The Interdisciplinary Center, Herzlia
Efi Arazi School of Computer Science
M.Sc. program

Vision-Based Musical Notes Recognition of String Instruments

by
Shir Goldstein
Thesis Supervisor: Prof. Yael Moses

Final project, submitted in partial fulfillment of the requirements for the M.Sc. degree, School of Computer Science The Interdisciplinary Center, Herzliya

September, 2017

This work was carried out under the supervision of Prof. Yael Moses as part of the M.Sc. program of Efi Arazi School of Computer Science, The Interdisciplinary Center, Herzliya.

Abstract

Music Information Retrieval (MIR) is an important research field in music, combining computer science, signal processing, physics, psychology and more. One of the most important subtasks of MIR is Automatic Music Transcription (AMT), which involves notating an unnotated musical piece. This task is of great value in the music discipline, but it is also complex and, for polyphonic musical pieces, is still below human performance. We present a novel approach for a specific AMT task, Note Tracking (NT) for guitars (AMT limited to only the pitch of the note and its occurrence in time), using only a silent video of a guitar, captured by a camera mounted on it. We use the vibration of the strings as a visual signal and analyze it using various signal processing and computer vision methods. We process each sting separately which practically allows reducing the complexity of a polyphonic NT to multiple monophonic NT. We also use the physical characteristics of the guitar, like the possible notes that can be played on a specific string, in order to limit our search space which is inapplicable in audio methods. We analyze the expected errors of our method, given the instrument, the string, and the frame rate of the camera. The performance of our method were tested on four different guitars, and it is shown that our algorithm can play an important role in solving the NT problem. Additional information is required to obtain perfect results.

Acknowledgements

First I wish to express my deep gratitude to my supervisor Prof. Yael Moses. I have been fortunate to work with a professional and a knowledgeable person, that provided me with support and guidance. It has been an inspiriting journey.
I would like to thank my parents and my brother and sister that encouraged and supported me.
Finally I'd like to thank Moty, my husband, for his love, support and understanding.

Contents

List of Figures

Abbreviations

AMT Automatic Music Transcription

DFT Discrete Fourier Transform

FFT Fast Fourier Transform

FPS Frames Per Second

GT Ground Truth

Hz Hertz

MIDI Musical Instrument Digital Interface

MIR Music Information Retrieval

MIREX Music Information Retrieval Evaluation eXchange

NT Note Tracking

STFT Short Term Fourier Transform

Chapter 1

Introduction

Sheet music is a form of musical notation that indicates instruments players how to play a certain musical piece. The most important elements in a musical sheet are the notes' pitches, onsets (the notes' beginning) and offsets (the notes' end), but a complete musical sheet may also consists of other elements, such as key, tempo and other rhythmic information, fingering and dynamics (e.g., loudness and intonation).

The task of music transcription is a key task in Music Information Retrieval (MIR). This task entails the translation of an audio recoding or performance of a musical piece to some kind of written form. Only limited number of the musical pieces are notated by the composer or performer, while others need to be manually extracted based on listening alone. Manually transcribing music is a non trivial task even for skilled musicians and can be time consuming and inaccurate, thus an automatic mechanism is required.

Automatic Music Transcription (AMT) is generally considered as the task of automatically constructing a musical sheet from an auditory musical data. Although audio data is a natural source of information for AMT, our goal is to use computer vision to generate a coherent musical sheet of music played by a string instrument, given only a visual data of the instrument (see Figure 1.2). As most audio-based AMT applications restrict themselves to the detection of the notes' pitch, onset, and duration or offset, which is usually referred to as Note Tracking (NT), we will restrict our system to those components as well. Our focus is on various guitars, but our method is mostly not instrument-specific and can be easily generalised to work on other string instruments.

Note tracking for monophonic music (only one voice playing at a given time) is considered to be solved by auditory applications. The main challenge for auditory application is to separate different notes in polyphonic music (two or more notes are played simultaneously), in particular, when there are several instruments playing together, some of which are of the same type (e.g., a string quartet that includes two violins). Another challenge is to determine the fingering configuration which is an important aspect of the transcription. However, it is often ambiguous in audio data since the same note (with the same pitch) can be produce by several strings (e.g., the note A2 in a classic guitar can be produced by pressing the 5th segment of the lowest string (E) and the second string (A) open).

