Instrument-centered Music Transcription of Bass Guitar Tracks

Jakob Abeßer[1] and Gerald Schuller[2]

[1]*Semantic Music Technologies group, Fraunhofer IDMT, Ilmenau, Germany*

[2]*Department of applied media systems, Technische Universität Ilmenau, Ilmenau, Germany*

Correspondence should be addressed to Jakob Abeßer (jakob.abesser@idmt.fraunhofer.de)

ABSTRACT

In this paper, we propose an instrument-centered bass guitar transcription algorithm. Instead of aiming at a general-purpose bass transcription algorithm, we incorporate knowledge about the instrument construction and typical playing techniques of the electric bass guitar. In addition to the commonly extracted score-level parameters note onset, offset, and pitch, we also estimate the additional instrument-level note parameters string number, fret number, plucking style, and expression style. The proposed algorithm achieved an F-measure value of 0.901 for the note-wise transcription evaluation on a novel evaluation dataset and outperformed three state-of-the-art bass transcription algorithms.

1. INTRODUCTION

Traditionally, *music transcription* denotes the process of estimating a parametric description of music recordings [10]. These parameters characterize each played note on the abstract level of a musical score, which usually does not properly reflect the (unique) sonic properties of musical instruments. As shown in [16], the performance of automatic melody transcription algorithms did not significantly improve in the last years—a "glass ceiling" seems to be reached. We believe that music transcription algorithms can be improved, if they are well-tuned to the acoustic properties and sound production mechanisms of the instruments that are to be transcribed.

2. GOALS

In this paper, we focus on bass transcription, i.e., the transcription of the lowest musical voice. In particular, instead of aiming at a *general-purpose* transcription algorithm that can be applied to various bass instruments (bass guitar, double bass, bass synthesizer, etc.), we want to implement an *instrument-centered* music transcription algorithm, which is tailored towards the *electric bass guitar* by incorporating knowledge about the instrument construction and the sound production, i.e., typical playing techniques of the instrument. In this paper, we assume that the bass guitar track is *perfectly isolated* without any overlap of other instruments. Separating the bass signal from polyphonic music is not in the scope of this paper. We propose to extend the conventionally used set of *score-level* note parameters (pitch, onset, and offset) by a set of *instrument-level* note parameters. For the electric bass guitar, we focus on *playing techniques* (plucking style and expression style) and *spatial positions on the instrument neck* (defined by fret number and string number) as instrument-level parameters [4, 2]. These additional parameters describe *how* and *where* the musician plays certain notes on the instrument neck, hence they are indented to capture the physical gestures of the musician playing the instrument.

3. CHALLENGES

Several challenges have to be faced. First, since frequency modulation techniques such as *bending*, *vibrato*, and *slide* are investigated, a suitable spectral representation must be found that allows us to track very low fundamental frequencies starting from $f_0 \approx 41\,\mathrm{Hz}$ with a reasonable temporal resolution. Second, the 11 plucking and expression styles analyzed in this paper result in musical notes with very different acoustic and spectral properties ranging from purely harmonic notes to strongly percussive notes with barely any overtone structure. Suitable acoustic features need to be identified that allow to automatically classify between different playing technique classes.

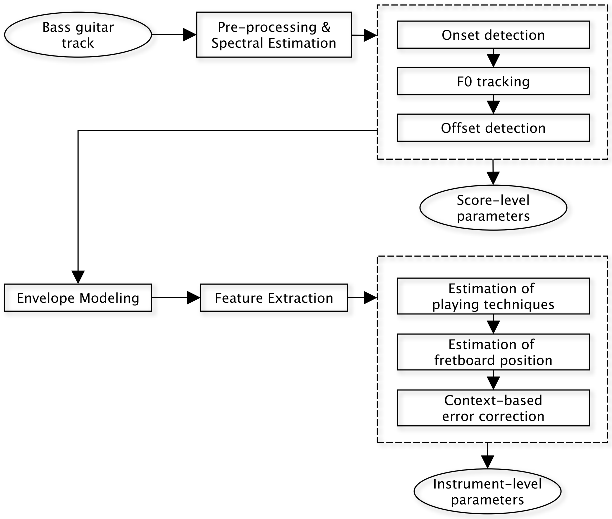
Bass guitar track
Pre-processing & Spectral Estimation
Onset detection
F0 tracking
Offset detection
Score-level parameters
Envelope Modeling
Feature Extraction
Estimation of playing techniques
Estimation of fretboard position
Context-based error correction
Instrument-level parameters

Fig. 1: Processing flowchart

4. BACKGROUND

4.1. Electric Bass Guitar

The bass guitar is a plucked string instrument, which was first developed in the 1930s. Ever since, it was strongly influenced by the electric guitar in terms of instrument construction, sound production, and playing styles.

The physical process of playing the bass guitar can be modeled by *two consecutive performance gestures* of the musician. The first gesture describes the initial plucking of a string using the playing hand, which will be referred to as the *plucking style*. The second gesture describes the way in which the fretting hand is used to manipulate the string vibration, which will be referred to as the *expression style*. Table 1 summarizes all 11 bass guitar plucking and expression styles investigated here. A detailed description of all styles can be found in [4] and [3].

