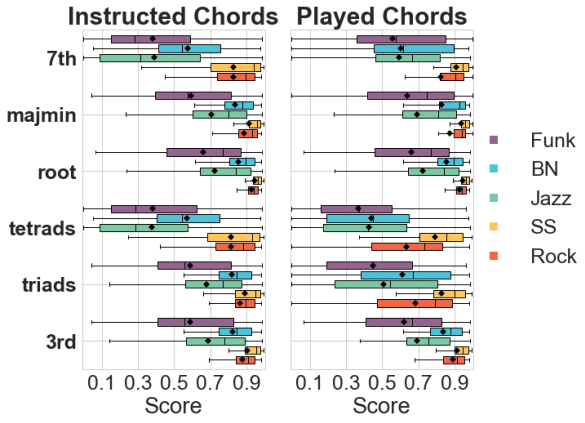

Figure 5. Chord recognition baseline algorithm results on GuitarSet, stratified by genre.

6.3 Beats and Downbeats

The performance of a state-of-the-art beat and downbeat detection algorithm [6] is evaluated on GuitarSet, and the results, stratified by player, are shown in Figure 6. More so than for the previous two tasks, there is a substantial difference between the beat tracker's performance for different players. This suggests that the guitarists have different characteristics in how they play that affect beat detection, such as their choice of strumming patterns or the strength of their attacks. For example, player 00 has a fast strumming style, and plays chords with embedded melodies, which proves difficult for the algorithm.

Dataset	Root	3rds	Triads	7ths	Tetrads
GuitarSet					
— Instructed	0.903	0.862	0.838	0.669	0.619
— Played	0.903	0.866	0.708	0.810	0.544
H. & B. [9]	0.861	0.836	0.812	0.729	0.671

Table 1. Median weighted recall scores for the baseline algorithm [14] performed on different datasets

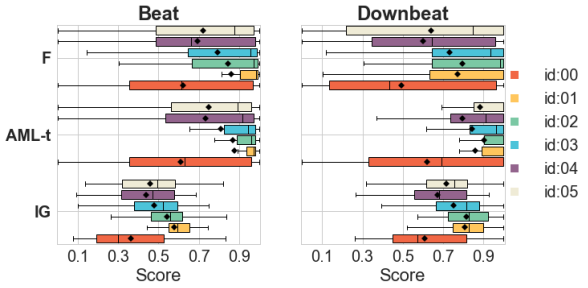

Figure 6. Evaluation of baseline beat/downbeat detection algorithm on GuitarSet, split by player. The metrics are `F` (F-measure), `AML-t` (Any Metric Level-Total), and `IG` (Information gain).

While the median beat and downbeat tracking F-measure is in the 90% range for several players (which is typical for state-of-the art-beat tracking [5]), several substratas of GuitarSet are challenging for beat and downbeat estimation. This is especially true because the tempo and meter do not change over time for each excerpt, yet the data is still challenging for a state-of-the-art beat and downbeat estimation algorithm.

7. CONCLUSIONS

In this paper, we presented a large and carefully annotated dataset of guitar recordings which is available as an open source resource to the research community. We gave a detailed overview of the data collection process and a description of the data itself. Finally, we described our novel process for efficiently and accurately creating note, chord, and beat annotations, and reported the performance of state-of-the-art algorithms on these annotations.

We hope GuitarSet will be useful beyond providing training and evaluation data for transcription models by providing a gateway to investigate interesting problems such as stroke analysis or harmony segmentation. We are pleased to release GuitarSet to the research community and hope that it will foster new, guitar-focused research.

8. ACKNOWLEDGEMENTS

Johan Pauwels has been partly funded by the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/L019981/1 and by the European Unions Horizon 2020 research and innovation programme under grant agreement N° 688382.

9. REFERENCES

[1] Iñigo Angulo, Sergio Giraldo, and Rafael Ramirez. Hexaphonic guitar transcription and visualisation. In *Proceedings of the Second International Conference on Technologies for Music Notation and Representation (TENOR)*, 2016.

[2] Ana M Barbancho, Isabel Barbancho, Lorenzo J Tardón, and Emilio Molina. *Database of Piano Chords: An Engineering View of Harmony*. Springer, 2013.