A small number of audio applications addressed the issue of identifying what string and fret is being played, but those are limited; either by a pre-set of known chords to identify, or by analyzing an individual string played in isolation. Most of

those methods work merely on a guitar and rely on its specifications, and cannot be simply generalised to work on other string instruments.

To overcome the audio limitations, vision-based methods were suggested. Those methods use tracking of the left hand fingers to determine the chords being played. Most of the existing vision-based methods or hybrid methods combining both audio and video data, focuses on the guitar. The underline technique for those methods is to visually identify the fretboard, strings and the player's left hand fingers positioning, in order to estimate what finger is pressing which string and fret, and using that to extract the chord (e.g., [1, 2, 3]). These methods carry a crucial, inherent limitation: They are unable to detect onsets and offsets, as they merely detect the left hand positioning needed to execute the chord. Neither a strumming nor a plucking of the strings is detected, thus making it impossible to determine whether the strings vibrate or are still. In other words, those methods cannot determine when a note or a chord is being actually played or rather if the player's hand merely rests on the fretboard. Furthermore, the left hand only determines the pitch of the notes, whilst the right hand is responsible to theirs order, duration and if they are actually being played at all, thus making it impossible to extract a specific melody or retrieve information about each of the notes' temporal information. Mostly visual methods offer only modest success rates, are restricted to some pre-set or are limited to identify playing on some parts of the guitar.

Recovering and reconstructing of sound from a vibration of an object in a video was shown in the work of Davis *et al.* [4]. We suggest a novel vision-based approach to further analyze the obtained signal (or rather - signals) to perform NT, and extract additional musical information from the video by using prior knowledge. Our method obtains multiple visual signals from the vibration of the strings, rather than tracking of the player's motions and actions. These signals allow us to use common signal processing methods, as used in the existing advanced audio-based NT systems. This is possible since the audio-captured sound wave is produced by the vibrating string, thus the audio signal has similar frequency components as the visual signal of the vibration. Furthermore, obtaining the change in intensities caused by the string displacement in a certain spatial location upon the string, correlates with the string vibration. As a result, the intensity change signal and the audio signal results in a similar response in the frequency domain. This was shown by Wu et al. [5] and Rubinstein [6]. They revealed signals that are imperceptible to the human eye by amplifying those color variation in a certain spatial location (pixel or a set of pixels). Thus for a stabilized video, like in our case, using this observation relinquishes a direct tracking of the string. We will further discuss the above-mentioned studies in the related work Chapter 3.

We demonstrate the correlation between an audio signal of a note produced by a vibrating string and a visually-obtained intensities change signal of this vibration by simultaneously capturing a note by a video recorder and an audio recorder and observing the respective frequency domain responses. Figure 1.1 shows those obtained signals when playing a single note, E2, with a fundamental frequency of approximately 82.41 Hz, and the correlating frequency response, using FFT. Those signals have roughly the same peaks in the frequency domain.

Our innovative approach allows to overcome some of the above-mentioned limitations of both visual and audio NT systems. However, it poses new challenges. The main challenge we address here is the relatively low frame rate of the camera