The bass guitar—as well as other string instruments—allows musicians to play notes within a certain pitch range at different *fretboard positions*. The fretboard position defines a location by the string number and the fret number. The most common string tuning is from the lowest to the highest string E1, A1, D2, and G2. Notes played on different strings not only vary in pitch but also in terms of timbre-related properties.

Table 1: Bass guitar plucking styles and expression styles.

Plucking Styles		**Expression Styles**	
Finger-Style	(FS)	Normal	(NO)
Picked	(PK)	Harmonics	(HA)
Muted	(MU)	Dead-notes	(DN)
Slap-Thumb	(ST)	Vibrato	(VI)
Slap-Pluck	(SP)	Bending	(BE)
		Slide	(SL)

5. RELATED WORK

All bass transcription algorithms presented in the literature so far can be categorized as *general-purpose* transcription algorithms, i.e., they are not tailored towards a particular bass instrument. The *bass line* is considered to be the dominant melodic voice in the lower pitch register with fundamental frequencies between around 40 Hz and 400 Hz. Due to the low fundamental frequency range of bass notes, *downsampling* is often applied to the analyzed audio signal to speed up the transcription [5, 8, 15]. At the same time, harmonic components from other instruments in higher frequency ranges are filtered out. Ryynänen and Klapuri estimate a variable, context-dependent upper f_0-limit for the bass line [14]. Using different source separation techniques such as the harmonic/percussion sound separation (HPSS) algorithm, signal components or the percussion instruments [17] are removed in the spectrum before the bass line is transcribed. Spectral whitening can be applied to suppress timbral characteristics of the applied bass instrument on the note envelopes [14]. *Note detection* is often performed either in the time domain [5, 9] by envelope extraction methods or in the frequency domain—usually after several frame-wise f_0 estimates are grouped to note events [13, 14, 8]. Due to its computational efficiency, the Short-time Fourier Transformation (STFT) is the most often used *spectral estimation* method [5, 9, 13, 14]. Other spectral representations such as the instantaneous frequency (IF) spectrogram [5, 8] or the constant-Q spectrogram [18] are computed to improve the achievable frequency resolution in the lower frequency bands. In order to estimate the frame-wise fundamental frequency, a *harmonic saliency function* can be computed, which provides a likelihood-measure for different f_0-values. Klapuri and Ryynänen propose to compute the harmonic salience of a f_0 candidate by summing up the spectral energy at the frequency bins of the corresponding harmonic frequencies [13]. Salomon and Goméz extract a saliency function from the mid-level chromagram-based representation of the Harmonic Pitch Class Profile (HPCP) in [15]. The HPCP is computed in the bass frequency band between 32.7 Hz and 261.6 Hz, using a rather high resolution of 120 bins per octave. Ryynänen and Klapuri present a hybrid transcription framework for bass and melody transcription in polyphonic music [14] by combining two modeling strategies, an acoustic note modeling and a musicological model of likely note transitions. Goto proposed the "PreFEst" (predominant-F0 estimation) algorithm in [8], which is used for a combined transcription of the main melody and the bass line: First, frequency components are extracted by using a STFT-based multi-resolution filter bank and computing the instantaneous-frequency (IF). The overall spectrogram is modeled as a weighted sum of different tone models, which are combined probability density functions (PDFs) of fundamental frequency components and the corresponding overtones. Based on the extracted harmonic saliency function, the most salient peaks are tracked over time and grouped to note events.

6. NEW APPROACH

6.1. Development data sets

Two development sets *DS-1* and *DS-2* were used for parameter optimization. Based on the *IDMT-SMT-Bass* dataset (previously published in [4]), development set *DS-1* comprises 550 randomly selected isolated bass guitar notes (50 notes for each plucking and expression technique) and *DS-2* comprises 1711 notes from the *IDMT-SMT-Bass* dataset, which were recorded with the Fame Baphomet 4 NTB bass guitar, which was also used to record the bass lines in the evaluation dataset introduced in Section 7.1.

In the following subsections, all processing steps of our proposed bass guitar transcription algorithm as illustrated in Figure 1 will be detailed.

6.2. Pre-processing & Spectral Estimation

First, we convert the audio signal to a monaural signal if necessary and down-sample to a sampling frequency of $f_s \approx 5.51$ kHz. Two different spectral representations are extracted. First, a *Short-time Fourier Transform (STFT) spectrogram* X is computed using a blocksize of $b = 512$ and hopsize of $h = 32$. The STFT spectrogram is used for the envelope modeling as will be explained in Section 7.3. Second, a *reassigned spectrogram* X_{IF} based on the instantaneous frequency (IF) is computed with the same values of b and h. The method proposed by Abe et al. in [1] is applied. We use a logarithmic frequency axis with a resolution of 120 bins per octave in the range between 29.1 Hz and $f_s/2$. In each time frame t, the magnitude values of the STFT magnitude spectrogram are reassigned and accumulated towards the logarithmic frequency bins that correspond to the IF values at the original frequency positions (as for instance shown in the upper plot in Figure 3)[1]. The *IF spectrogram* is used for onset detection and f_0 tracking as will be shown in Section 6.3 and Section 6.4.

