Guitar Music Transcription from Silent Video

Shir Goldstein
shirgoldstein@gmail.com

Efi Arazi School of Computer Science
The Interdisciplinary Center
Herzliya, Israel

Yael Moses
yael@idc.ac.il

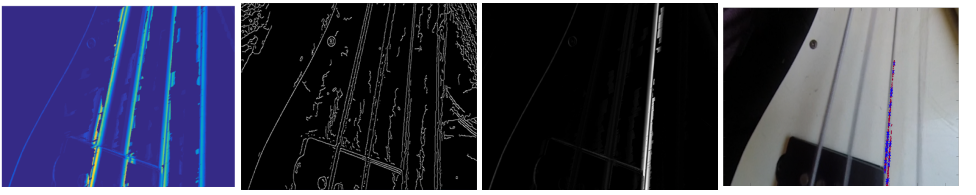
Abstract

Musical note tracking (NT), identifying the pitch of played notes and their temporal information, is typically computed from audio data. Although audio is the natural source of information for NT, audio-based methods have limitations, mostly for polyphonic music analysis. When a string instrument is played, each of its strings vibrates at a certain frequency, producing a sound wave. We propose a novel, physics-based method for polyphonic NT of string instruments. First, the string vibrations are recovered from silent video captured by a commercial camera mounted on the instrument. These vibrations are also used to detect the string locations in the video. The NT of each string is then computed from a set of 1D signals extracted from the video. Analyzing each string separately allows us to overcome the limitations of audio-based polyphonic NT. By directly considering the expected frequencies of the played notes, their aliases, and their harmonics, we can overcome some limitations posed by the relatively low sampling rate of the camera. For a given frame rate, we analyze the set of notes that cannot be detected due to noise as well as indistinguishable pairs of notes. Our method is tested on real data, and its output is sheet music that can allow musicians to play the visually captured music. Our results show that the visual-based NT method can play an important role in solving the NT problem.

1 Introduction

Music transcription is the task of notating a musical piece to create a music score (or sheet music). Many musical pieces are not notated by the composer and have to be manually transcribed based on listening alone. Manually transcribing music is a non-trivial task even for skilled musicians and can be time consuming and inaccurate. An automatic mechanism would thus be convenient. Existing automatic music transcription (AMT) methods mostly consider auditory data, which is the natural source of information for music. The output typically consists of the pitch, onset, and duration of each played note of the musical piece. Thus is usually referred to as Note Tracking (NT).

We propose a method for NT of music played on a string instrument, based on *only visual data* of its strings captured by an ordinary GoPro camera. The input is a silent video captured by a camera mounted on a guitar (other string instruments could also be considered). Our method, which is complementary to audio-based NT, helps to overcome their inherent



(a) (b) (c) (d)

Figure 1: (a) The energy map of a bass string strumming for the 4 known played notes frequencies. (b) The guitar image edge map (no dilation). (c) The energy map of 73.42Hz. (d) A frame captured by the GoPro. The red dots are threshold pixels of (c) and blue are the randomly chosen string-pixels.

limitations. However, new challenges must be addressed since the visual data suffers from low frame rate (FR) and limited frame resolution.

In general, the string vibration causes air pressure fluctuations at the same frequency, that produce the sound wave that we hear. The basic idea behind visual NT is that the string vibration is correlated with the intensity change at a *string-pixel*, a pixel located on the projection of the string to the frame, or very close to it ([27, 32]). Hence, we can use the intensity change of a string-pixel to determine the note played.

NT for monophonic music (only one note playing at a given time) is considered solved by auditory-based techniques, when clean data is available. However, polyphonic NT is considered much harder, in particular for several instruments of the same type (e.g., a string quartet that includes 2 violins) or when there is significant background noise. Another challenge of auditory-based applications is that the same note (pitch) can be produced by several strings (e.g., A2 in a classical guitar can be produced by playing the 5th fret of string E and open string A). Hence, extracting the string played, which is another important aspect of the transcription, is not trivial. Our method makes it possible to overcome these limitations, since each string has a distinct spatial location and hence can be analyzed separately, independent of music played on other strings, other instruments, or background noise.

The proposed method consists of three steps. (i) A novel string detection algorithm that uses the frequency information available from the video (Fig. 1), thus bypassing the challenges of classic geometric-based string detection algorithms. The intensity change function of the string-pixels is the input to the NT computation of each string; (ii) A temporal segmentation of each note played on a given string computed by segmenting the set of spectrograms of the string-pixel signals; (iii) The pitch is computed for each temporal segment of a note. The main challenge for computing the pitch is the low sample rate of the camera, which is 240fps (whereas audio has 44.1kHz). According to the Nyquist-Shannon sample rate theorem [23, 29], without prior knowledge of the possible set of frequencies, half the sample rate determines the set of frequencies that can be detected (which covers only about 2/3 for the bass guitar notes and 1/4 for the other guitar notes we consider). To overcome the low FR of the visual data, we use prior knowledge of the possible notes played on a given string. In particular, we use the expected harmonics and aliased harmonics of each note.