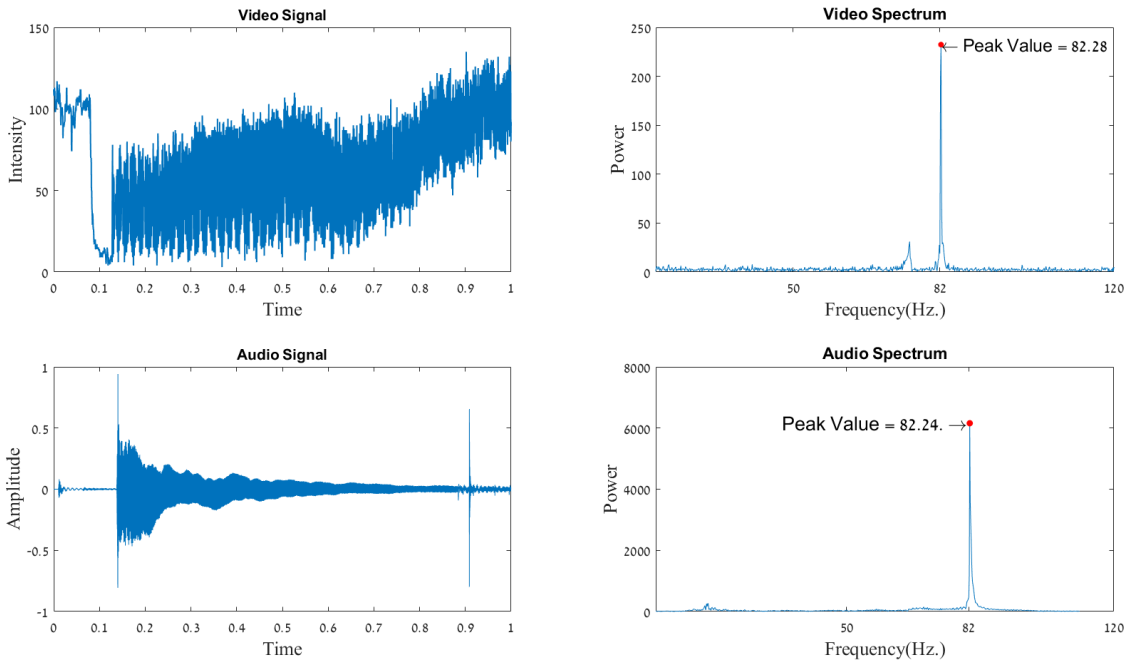


Figure 1.1: The signals captured from both audio and video data of a guitar playing the note E2 (82.41 Hz) by plucking the open lowest string are shown on the left column. The signals' representation in the frequency domain are shown on the right column. For the video signal, there is an obvious peak in 82.28 Hz, and for the audio one there is an obvious peak in 82.24 Hz.

(240 fps in our case) with respect to a typical audio sampling rate (44.1 kHz). Low frame rate causes the detection of the note's pitch to become substantially more challenging. In particular, the Nyquist-Shannon sample rate theorem [7, 8] guarantees perfect reconstruction (or reliable analysis) only for a signal with a bandlimit smaller than half the frame rate (see Section 2.2.1). For detection of higher frequencies, the aliasing phenomenon must be considered. Moreover, the sensitivity to noise is more substantial for a low frame rate.

An additional challenge for a fully automatic system, is to detect in all frames the key pixel we call a *string-pixel* (or pixels) on which the vibration is measured. For simplicity, we assume that the camera is mounted on the instrument, hence no stabilization of the video is required. This setup is a first step toward more general setups that include using an unmounted camera, using two cameras, etc. Although some obfuscation of the strings by the player's hand or body is allowed, we assume that some parts between the left (fretting) hand and the right (strumming) hand are visible throughout the entire video.

We explored two different approaches for strings detection, which is necessary for obtaining the string-pixels. The first approach is geometric, is based on the guitar physical structure and obtained from a single frame. The second, which is more effective, is a spatial-temporal approach that considers the strings vibration throughout multiple frame.

The rest of the thesis is organized as follows: In Chapter 2 we will provide a short theoretical background of both musical terms and mathematical ones. We briefly review the theory of string oscillation and explain about harmonics and the fundamental frequency of a note. We also review the time-frequency analysis for

MIR applications, which is a basic components to a vast majority of NT systems. In Chapter 3 we review the related work of both auditory methods and visual ones. In Chapter 4 we present our method. We explain the flow of the method and its expected limitations. We review the aliasing problem and the effect it has on our method. In Chapter 5 we present the testing of our algorithm, and also each of its components. Several measurements were used for evaluating the performance of our method. Finally, in Chapter 6 we conclude our work and discuss optional future work.