6.3. Onset Detection

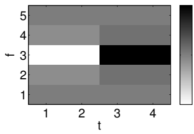

Fig. 2: Harmonic novelty kernel for onset detection

We obtain an onset detection function that measures the *harmonic novelty* as follows. The IF spectrogram X_{IF} is convolved using a two-dimensional kernel K. The kernel has two properties: a filtering of sparse components along the frequency axis (presumably overtones) and a detection of rising magnitude slopes along the time axis (presumably onsets). The kernel is generated via matrix multiplication $K = \begin{bmatrix} 0 & 0.1 & 1 & 0.1 & 0 \end{bmatrix}^T \times \begin{bmatrix} -1 & -1 & 1 & 1 \end{bmatrix}$. A 2D convolution is applied between the IF spectrogram and the kernel as $X_{\text{IF,K}} = X_{\text{IF}} * K$. The onset detection function $o(t)$ is obtained by summing up over all frequency bins of $X_{\text{IF,K}}$ as $o(t) = \sum_f X_{\text{IF,K}}(f,t)$. Onsets t_{on} are detected at all local maxima of $o(t)$ greater than $o_{\min} = 0.2 \max_t o(t)$.[2] This empirical threshold was found based on a *development set* of 550 isolated bass guitar notes with manually annotated onset positions by maximizing the F-measure ($FM = 0.95$).

[1] The mapping from the continuous IF to the discrete logarithmic frequency scale is performed in order to perform the cross-correlation as will be explained in Section 6.4.

[2] However, this approach of using a fixed threshold could lead to missed note events for recordings with a large dynamic range.

Using the development set *DS-1*, we found that the harmonic novelty approach outperformed an onset detection based on the spectral flux ($FM = 0.91$). Figure 3 shows the IF spectrogram of an excerpt of a bass line in the upper plot and the onset detection function $o(t)$ in the lower plot with the detected onset positions. All detected note onsets are further investigated as note candidates.

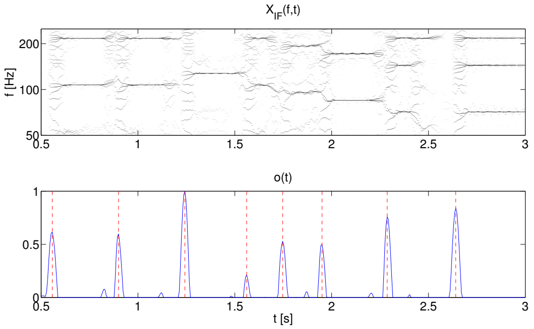

Fig. 3: Bass line excerpt: IF spectrogram X_{IF} (upper plot), onset detection function $o(t)$ (solid blue line, lower plot), detected onset positions t_{on} (dashed red line)

6.4. f_0-tracking & Offset Detection

Two processing steps are performed: pre-estimation of the note's f_0 and f_0 tracking. First, for the n-th note candidate, the spectral frames in X_{IF} are averaged over the first 20 % of the time frames between the onset positions $t_{\mathrm{on}}(n)$ and $t_{\mathrm{on}}(n+1)$ to obtain an *accumulated spectrum* $X_{\mathrm{IF,acc},n}(f)$. We focus on the beginning period to prevent a smearing of harmonic peaks for notes played with modulation techniques such as bending, vibrato, and slides, which have a time-varying note fundamental frequency. The pre-estimate $\hat{f}_{0,n}$ is detected at the frequency bin with the highest cross-correlation between $X_{\mathrm{IF,acc},n}(f)$ and a *harmonic comb filter* $c(f)$, which has combs at the harmonic frequency positions $f_k \approx f_0(k+1)\sqrt{1+\beta(k+1)^2}$ [6] on a logarithmic frequency axis (as used for $X_{\mathrm{IF},n}$). The inharmonicity coefficient was set to $\beta = 3E-4$, which is an average over multiple notes of *DS-1*. Using 500 notes from the development set *DS-1* (the 50 notes played with the *dead-note* expression style were excluded since they are percussive without a perceivable stable pitch), we compared comb filters with a varying number of harmonic peaks for the task of pitch detection. Furthermore, we compared comb filters with peaks having unit magnitudes and comb filters with doubled magnitude on the first two peaks. As shown in Figure 4, a comb filter with 10 combs and emphasis on the first two peaks achieved the highest pitch detection accuracy, i.e., the percentage of correctly identified note pitches (based on the rounded f_0) of 0.98 on the development set. This configuration is illustrated in Figure 5. We did not observe an improvement in pitch detection accuracy by using linearly decaying values for the filter peaks.

The frame-wise f_0-tracking is initialized at the frame t_{Start}, which was chosen to be in the middle of the period used for averaging the spectrogram (as explained above). The tracking is performed over adjacent frames in two directions—backwards until reaching the note onset and forwards until reaching the following onset or the last frame. We use a *continuity-constraint*, i.e., in each frame, we only consider the frequency bins around the f_0 bin from the preceding frame as potential f_0 candidates. Again, the highest cross-correlation between the spectral frame and the comb filter is retrieved.