The main contributions of this paper are (i) a novel physics-based solution to the new challenge of visual-based NT using an off-the-shelf camera; (ii) using prior knowledge of string notes to resolve ambiguities in computing a note's pitch; (iii) locating a string using its vibrations. Finally, we present extensive experiments on 4 different guitars, and analyze the method's expected failures caused by the low FR.

2 Related Work

To the best of our knowledge, our method is the first to solve NT using string vibration analysis. Our work was inspired by the innovative study by David *et al.* [11], where sounds were recovered from a silent video. They showed that when sound hits an object, its surface vibrates, and it is possible to partially reconstruct the original sound by processing the signals obtained from those vibrations. Wu *et al.* [32] suggested that the temporal color changes in a certain image location could be extracted and amplified to make them visible to the naked eye. Rubinstein [27] demonstrates that such color change in a video of a vibrating guitar string, can reveal its vibration frequency. These studies used a high FR camera, or use rolling shutter properties to effectively increase the FR, which is not considered here. Owens *et al*. [24], used supervised learning to predict the sound of an object being hit/scratched in a silent video. It demonstrates that audio information could be retrieved by visual data alone.

Audio-based NT is a widely studied field. The Music Information Retrieval Evaluation eXchange (MIREX) [14] is an annual evaluation campaign for audio-based NT algorithms. Although monophonic audio-based NT is considered solved, polyphonic NT systems perform well below humans level [8]. Some audio applications attempted to retrieve the fingering information(e.g., [4]), however they are limited by the set of pre-defined chords, by the polyphony level [6, 13], an isolated single note [4], or a single instrument playing alone [13, 18]. Solving the fingering task is straightforward in our case, since each string is independently analyzed. See [5, 8, 16, 22, 30] for detailed surveys on NT and general AMT.

Visual NT methods were proposed, mostly for guitars. Most methods tracked the left-hand and guitar to determine the played chords. The guitar is tracked using its geometric structure [25, 26, 28], or by using markers on the guitar [10, 17, 19, 20, 21]; a mounted camera was used for this purpose in [9]. The left hand is tracked based on its shape [9, 31, 33], skin color [25, 26, 28, 31, 33], or using markers [19, 21]. In [17] identification of different chord voicing was attempted by training a set of rectified images of the fretboard. Since only the left hand is analyzed, these methods cannot determine when a note is actually being played or whether the players's hand merely rests on the fretboard and no sound is produced. In [33] it was attempted to determine if a finger is pressing a string of a violin, but results were poor. In [2, 12, 31] the motion of the players was used to detect onsets. Several methods that combine visual and audio data were suggested. In [25, 26] notes' onsets was retrieved from audio data and in [7] the visual data was used to solve ambiguities in audio NT.

3 Method

Our input is a video captured by a single camera mounted on the guitar (Fig. 1(d)). We assume that despite partial occlusion of string by the hands, there are parts of the strings that are visible throughout the entire video. For each string s, we first detect a set of string-pixels, α_s, which are roughly located on the projection of s (Sec. 3.3). A string-pixel signal, $q_s^i(t)$ for $i \in \alpha_s$, is the intensity of q_s^i as a function of the video frame and it is correlated with the string vibration frequency. The set $\{q_s^i(t)\}_{i\in\alpha_s}$ is the input for computing the temporal segmentation (Sec. 3.1) and the fundamental frequency, f_0 (Sec. 3.2) of the notes played on s. Multiple signals per string are used to achieve robustness due to noise of obfuscation that could occur during the video. The notes' pitch and temporal information from all the strings can then be rearranged to form a coherent musical sheet (tablature or MIDI-like).

3.1 Temporal Information

We propose an algorithm to extract the temporal information of the played music on a given string (each note's onset and offset). For each string s, we compute a set of temporal intervals $\{\tau^r(s)\}$, each corresponding to a single note played on s.

A note played on a string causes the string to move (fast) at the note's frequency. Hence, we segment the spectrogram representation of each of the input signals, $q_s^i(t)$. The final temporal segmentation of a given string is chosen to be the set of frames that appear in at least 65% of the segments computed for the set $\{\tau_i^r(s)\}$ (Fig. 2c).