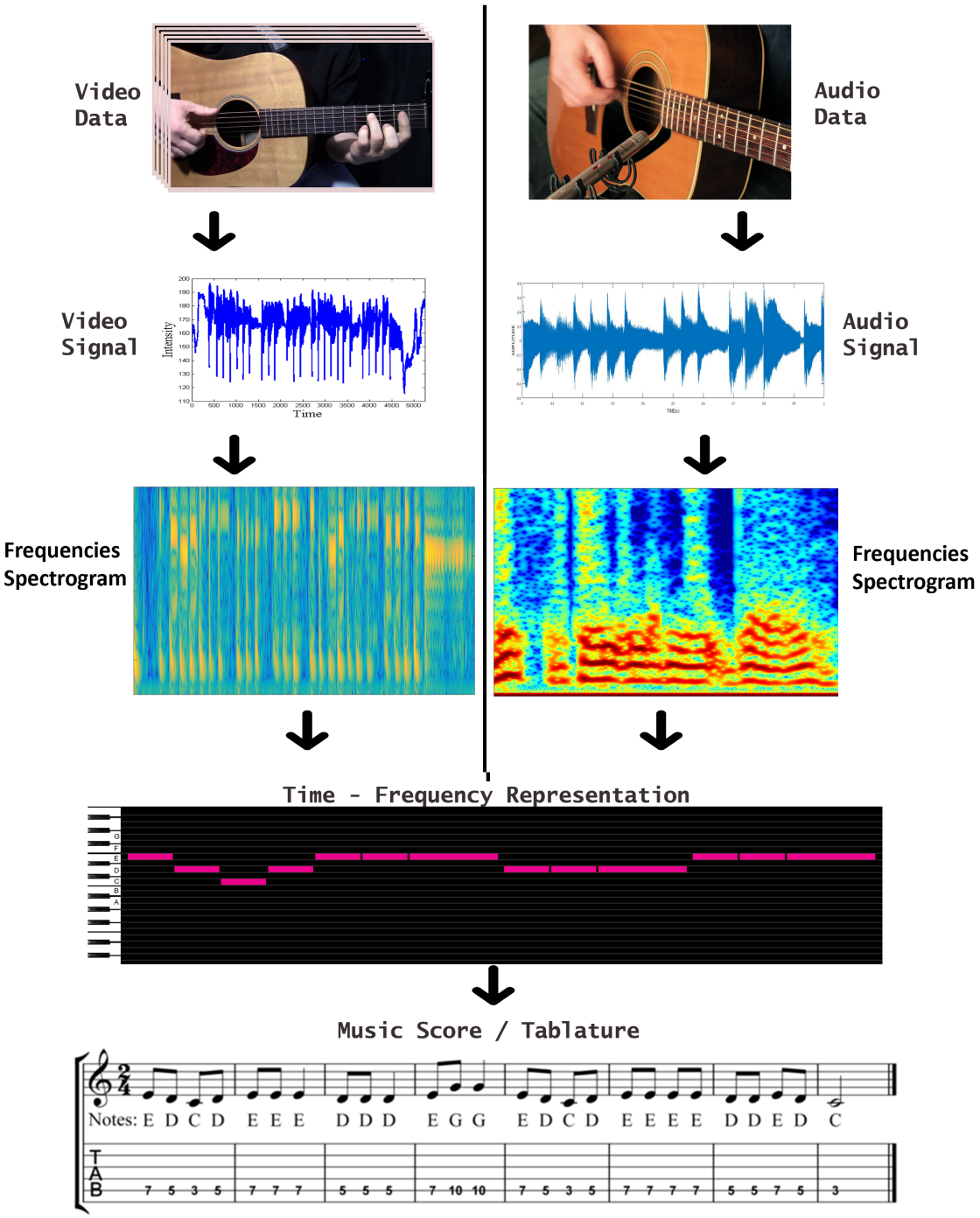


Figure 1.2: Illustration of the automatic music transcription process, for both audio and video data. First, the signals are obtained from either audio of visual data, then anaylzed using signal processing methods, then represented in a time-frequency representation, and finally arranged to a complete musical score or tablature.