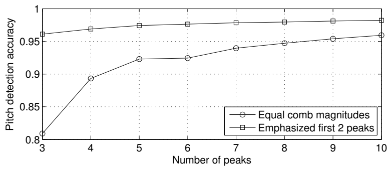

Fig. 4: Pitch detection accuracy over 500 isolated notes (including all plucking and expression styles but *dead-notes*). Circles indicate accuracy values obtained with comb filters with unit magnitude, squares indicate comb filters with doubled magnitude for the first two peaks.

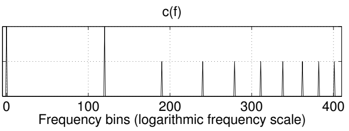

Fig. 5: Optimal harmonic comb filter $c(f)$ based on a logarithmic frequency axis.

The maximum cross-correlation value is stored for each frame—high values indicate a harmonic magnitude characteristic of the spectrum, low values indicate a percussive, wide-band characteristic. We determine the *offset* *position* $t_{\mathrm{off}}(n)$ of each note, where the maximum cross-correlation value remains below a threshold of 0.05 for at least 4 frames or a new note begins.

6.5. Spectral Envelope Modeling

After detecting the onset and offset positions t_{on} and t_{off} as well as tracking the fundamental frequency slope $f_0(t)$, we aim to model the spectral magnitude envelope of a given note using a simple parametric model. Here, we focus on the fundamental frequency and the overtones and neglect wide-band noise-like signal components such as the attack transients. The main motivation is to parametrize all possible spectral envelopes of bass guitar notes using a simple model, which can be used for feature extraction. Each time frame in the STFT magnitude spectrum $X(f,t)$ is modeled as a sum of magnitude-scaled atom functions $h_X(f)$ shifted in frequency, which represent the harmonic components:

$$X(f,t) \approx \sum_{k=0}^{N_{\mathrm{Harm}}} a_k(t)\, h_X(f - f_k(t)) \qquad (1)$$

The atom function $h_X(f)$ is the Fourier transform of the Hanning window $h(t)$, which is applied in the time domain to compute the STFT spectrogram X. We initially truncate $h_X(f)$ outside its first two side-lobes and normalize it to unit magnitude. The harmonic frequencies of string instruments such as the bass guitar follow an inharmonic relationship as discussed before. For the sake of convenience, we initially compute the inharmonicity coefficient β at t_{Start} in the beginning of the note decay part (compare Section 6.4) and assume it to be constant over the duration of the note. We perform a grid search within $\hat{\beta} \in [0, 0.001]$. For each estimate of $\hat{\beta}$, we compute the hypothetical harmonic frequencies $\hat{f}_k$. Then, we apply linear interpolation to estimate the spectral magnitudes $X(\hat{f}_k, t)$, and sum up the magnitude values to obtain a likelihood-value for $\hat{\beta}$. By maximizing the likelihood-value, the optimal value for β is retrieved.

Starting from the frame t_{Start}, a frame-wise optimization is performed by stepping forward and backward in time to determine the overtone magnitudes $a_k(t)$. In each frame, the optimal parameter set $(a_k(t), \beta, f_0(t))$ obtained in the previously computed frame is used as initialization for the Expectation-Maximization (EM) algorithm [11]. We use $f_0(t)$ obtained from the f_0-tracking as detailed in the previous section. For the EM algorithm, the magnitude spectrogram is normalized to a probability density function and the obtained parameters are rescaled after the optimization. We use 5 iterations for the starting frame t_{Start} and 2 iterations for each of the remaining frames. After the envelope modeling, each note is described by a set of envelope parameters $[a_k(t), \beta, f_0(t)]$, which are then used for feature extraction as will be described in the following sections. In Figure 6, an example of the note modeling is shown for a bass guitar note with vibrato. The magnitude envelopes of the overtones were well-captured (including the typical phenomena of *string beating*), attack transients were neglected due to the discussed modeling approach.

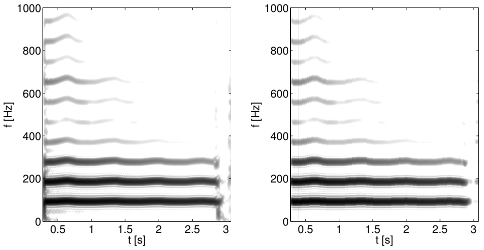

Fig. 6: STFT spectrogram $X(f,t)$ of a vibrato note: original (left) and modeled (right). The start frame for the optimization is shown as vertical line in the right figure.

6.6. Feature Extraction

Audio features are extracted on three different levels: *note-wise* features are extracted once for each note event, *frame-wise* features are computed in multiple time frames of each note event, and *envelope-wise* features are extracted over frequency and magnitude envelopes of individual harmonic components of a note event (either the fundamental frequency or an overtone). In order to segment the note magnitude envelope, we consider a two-stage model that consists of an *attack part*, which is characterized by a rapidly increasing magnitude envelope and a *decay part*, which is characterized by a decaying magnitude envelope.