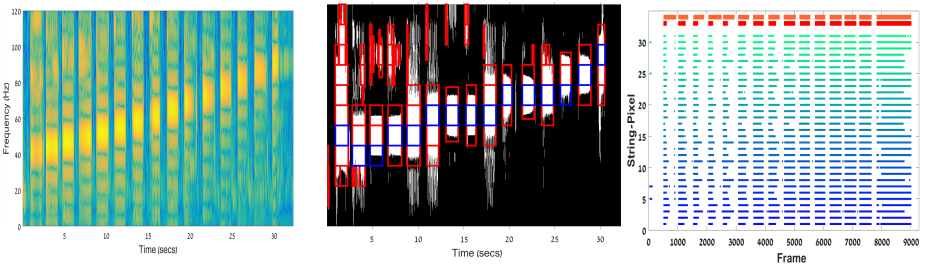
The spectogram is a time-frequency representation, a $T \times f$ map. It consists of a short-term Fourier transform (STFT), which is a FFT applied to a set of temporal segments of the signal, each of size N. Each of its columns is the segment's power spectral function, and the value at an entry $\gamma(t,f)$ is the amplitude of the frequency f computed for an interval of size N starting at t (Fig. 2a). The temporal resolution and frequency resolution of a spectrogram are complementary: large N is required for the frequency resolution while small N improves the time resolution. Here we favor the temporal information (we use $N = 20$ and large overlap between intervals), and largely ignore the pitch information, which is computed as a successive step when the temporal information is available (Sec. 3.2).

Segmentation: The spectrogram is expected to have high energy regions in the corresponding temporal interval when a note is played, roughly centered around the note's visible frequencies (defined in Sec 3.2.) Other regions of the spectrogram are expected to have low energy, except for noisy regions, typically at low frequencies. In order to segment the computed spectrogram, we use the observation that $q_s^i(t)$ is a signal obtained from a single string; hence only a single note is played at a given time. Moreover, N is sufficiently small to capture only a single note played on s.

We use a naïve segmentation method – a bounding box is set around each region of the spectrogram with energy higher than a given threshold. (Clearly, a more sophisticated onset detection method can be considered.) The bounding box coordinates in the temporal axes correspond to a temporal segment, denoted by $\tau_i^r(s) = (t_i^r(s), t_i^r(s) + \delta t_i^r(s))$. Overlapping temporal segments are discarded by choosing the more dominant regions, as at each given time only one note can be played. Temporally close segments are merged (Fig. 2b). We use a threshold that is data dependent (the top 20% of the energy is assumed to be played notes). In practice, we found that a false positive (FP) note can be detected on an unplayed string which is not a typical situation when playing on a guitar (in particular when chords are played). The FP detected are due to the guitar vibration when playing on another string. We avoid such false detections by testing whether the string move at a high velocity; When a string is played, its image is blurred and so no edge is likely to be detected along the string image. On the other hand, when a string is not played, edges are likely to be detected along the string image. Details on this pre-processing step, and the manner in which they are used for the segmentation, are given in the supplementary material (SM) [4].

3.2 Computing the Note's Fundamental Frequency

The name of a note is its fundamental frequency f_0 (or: *fundamental*). It is the note's lowest frequency component (a note may consists of several frequencies). Detecting f_0 directly from the computed spectrogram is not applicable in our case since the frequency resolution is too low. To increase it, FFT is applied to each temporal note segment, $\tau^r(s)$, computed in Sec. 3.1. Typically $\tau^r(s) > N$, since N was chosen to be small to improve temporal resolution



(a) (b) (c)

Figure 2: (a) The spectrogram computed for a chromatic scale (E on acoustic guitar). (b) The initial bounding boxes (after discrading temproally small ones that are considered noise) placed around the thresholded spectrogram (red), and the final interval selected, after merging temporally close intervals (blue). (c) The temporal segmentation results of each string-pixel for a given string, the voting result (red) and the GT temoral segmentation (orange).

of the spectrogram. The result is a power spectra function of each note, $S(f)$, from which we want to compute f_0. (For notation simplicity, we ignore the indexes of the string-pixel and the temporal segment in the notation.) The final f_0 of the note played at a given temporal segment is obtained by a majority voting of f_0 computed from each string-pixel.

The main challenge we address is the video's low sampling rate, $f_s = 240fps$. According to the Nyquist-Shannon sample rate theorem, a perfect reconstruction is guaranteed for frequencies within a bandwidth limit $B < f_s/2$, which we refer to as the visible range of the power spectrum. Hence, the video data we use is limited to frequencies below 120Hz. Observe that high frequencies up to 22.05kHz can be obtained from audio signals, sampled at 44.1kHz.