[3] Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on speech and audio processing*, 2005.

[4] R.M. Bittner, B. McFee, J. Salamon, P. Li, and J.P. Bello. Deep salience representations for f_0 estimation in polyphonic music. In *18th International Society for Music Information Retrieval Conference*, ISMIR, 2017.

[5] Sebastian Böck, Florian Krebs, and Gerhard Widmer. Joint beat and downbeat tracking with recurrent neural networks. In *ISMIR*, 2016.

[6] S. Durand, J. P. Bello, B. David, and G. Richard. Robust downbeat tracking using an ensemble of convolutional networks. *IEEE Transactions on Audio, Speech, and Language Processing*, 25(1):76–89, 2017.

[7] Valentin Emiya, Roland Badeau, and Bertrand 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.

[8] Sebastian Ewert and Mark B Sandler. An augmented lagrangian method for piano transcription using equal loudness thresholding and lstm-based decoding. *arXiv preprint arXiv:1707.00160*, 2017.

[9] Eric J Humphrey and Juan Pablo Bello. Four timely insights on automatic chord estimation. In *ISMIR*, pages 673–679, 2015.

[10] Eric J. Humphrey, Justin Salamon, Oriol Nieto, Jon Forsyth, Rachel M. Bittner, and Juan P. Bello. JAMS: A JSON annotated music specification for reproducible MIR research. In *International Society of Music Information Retrieval (ISMIR)*, October 2014.

[11] Christian Kehling, Jakob Abeßer, Christian Dittmar, and Gerald Schuller. Automatic tablature transcription of electric guitar recordings by estimation of score-and instrument-related parameters. In *DAFx*, pages 219–226, 2014.

[12] Matthias Mauch, Chris Cannam, Rachel Bittner, George Fazekas, Justin Salamon, Jiajie Dai, Juan Bello, and Simon Dixon. Computer-aided melody note transcription using the tony software: Accuracy and efficiency. In *2015 International Conference on Technologies for Music Notation and Representation*, May 2015.

[13] Matthias Mauch and Simon Dixon. pyin: A fundamental frequency estimator using probabilistic threshold distributions. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 659–663. IEEE, 2014.

[14] B. McFee and J.P. Bello. Structured training for large-vocabulary chord recognition. In *18th International Society for Music Information Retrieval Conference*, ISMIR, 2017.

[15] Raymond Vincent Migneco. *Analysis and synthesis of expressive guitar performance*. PhD dissertation, Drexel University, 2012.

[16] Paul D. O'Grady and Scott T. Rickard. Automatic hexaphonic and guitar transcription and using and non-negative constraints. In *IET Irish Signals and Systems Conference (ISSC 2009)*. IET, 2009.

[17] Thomas Prätzlich, Rachel M. Bittner, Antoine Liutkus, and Meinard Muller. Kernel additive modeling for interference reduction in multi-channel music recording. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, May 2015.

[18] Justin Salamon, Rachel M Bittner, Jordi Bonada, Juan José Bosch Vicente, Emilia Gómez Gutiérrez, and Juan P Bello. An analysis/synthesis framework for automatic f0 annotation of multitrack datasets. In *18th International Society of Music Information Retrieval (ISMIR) Conference*, October 2017.

[19] Zhengshan Shi, Kumaran Arul, and Julius O Smith. Modeling and digitizing reproducing piano rolls. In *18th International Society for Music Information Retrieval Conference*, ISMIR, 2017.

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

[21] Michael Stein, Jakob Abeßer, Christian Dittmar, and Gerald Schuller. Automatic detection of audio effects in guitar and bass recordings. In *Audio Engineering Society Convention 128*. Audio Engineering Society, 2010.

[22] Li Su and Yi-Hsuan Yang. Escaping from the abyss of manual annotation: New methodology of building polyphonic datasets for automatic music transcription. In *International Symposium on Computer Music Multidisciplinary Research*, pages 309–321. Springer, 2015.

[23] Li Su, Li-Fan Yu, and Yi-Hsuan Yang. Sparse cepstral, phase codes for guitar playing technique classification. In *ISMIR*, pages 9–14, 2014.