6.6.1. Frame-wise features

The first group of frame-wise features are extracted in all time frames of a note based on the *overtone's magnitudes* $a_k(t)$ and *frequencies* $f_k(t)$. As previously detailed in [2], we compute as features the relative harmonic magnitudes $a_{\mathrm{rel},k} = a_k(t)/a_0(t)$, the estimated linear slope of

$a_{\mathrm{rel},k}$ over the harmonic index k, and the inharmonicity coefficient $\beta(t)$. We compute the hypothetical harmonic frequencies $\hat{f}_k(t)$ from the fundamental frequency $f_0(t)$ and the inharmonicity coefficient $\beta(t)$. We compute features from the normalized frequency deviations between the theoretical harmonic frequencies $\hat{f}_k(t)$ and the measured harmonic frequencies $f_k(t)$. The second group of frame-wise features characterize the *instrument noise*. Playing styles such as *slap-thumb*, *slap-pluck*, and especially *dead-notes* have a percussive, wide-band characteristic in the attack part of the spectrum. Therefore, we compute a feature to measure the wide-band noise in between the overtones as follows: We remove the harmonic components from a spectral frame $X(f,t_x)$ using a comb-filter, which is tuned to the current fundamental frequency $f_0(t_x)$. Then, we compute the sum over all magnitude bins and compute the ratio to the magnitude sum in the original spectral frame as feature.

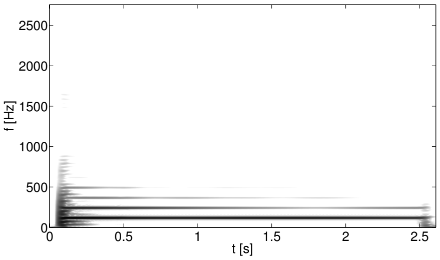

Fig. 7: STFT spectrogram $X(f,t)$ of a note played with the *harmonics* expression style (E string, 7th fret).

Figure 7 illustrates a note played with the *harmonics* expression style on the low E string ($\tilde{f}_0 = 41.2$ Hz). The fundamental frequency of the open string can be seen in the beginning of the note decay part (between 0.1 s and 0.25 s). However, due to the string damping, which is characteristic for this expression style, only the third vibration mode and its octaves remain audible after approx. 0.25 s. Hence, the perceived fundamental frequency of the note is three octaves higher ($f_0 = 3\tilde{f}_0 = 123.6$ Hz). In order to distinguish notes played with the *harmonics* expression style from notes played with other expression styles but having the same perceived fundamental frequency, we compute features to characterize the spectral energy and various *subharmonic frequency positions* (given the fundamental frequency f_0, the aforementioned harmonic components in the beginning of the note decay parts are considered as *subharmonics*). In order to extract *mode likelihood-values* that indicate the presence of subharmonics, we construct comb-filters that are tuned to different virtual fundamental frequencies $f_{0,\mathrm{virtual}}(m) = f_0/m$ with $m \in [2,7]$ (the highest vibration mode, which is played with the *harmonics* style in the dataset is $m = 7$). At the same time, these comb-filters are modified in such way that they have no peaks at multiples of the "real" fundamental frequency. We obtain a likelihood-value for $f_{0,\mathrm{virtual}}(m)$ by filtering the spectrum using the modified comb-filter (in the same way as the measure for the instrument noise was computed). The third group of features are *string likelihood values*. If *harmonics* were played on a particular string, then the energy of the harmonic peaks is likely to be captured by the comb-filter that is tuned to the open string fundamental frequency. We compute this feature for all 4 strings.

6.6.2. Note-wise features

We obtain several timbre-related note-wise features, which were shown to perform very well for the classification of different plucking styles in [4]. We approximate the note magnitude envelope as increasing linear function in the note attack part and as exponentially decaying function in the note decay part. Furthermore, we compute the tristimulus, the odd-to-even-ratio, and the spectral irregularity from the harmonic magnitudes $a_k(t)$ on a frame-level and obtain different statistical measures as features by aggregating the frame-wise feature values over the attack and decay part. We perform the same two-part aggregation over the frame-wise low-level features f_0-adjusted spectral centroid, spectral crest factor (also as first derivative), spectral roll-off, spectral slope, and spectral spread to obtain further features [7].

6.6.3. Envelope-wise features

From the temporal progression of the fundamental frequency $f_0(t)$, we obtain the modulation frequency, a dominance measure of modulation, the number of modulation periods, as well as the pitch difference between the beginning and the end part of the note decay part [3]. These features have been shown to discriminate well among different frequency modulation techniques such as *bending*, *vibrato*, and *slide*. In total, we obtain a 210-dimensional feature vector for each note event.