Aliasing & Harmonics: A peak of $S(f)$ is expected at f_0, if it is in the visible range ($f_0 < f_s/2$). Otherwise, it is theoretically guaranteed that one of its aliased frequencies, denoted by $f_A(f_0)$, is visible. The aliased frequency of f is defined by $f_A(f) = |f - kf_s| < f_s/2$ for a $k \in N^+$. The visible frequency of f (given f_s) is defined by:

$$f_v(f) = \begin{cases} f & f \leq f_s/2 \\ f_A(f) & f > f_s/2 \end{cases} .$$

Hence, if $S(f)$ has a single peak at f', we could infer that $f' = f_0$ or that $f' = f_A(f_0)$. Additional peaks are expected at the harmonics (or visible aliased frequencies) of f_0. Formally, let f_0 be a fundamental. Then the harmonic series of f_0 is given by $\{if_0\}_{i \in N}$ (f_0 is the first harmonic of itself). A peak of $S(f)$ is expected at each $f_v(f)$, where $f = if_0$. In practice, the harmonic's energy decreases as i increases: hence we consider only the fundamental and two additional harmonics of f_0. We expect $S(f)$ to have peaks at: $F_V(f_0) = \{f_v(f_0), f_v(2f_0), f_v(3f_0)\}$.

Thus, the structure of the music harmonics and our limited bandwidth of visible frequencies can cause additional ambiguities, since one also needs to discriminate between f_0 and its harmonics. That is, a peak of $S(f)$ at a frequency f' may be obtained for $f' = f_0$, $f' = f_A(f_0)$ or $f' \in \{f_v(2f_0), f_v(3f_0)\}$. For example, a peak in 110Hz can occur because a note's fundamental is 110Hz, because it is an aliased frequency of 130Hz (as $240 - 130 = 110$), or because it is a harmonic of a note with a fundamental of 55Hz (as $2 * 55 = 110$). Some peaks may shift due to $S(f)$ resolution, and may fall within the noise range ($\leq$ 20Hz), thus will not be visible.

Using prior knowledge: Our goal is to compute f_0 despite the expected abovementioned ambiguities. The most significant information we have is that $S(f)$ consists only of a single note. Moreover, the set of possible fundamentals that can be played on a given string s, $F_{string}(s)$, is also known (assuming a tuned instrument). As a result, if $f_0 \in F_{string}(s)$ is played, we expect the peaks of $S(f)$ to be the set $F_V(f_0)$. We define a score for each possible $f \in F_{strings}(s)$ and for the computed set of peaks, and choose the note with the highest score as defined below.

In audio analysis, no aliasing exists since the bandwidth is sufficiently large and an anti-aliasing filter is used. However, harmonics frequencies are visible and it is hard to discriminate whether a single note is played or both the note and its harmonics. We do not have to consider such ambiguity, since only a single note is played at a given temporal segment.

Score definition: Let $F_{peaks}(S(f))$ be the set of the highest six peaks of $S(f)$. The score of $f_0 \in F_{string}(s)$ is computed as the weighted sum of the distances of each $f \in F_V(f_0)$ from its nearest peak in $F_{peaks}(S(f))$. We normalize the distances between f and $f^p \in F_{peaks}(S(f))$, since the results of the discrete FFT are equally spaced while the distances between semitones are larger for high frequencies and smaller for lower ones. Formally, let $f < f'$ be two successive semitones, and let $f < f^p \in F_{peaks}(S(f))$ be the peak frequency. The distance between f and f^p is given by $d(f,f^p) = (f-f^p)/(f-f')$. Similarly, we define the distance $d(f,f^p)$ where $f > f^p$, using $f' < f$. The frequency of the nearest peak to f is given by

$$\hat{f} = argmin_{f^p \in F_{peaks}(S(f))} d(f,f^p).$$

The contribution of the i^{th} harmonic of f_0, $f = if_0$, to the score of f_0 is given by

$$e_i = e^{-d(f,\hat{f})} S(\hat{f}).$$

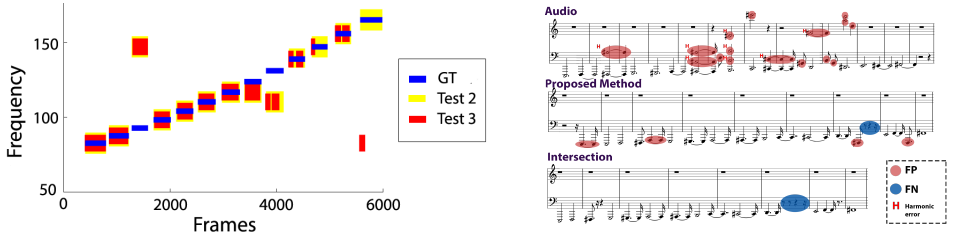
Then the score is defined by the weighted sum:

$$score(f_0) = \sum_{i=1}^{3} w_i e_i,$$

where w_i is a weight of the i^{th} harmonic. In our implementation the weights are set to be $w_1 = 0.6, w_2 = 0.25, w_3 = 0.15$, if all expected frequencies are above the noise range. Otherwise the weights are set to 0 for an invisible frequency, and the rest are set accordingly.

High energy frequencies at $F_V(f_0)$ are used as evidence to support that f_0 was played. However, special attention should be paid to two notes that are likely to be confused. These include a pair of notes such that one of them is the harmonic of the other, that is, $f_0, 2f_0 \in F_{string}(s)$. In this case, $score(2f_0)$ may be larger than $score(f_0)$ but f_0 is the correct fundamental frequency. To avoid such errors, we reexamine which note in the pair is the fundamental frequency. We expect $e_1(f_0)$ to be low if $2f_0$ is the correct one. We test this according to the rank of $e_1(f_0)$ in the set of $e_1(f), \forall f \in F_{string}(s)$. In our implementation we choose $2f_0$ if the rank is below 70%.

Expected Failures of Pitch Detection: The FR and the notes' properties can be used to predict failures of our method. We classify notes according to the expected failures (Fig. 4a), and use it to analyze the errors of our method (in SM [2]). The visible frequency components, $F_v(f_0)$, of a *noise-incident note*, v, is in the noise range. If $f_v(f_0)$ is in the noise range it has the greater affect on the expected error, since it has often higher energy than the other visible frequencies. Two notes are *indistinguishable* if their observed corresponding harmonic series



(a) (b)

Figure 3: (a) MIDI-like presentation: E string chromatic scale on acoustic guitar. (b) A music score sheet of Test 6: audio-based (first row), our method (second row), and their intersection (last row).

are identical. That is, $f_0 \neq f_0'$ but $F_V(f_0) = F_V(f_0')$. B♭2 (116.5Hz) and B2 (123.5Hz) are an example of indistinguishable notes ($F_V(116.5) = F_V(123.5) = \{116.5, 7, 109.5\}$). Also, we consider *harmonic pair* when one note is the 2nd harmonic of another note on the same string. Finally, we expect more errors when string's fundamentals are dense (Fig. 4b).

3.3 String-pixel Detection

The input for both the temporal note segmentation and fundamental frequency computation are the sets of 31 signals for each sting, $\{q_s^i(t)\}$. Each signal consists of the intensity change of a single string-pixel over time. Here we present our algorithm for locating the string-pixels using the frequency information from image pixels.

The signal of all pixels on or near the projection of the string are expected to change in the same frequency as the played note. This frequency can be computed using FFT as described in previous sections. Hence, we can consider all image pixels that change in the same frequency as the note, to detect the string and choose a subset of these pixels as the string-pixels. For efficiency, we consider only image pixels that are located near an image edge (using dilated edge image), since a string is expected to give rise to an edge on a single image.

To do so, an initialization video is captured where a predefined chord, with known frequencies for each string, is played. Then the signals of all pixels that are located by the image edges are computed, and 31 string-pixels are randomly chosen from the set of pixels that yielded high energy for the expected frequency of the string (see Fig. 1). Note that these pixels should be detected only once, since a mounted camera is used.

4 Experiments

The algorithm was implemented in Matlab and run on a standard PC.

Since no existing visual data is available, we collected a new dataset available on project webpage [3]. which consists of videos captured by a camera (GoPro HERO3+ with 240 FPS in WVGA resolution) mounted to the guitar. Four tuned guitars were used; classical, electric, electric bass and acoustic. These guitars' strings are made from different materials that constitute a representative sample of all common guitar strings. We consider the four lower strings,(E, A, D, G) and the 13 lower notes of each string for standard guitar and 15 for the Bass, since higher strings are thinner hence higher camera resolution is required; higher notes are harder to detect, probably because the strings are shortened hence the strings' amplitude

Str.	Ons.		Offs.	Pitch	
	R	P	TP^*	GT τ	auto τ
E(1)	.97	.84	.79	.68	.68
A(2)	.94	.82	.74	.75	.73
D(3)	.80	.7	.68	.6	.62
G(4)	.83	.61	.59	.62	.62
Tot.	**.88**	**.74**	**.7**	**.66**	**.67**

(a)

Str.	B.	Ac.	Cl.	El.	All
E(1)	93	64	66	64	72
A(2)	83	71	62	70	71
D(3)	71	67	64	67	67
G(4)	68	55	60	69	63
Tot.	79	64	63	68	**68**

(b)

Table 1: (a) Tests 1-3 results per string (str.). TP^* is TP offsets out of TP onsets. (b) Test 4 results (in %), per string and instrument (bass (B.), acoustic (Ac.), classical (Cl.) electric (El.).

is smaller. We played the chromatic scale (consecutive semitones on all the considered frets) four times on each guitar, $864 = 15 \times 4 \times 4 \times 1 + 13 \times 4 \times 4 \times 3$ notes in all. The chromatic scale allows a systematic evaluation of all possible notes played on each string, independent on the notes of a specific set of melodies. Moreover, although playing a chromatic scale is easier than playing a melody, NT of a chromatic scale may be harder since the played notes are adjacent both temporally and in pitch. We made no assumption of the type of music played; hence, this data represents well all musical pieces played in the considered range. This data was used for tests 1-4. An example of polyphonic chord playing and melodies are presented as well.