6.7. Estimation of Plucking Style, Expression Style, and String Number

Based on the estimated feature vector, we use three classification models (trained with the complete development set—approx. 1700 single notes) to automatically classify the plucking style, the expression style, and the string number. For each classifier, we first normalize the feature values to zero mean and unit variance. Second, the supervised feature selection method Inertia Ratio Maximization using Feature Space Projection (IRMFSP) [12], which takes the class labels into account, is applied to reduce the dimensionality of the feature space to $D = 60$. Third, the feature space transformation method Linear Discriminant Analysis (LDA) is applied to further reduce the dimensionality of the feature space to $D = N_{\text{classes}} - 1$. Finally, a Support Vector Machine (SVM) classifier with a radial basis function (RBF) kernel is trained for each of the three classification tasks [4]. The one-against-one multi-class SVM algorithm is used. The fret number is derived depending on the expression style, three cases are differentiated. For *dead-notes*, the fret number is not considered to be relevant and the string number is set to the string number of the closest note, which was played in one of the expression styles *normal*, *vibrato*, *bending*, or *slide* (this makes the bass line easier to play). For notes played with the *harmonics* style, we first obtain the string number by maximizing the aforementioned string likelihood values and mode number by maximizing the mode likelihood values. Since *harmonics* with a given mode can be played on multiple fret positions, we set the fret number to be preferably close to the fret numbers of previous notes based on the estimated mode $\hat{m}$. For all other techniques, the fret number is derived from the pitch and the string number as explained in [2].

6.8. Context-based Error Correction

We use the following constraints in order to improve the estimation of the instrument-related parameters. First, as shown in [2], before the string number is classified, we set all class probability values of those string classes, on which a given pitch P cannot be played to zero. This allows us to avoid meaningless fretboard positions that cannot be played. Second, we partition all plucking styles into the four classes *finger-style*, *muted*, *picked*, as well as *slap*, which includes *slap-pluck* and *slap-thumb*. In this evaluation, we assume that for each bass line, only one plucking style class is present, which is a reasonable simplification for most bass lines in music practice. Hence, we first accumulate all plucking style class probabilities over all notes and determine the most likely plucking style class. Then, all plucking styles are set accordingly. For the *slap* class, we obtain the plucking style from the most likely style *slap-pluck* or *slap-thumb*.

7. EVALUATION

7.1. Dataset

The evaluation of the proposed methods is performed under idealized conditions. The previously published *IDMT-SMT-BASS-SINGLE-TRACKS*[3] dataset is used for evaluation. It consists of 17 bass lines that cover different music styles (blues, rock, funk, bossa nova, and hip hop). The bass lines consist of around 1000 notes and cover all discussed plucking and expression styles as well as all 4 strings of the bass guitar. Figure 8 illustrates a pitch histogram over the evaluation dataset. The notes with a MIDI pitch above 48 are played with the *harmonics* expression style.

All bass lines were recorded with the same electric bass guitar as the notes in the development set. Therefore, the conditions are idealized in such way that the same instrument was used to record the development set, which was used to optimized the proposed algorithm, and the evaluation set, which was used to compare the method's performance to other bass transcription algorithms that were not optimized to this data. However, we believe that the experimental results are informative since these conditions are for instance given in music education software, which can be optimized over time with recordings of the user's instrument in order to improve the transcription performance.

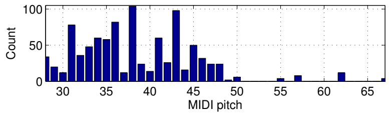

Fig. 8: Pitch histogram over the evaluation data set.

7.2. Algorithms

The presented algorithm (denoted as **A̲**) was compared to the following state-of-the-art bass transcription algorithms by Ryynänen & Klapuri [14] (**R**), Salomon [15] (**S**), and Dittmar et al. [5] (**D**). Algorithm **S** is limited

[3] see http://www.idmt.fraunhofer.de/en/Departments_and_Groups/smt/bass_lines.html

to a two-octave pitch range between the MIDI pitch values 21 and 45 (fundamental frequency values between 27.5 Hz - 110 Hz).

7.3. Evaluation measures

The algorithms **R**, **D**, and **A** provide note-wise transcription results, hence each detected note is characterized by its onset and offset position as well as its MIDI pitch. The algorithm **S** provides only frame-wise estimates of the fundamental frequency. Therefore, only the frame-wise evaluation results can be computed here. For the algorithms **R**, **D**, and **A**, frame-wise fundamental frequency values are obtained using the same temporal resolution of $\Delta t = 5.8$ ms as in **S**. The *instrument-level* evaluation measures are only computed for algorithm **A**.

7.3.1. Score-level Evaluation

As *note-wise evaluation measures*, we use the Recall *REC*, the Precision *PRE*, and the combined F-Measure *FM*. The Recall is defined as the number of correctly transcribed notes divided by the total number of reference notes. The Precision is defined as the number of correctly transcribed notes divided by the total number of transcribed notes. A note is considered as correctly transcribed, if it can be assigned to a reference note with the same MIDI pitch and if it's onset has a maximum absolute deviation to the ground truth of 150 ms as proposed in [13]. As *frame-wise evaluation measures*, we use five global measures first used in the MIREX 2005 competition for melody extraction as explained for instance in [16]—Voicing Recall Rate *VRC* (proportion of correctly detected ground-truth melody frames), Voicing False Alarm Rate *VFAR* (proportion of ground-truth non-melody frames mistakenly detected as melody frames), Raw Pitch Accuracy *RPA* (proportion of detected melody frames with the correct pitch), Raw Chroma Accuracy *RCA* (proportion of detected melody frames with the correct pitch, octave errors are ignored), as well as Overall Accuracy *OA* (combined performance measure for pitch estimation and voicing detection).