The string-pixels were chosen using the temporal-spectral algorithm, the notes were played on the 6th, 7th, or 8th fret. We found that the method is more robust when these notes are chosen rather than open string notes or manually chosen ones (see SM [3]).

We evaluate the performance of each component of our algorithm separately, as well as the performance of the entire system. Due to space limitations, we present the average results for all guitars and strings. In the SM [3], we analyze the results per strings and instruments using the expected failures described above. Additional results and analyses per instrument, string fret, and additional thresholds of 12, 24, and 36 frames are also found in SM [3]. A visualization of the different tests results is shown in Fig. 3a.

Our method is the first to compute NT from string vibration captured by video; hence, it cannot be compared to existing methods. However, we present various demonstrations and tests for comparing to existing audio-based methods.

Test 1: Temporal Note Segmentation. Here we ignore the note's pitch. For evaluation, the ground truth (GT) intervals were estimated manually using the spectrogram of a single-string pixel. It is expected to be inaccurate. An onset detection is considered correct, if it is within a tolerance time window defined by a threshold around the nearest GT onset. Given a correctly detected onset, the offset is considered correct if it is within a tolerance time window (we use 80 frames) around the GT. The average recall of 0.88 and precision of 0.74 (F-measure of 0.8). In addition, 70% of the notes for which onsets were detected were also successfully matched with offsets.

Failures to detect a note are expected for noise-incident notes. A split of a note's temporal interval results in both false onset detection and offset error. Noisy data may cause false detection of a note. Finally, fading out of a note may cause offset errors.

The 80 frame threshold we use (which is 0.03 of 2205) corresponds to $1/3$ of a second. Although slightly slow for professional guitarists, it is a reasonable pace for educational

purposes and chord playing. For comparison, 50ms the threshold used in MIREX, which correspond to 12 frames in our camera FR and to 2205 samples in audio recorded at 44.1 kHz. Our results cannot be compared to MIREX (2016) since the authors considered various musical instruments (including unpitched ones) and different musical pieces and GT annotation. However, note that the F-measure reported by MIREX (2016) is 0.87, which is only 0.07 better than our results.

In addition we test a variation of our temporal note segmentation method, that deals with unplayed strings as well. The frames are processed in temporal chunks. Due to fast motion of the vibrations, a played string is expected to be blurred (no edges) for the entire chuck. The evaluation threshold is proportional to the number of frames in the chunk. The data consists of 8 chromatic scale videos played on the Bass, 2 for each string. No assumption is made regarding which string is played. The TP remains high but the FP increased, mainly for unplayed strings.

Test 2: Pitch Detection using GT intervals. We tested the pitch detection algorithm independent of the temporal segmentation method, by using the GT temporal intervals as an input. Overall, 66% of the notes' pitches were detected correctly: 86% for the bass guitar, and an average of 58% for the others. The better performance on the Bass is expected due to the strings' thickness and length and the fewer noise-incident notes.

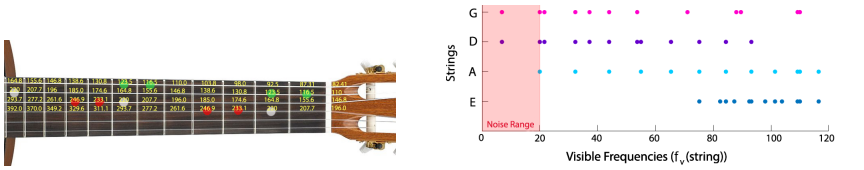
Test 3: Pitch Detection using computed intervals. For a complete evaluation, we present the pitch detection results using the temporal note segments computed by our method. We compare the pitch of each GT interval with the pitch of the computed interval with the largest overlap. Notes for which no overlapped temporal interval was detected were ignored. Overall, 94% of the GT temporal intervals were matched with the detected ones (96% for the bass and 93% on average for the other guitars). The pitch of the matched intervals was correctly detected in 67% of the notes (88% for the bass and 58% on average for the other guitars).

Test 4: Frame-by-Frame Evaluation. The entire method is evaluated using frame-by-frame comparison of our results to the GT, as used in MIREX 2016 [2] for f_0-estimation. A frame is considered correct if the detected pitch is identical to the GT or if no note was played and no note was detected (Table 1b). Overall, 68% of the frames were correctly detected. In this test, as in the previous tests, the results on the Bass were better than for the other guitars.

Test 5: Melodies and Polyphony Data. For completeness, we also tested our method on melodies played on the Bass (Billy Jean, Feel Good, Sunshine of Your Love, Come as You Are). We evaluated our system only on the played strings, and achieved results of 82% for frame-by-frame, onset f-measure of 0.79 and 83% for pitch detection using the calculated temporal segments.

In addition, we demonstrate that our method can deal with polyphonic data played on acoustic guitar (chords). This reveals two important properties of our method. First, polyphony does not affect the signals extracted from each string separately. Second, since every string is analyzed separately, an error on one or more strings may not prevent the correct identification of a chord. It remains for future work to develop a chord-oriented system that could analyze the played notes as chords. (See SM [3] for details).

Test 6: Audio-based method. We demonstrate the performance of an audio-system (AnthemScore [1]) and compare it to our method. We captured a video, using also the camera's microphone, of a chromatic scale played on the E string on the Bass (15 notes). Although monophonic data is considered to be solved by audio-based systems, the audio-based onset detection has 15 TP and 16 FP (precision of 0.48 and recall of 1). This maybe due to the



(a) (b)

Figure 4: (a) The standard guitars' fundamentals (better viewed on screen). The pairs of red and green dots mark a pair of indistinguishable notes. White dots mark notes whose f_0 is within the noise range. Frequencies 233.1 and 246.9 are marked as indistinguishable although their f_0 and $2f_0$ are within the noise range (6.9Hz), hence undetectable in practice. (b) The density of the $f_v(f_0)$ for each $f_0 \in F_{String}$ for all guitars except the bass guitar.

quality of the camera's microphone. Our algorithm, with no assumption made on the string on which the music is played, had only 1 FN and 4 FP (0.8 precision, 0.93 recall). Both systems found the correct pitch for every true detected temporal segment. A visualization of the results are presented in Fig. 3b. Note that different errors were obtained by the different methods, and the the intersection of the detected notes removes all false positive detections, resulting in a single FN error. This supports our claim that our system can have a significant contribute as a complementary system to an audio-based one.

5 Conclusion and Future Work

We present a method to use visual detection of string vibrations for extracting musical information (NT) from a silent video and demonstrate it on guitars. Our method overcomes challenges of audio-based polyphonic NT played on guitars by effectively reducing the complexity of polyphonic music analysis to the analysis of multiple signals capturing monophonic music. We resolve issues that arise from the camera's low FR by considering the expected notes' visible frequencies. Our method is mostly not instrument-specific and should be easily generalized to other string instruments.

Future directions may include changes in camera settings. Additional information can be obtained by considering a setting of two cameras with different frame rates, which will decrease ambiguities cause by aliasing. Conversely, an unmounted camera may introduce challenges and require tracking, but is more user friendly. Here, methods to amplify the subtle string vibration inside the player or guitar larger motion, as suggested in [15, 34], can be used. Additionally, using a higher FR camera should improve results.

Our method is the first to use visual detection of string vibrations for NT, which opens the door to exciting opportunities. In particular, a hybrid system combining visual and audio data can be of high value, as visual processing reveals information that is harder to extract from audio signals. The visual system can be a complementary tool for musicians in the task on NT, as it overcomes some of the audio limitations.

Moreover, since our data consists of 1D signals, which is similar to audio 1D signals, integrating components from existing audio methods to our system should be considered. Post-processing steps are expected to improve results, especially ones that consider the musical context. Finally, it remains to be seen whether replacing our physics-based approach with learning methods can improve the results.

Acknowledgments. This work was partially supported by the Israel Science Foundation, grant no. 930/12.

References

[1] AnthemScore: automatic music transcription software. https://www.lunaverus.com/.

[2] Music Information Retrieval Evaluation eXchange 2016: multiple fundamental frequency estimation and tracking. http://www.music-ir.org/mirex/wiki/2016:Multiple_Fundamental_Frequency_Estimation_%26_Tracking.

[3] Guitar Music Transcription from Silent Video: porject website, including data, code, and supplemental material. https://shirgoldstein.wixsite.com/shirg/nt-from-silent-video.

[4] J. Abeßer. Automatic string detection for bass guitar and electric guitar. In *International Symposium on Computer Music Modeling and Retrieval*, 2012.

[5] F. Argenti, P. Nesi, and G. Pantaleo. Automatic music transcription: from monophonic to polyphonic. *Musical Robots and Interactive Multimodal Systems*, 74:27, 2011.

[6] A. M. Barbancho, A. Klapuri, L. J. Tardón, and I. Barbancho. Automatic transcription of guitar chords and fingering from audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(3):915–921, 2012.