7.3.2. Instrument-level Evaluation

In order to evaluate the estimation of the instrument-related parameters plucking style (PS), expression style (ES), and string number (SN), three classification experiments were performed as follows. In order to eliminate the onset and pitch estimation as potential error sources, we use ground truth annotations for the note pitch, onset, and offset instead. The three classifiers are trained with notes from the development set *DS-2*. For the PS classifier, all notes from *DS-2* were selected that were played with the *normal* expression style (NO). For the ES classifier, all notes were used that were played with the *finger-style* (FS) plucking style. For the SN classifier, we chose all notes that were not played with the *dead-note* (DN) nor the *harmonics* (HA) expression style. The evaluation of string number classification from *harmonics* notes was performed here.

8. RESULTS

8.1. Score-level Evaluation

The results of the *frame-wise score-level evalution* are shown in Table 2. Algorithm **S** outperforms the others in the detection of voiced frames with the highest Voicing Recall Rate of $VRC = 0.934$. In terms of pitch estimation, the proposed algorithm **A** outperforms the others with a Raw Pitch Accuracy of $RPA = 0.765$. However, if the octave information is neglected, algorithm **S** shows the best performance for the Raw Chroma Accuracy with $RCA = 0.82$. Keeping in mind that the algorithm **S** only considers a limited pitch range of two octaves, a better performance of **S** for the Raw Chroma Accuracy is likely if a larger pitch range would be considered. Table 3 illustrates the results of the event-wise score-level evaluation. Here, the proposed algorithm **A** clearly outperforms the other two algorithms **R** and **D** in recall ($REC = 0.897$), precision ($PRE = 0.908$), and F-measure ($FM = 0.901$). While **R** and **D** show comparable precision values, **R** clearly has the higher recall value in the direct comparison.

Table 2: Frame-wise evaluation results (best scores are given in bold print).

Algorithm	Evaluation Measures				
	VRC	*VFAR*	*RPA*	*RCA*	*OA*
R	0.835	**0.209**	0.696	0.794	0.728
S	**0.934**	0.296	0.701	**0.82**	0.698
D	0.741	0.291	0.585	0.624	0.606
A	0.89	0.427	**0.765**	0.796	**0.735**

8.2. Instrument-level Evaluation

The confusion matrices for the estimation of the instrument-related parameters PS, ES, and SN are shown in Figure 9. For PS and SN classification, a main diagonal is clearly visible—mean classification accuracy

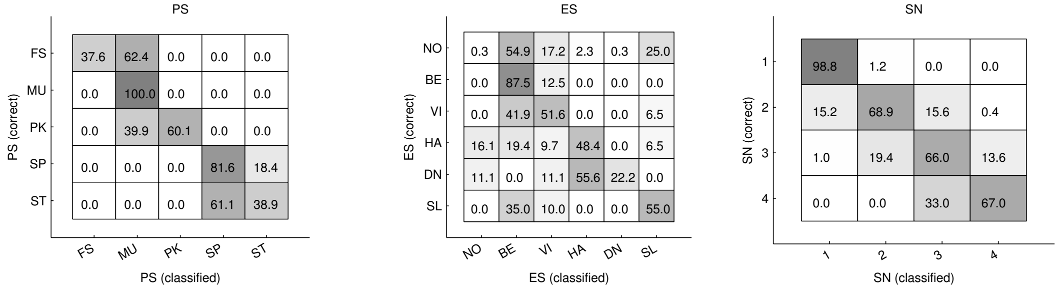

Fig. 9: Confusion matrices for the estimation of instrument-level parameters. All values given in percent.

Table 3: Event-wise evaluation results (best scores are given in bold print).

Algorithm	Evaluation Measures		
	REC	*PRE*	*FM*
R	75.1	84.1	78.7
D	51.2	81.5	59.9
A	**89.7**	**90.8**	**90.1**

values of 0.64 and 0.75 were achieved. For the ES classification, only the BE class shows satisfying results. The other classes—especially NO and DN show strong confusions towards other classes. The mean accuracy for ES classification is 0.44.

As shown before in [4] and [2], the presented approach of feature-based classification of the instrument-related parameters achieves very high classification results if isolated notes are used for training and prediction. Due to the rhythmic structure of the bass lines in the evaluation set, note durations are much shorter than in the training set. We assume that this causes the strong confusion of FS notes towards the MU expression style class. Also, while the training set contains only 11 different combinations of plucking and expression, the evaluation set includes 19 different combinations, which lead to a greater variety in different instrument sounds and which makes it very challenging for the classifier models to make the right class predictions. A possible solution to overcome these problems is to include shorter notes from different bass lines into the training set. The prominent misclassification of NO, VI, and SL notes to the bending class (BE) is likely due to incorrect f_0 tracking results, which would affect the envelope features that allow to discriminate between the frequency modulation techniques.

9. CONCLUSIONS

We proposed an *instrument-centered bass guitar transcription algorithm.* The evaluation was performed under idealized conditions as discussed before. The proposed algorithm outperforms existing state-of-the-art bass algorithms for both frame-wise and event-wise evaluation. However, even with the achieved F-measure value of $FM = 0.901$, the task of monophonic bass transcription cannot be considered as completely solved. Especially the use of different plucking and expression styles lead to very diverse acoustic characteristics of the analyzed note events in the bass guitar track. This complicates the precise detection of the discussed note parameters—both on the score-level and the instrument-level. The current limitations of the proposed algorithm is that it would require a preceding source separation stage to be applicable to polyphonic, multi-timbral music. Future experiments must include the offset into the score-level evaluation and also test the performance of the proposed method on isolated bass guitar recordings from other instrument models.

10. ACKNOWLEDGEMENTS

The authors would like to thank Christian Dittmar, Justin Salamon, Anssi Klapuri, and Matti Ryynänen for providing transcription results and the reviewers for giving helpful comments.

11. REFERENCES

[1] T. Abe, T. Kobayashi, and S. Imai. Robust Pitch Estimation with Harmonics Enhancement in Noisy Environments based on Instantaneous Frequency. In *Proceedings of the* 4^{th} *International Conference*

on Spoken Language Processing (ICSLP), Philadelphia, PA, USA, 1996.

[2] J. Abeßer. Automatic String Detection for Bass Guitar and Electric Guitar. In M. Aramaki, M. Barthet, R. Kronland-Martinet, and S. Ystad, editors, *From Sounds to Music and Emotions - 9th International Symposium, CMMR 2012, London, UK, June 19-22, 2012, Revised Selected Papers*, volume 7900, pages 333–352, London, UK, 2013. Springer.

[3] J. Abeßer, C. Dittmar, and G. Schuller. Automatic Recognition and Parametrization of Frequency Modulation Techniques in Bass Guitar Recordings. In *Proceedings of the 42nd Audio Engineering Society (AES) International Conference on Semantic Audio*, pages 1–8, Ilmenau, Germany, 2011.

[4] J. Abeßer, H. Lukashevich, and G. Schuller. Feature-based Extraction of Plucking and Expression Styles of the Electric Bass Guitar. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2290–2293, Dallas, USA, 2010.

[5] C. Dittmar, K. Dressler, and K. Rosenbauer. A Toolbox for Automatic Transcription of Polyphonic Music. *Proceeding of the Audio Mostly Conference*, pages 58–65, 2007.

[6] N. H. Fletcher and T. D. Rossing. *The Physics Of Musical Instruments*. Springer, New York, London, 2nd edition, 1998.

[7] Geoffroy Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Technical report, Ircam, Analysis/Synthesis Team, Paris, France, 2004.

[8] M. Goto. A Real-Time Music-Scene-Description System - Predominant-F0 Estimation for Detecting Melody and Bass Lines in Real-World Audio Signals. *Speech Communication*, 43(4):311–329, Sept. 2004.

[9] S. W. Hainsworth and M. D. Macleod. Automatic bass line transcription from polyphonic music. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 431–434, La Habana, Cuba, 2001.

[10] A. Klapuri and M. Davy, editors. *Signal Processing Methods for Music Transcription*. Springer Science+Business Media, 2006.

[11] T. K. Moon. The Expectation Maximization Algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, 1996.

[12] G. Peeters and X. Rodet. Hierarchical gaussian tree with inertia ratio maximization for the classification of large musical instruments databases. In *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx)*, pages 1–6, London, UK, 2003.

[13] M. Ryynänen and A. Klapuri. Automatic Bass Line Transcription from Streaming Polyphonic Audio. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP')*, pages 1437–1440, Honolulu, Hawaii, USA, 2007.

[14] M. P. Ryynänen and A. Klapuri. Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, 32:72–86, Apr. 2008.

[15] J. Salamon and E. Gómez. A Chroma-Based Salience Function for Melody and Bass Line Estimation From Music Audio Signals. In *Proceedings of the 6th Sound and Music Computing Conference (SMC)*, pages 23–25, Porto, Portugal, 2009.

[16] J. Salamon, E. Gómez, D. P. Ellis, and G. Richard. Melody Extraction from Polyphonic Music Signals : Approaches, Applications and Challenges. *IEEE Signal Processing Magazine*, 2013.

[17] E. Tsunoo, N. Ono, and S. Sagayama. Musical Bass-Line Pattern Clustering and its Application to Audio Genre Classification. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, pages 219–224, Kobe, Japan, 2009.

[18] E. Tsunoo, G. Tzanetakis, N. Ono, and S. Sagayama. Beyond Timbral Statistics: Improving Music Classification Using Percussive Patterns and Bass Lines. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):1003–1014, 2011.