[7] A. Bazzica, J. C. van Gemert, C. S. S. Liem, and A. Hanjalic. Vision-based detection of acoustic timed events: a case study on clarinet note onsets. In *Proceedings of the First International Conference on Deep Learning and Music*, pages 31–36, 2017.

[8] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407–434, 2013.

[9] A. M. Burns and M. M. Wanderley. Visual methods for the retrieval of guitarist fingering. In *Conference on New interfaces for musical expression (NIME)*, 2006.

[10] M. Cicconet. *The Guitar as a Human-Computer Interface*. PhD thesis, D. Sc. Thesis. National Institute of Pure and Applied Mathematics. Rio de Janeiro, 2010.

[11] A. Davis, M. Rubinstein, N. Wadhwa, G. J. Mysore, F. Durand, and W. T. Freeman. The visual microphone: Passive recovery of sound from video. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 33(4):79:1–79:10, 2014.

[12] K. Dinesh, B. Li, X. Liu, Z. Duan, and G. Sharma. Visually informed multi-pitch analysis of string ensembles. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3021–3025, 2017.

[13] C. Dittmar, A. Männchen, and J. Abeßer. Real-time guitar string detection for music education software. In *International Workshop on Image Analysis for Multimedia Interactive Services*, 2013.

[14] J. Downie, A. Ehmann, M. Bay, and M. Jones. The music information retrieval evaluation exchange: Some observations and insights. *Advances in music information retrieval*, pages 93–115, 2010.

[15] M. Elgharib, M. Hefeeda, F. Durand, and W. T. Freeman. Video magnification in presence of large motions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4119–4127, 2015.

[16] B. S. Gowrishankar and N. U. Bhajantri. An exhaustive review of automatic music transcription techniques: Survey of music transcription techniques. In *Signal Processing, Communication, Power and Embedded System*, 2016.

[17] A. Hrybyk and Y. Kim. Combined audio and video analysis for guitar chord identification. In *International Society on Music Information Retrieval (ISMIR)*, 2010.

[18] C. Kehling, J. Abeßer, C. Dittmar, and G. Schuller. Automatic tablature transcription of electric guitar recordings by estimation of score-and instrument-related parameters. In *Conference on Digital Audio Effects (DAFx)*, 2014.

[19] C. Kerdvibulvech and H. Saito. Real-time guitar chord estimation by stereo cameras for supporting guitarists. In *International Workshop on Advanced Image Technology (IWAIT)*, 2007.

[20] C. Kerdvibulvech and H. Saito. Vision-based detection of guitar players' fingertips without markers. In *Computer Graphics Imaging and Visualisation (CGIV)*, 2007.

[21] C. Kerdvibulvech and H. Saito. Vision-based guitarist fingering tracking using a bayesian classifier and particle filters. In *Pacific-Rim Symposium on Image and Video Technology (PSIVT)*, 2007.

[22] A. Klapuri. Automatic music transcription as we know it today. *Journal of New Music Research*, 33(3):269–282, 2004.

[23] H. Nyquist. Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, 47(2):617–644, 1928.

[24] A. Owens, P. Isola, J. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman. Visually indicated sounds. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[25] M. Paleari, B. Huet, A. Schutz D., and Slock. A multimodal approach to music transcription. In *International Conference on Image Processing (ICI)*, 2008.

[26] G. Quested, R. Boyle, and K. Ng. Polyphonic note tracking using multimodal retrieval of musical events. In *International Computer Music Conference (ICMC)*, 2008.

[27] M. Rubinstein. *Analysis and Visualization of Temporal Variations in Video*. PhD thesis, Massachusetts Institute of Technology, Feb 2014.

[28] J. Scarr and R. Green. Retrieval of guitarist fingering information using computer vision. In *Image and Vision Computing New Zealand (IVCNZ)*, 2010.

[29] C. E. Shannon. Communication in the presence of noise. *Proceedings of the Institute of Radio Engineers (IRE)*, 37(1):10–21, Jan 1949.

[30] T. F. Tavares, J. G. A. Barbedo, R. Attux, and A. Lopes. Survey on automatic transcription of music. *Journal of the Brazilian Computer Society*, 19(4):589–604, 2013.

[31] Z. WANG and J. OHYA. Tracking the guitarist's fingers as well as recognizing pressed chords from a video sequence. *Electronic Imaging*, 2016(15):1–6, 2016.

[32] H.-Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand, and W. T. Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM Trans. Graph. (Proceedings SIGGRAPH 2012)*, 31(4), 2012.

[33] B. Zhang, J. Zhu, Y. Wang, and W. K. Leow. Visual analysis of fingering for pedagogical violin transcription. In *Proceedings of the 15th ACM international conference on Multimedia (ACMMM)*, 2007.

[34] Y. Zhang, S. L. Pintea, and J. C. van Gemert. Video acceleration magnification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 502–510, 2017.