Chapter 2

Theoretical Background

In this chapter we introduce the technical terms relevant to this work. Those include music related terminology as well as signal processing and frequency domain analysis terms.

2.1 A Musical Note

The musical note has many meanings, amongst a musical entity, a notation sign, a pitch and more. The basic components of a played musical note are threefold:

- Height - How high or low does a note sounds.
- Timbre - The sound of the note that differentiate two notes with the same height played on different instruments.
- Temporal information – Note's duration, and in a context of a musical piece - when is a note played.

We hereby define the meaning of a musical note to be used throughout this work; a *musical note* (or simply: note) is a musical entity, produced by a pitched instrument, that consists of a tone (also: pitch), and in the context of a musical piece carries additional temporal information (e.g., when a note is played). Timbre will not be addressed throughout this thesis. We will only address string instruments, and not other pitched instruments as brass instruments.

2.1.1 The Physicality of the Musical Note

When an ideal string that is fixed in both ends vibrates, it produces different waves, each with a particular frequency, amplitude and phase. The lowest frequency in which it vibrates is called the *fundamental frequency* (or simply: fundamental, and abbreviated as f_0). The entire set of frequencies in which the string vibrates are an integer multiple of the fundamental frequency and are called *harmonics*. Each harmonic is a member of the harmonic series defined by the fundamental. Formally, let f_0 be a fundamental frequency, then the harmonic series is defined as

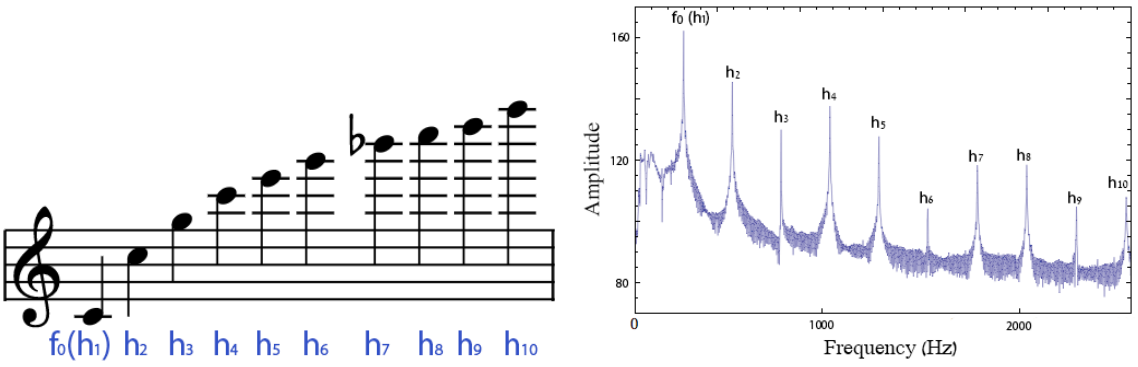
$$H_n = \{h_i \mid h_i = i \cdot f_0, i \in \mathbb{N}^+\}.$$

It follows that f_0 is the first harmonic and is equal to h_1, the second harmonic is h_2 and so on.

When a string vibrates in a certain frequency, it causes pressure fluctuations in the air around it in the same frequency. These fluctuations eventually vibrates our ear drum and produce a sound. But in practice, especially when a string is plucked or struck by a hammer (as in a guitar and a piano respectively) as opposed to bowed for example, the string vibrates in slightly different set of frequencies. This causes the instrument to produce a complex tone, that is constructed from pure (also: musical or simple) tones. Those tones are periodic and have a single sinusoidal waveform, i.e., contains a single frequency. Those tones, called *partials*, mainly include the harmonics. Any partials which are not harmonics are called *inharmonic partials*. Throughout this work we will address the instruments as producing only harmonic partials, as the partials it actually produces are very close to the harmonics and the inharmonic partials tend to decay rapidly.

The *pitch* of a note is a perceptual term that is used to describe how high or low a tone is. Pitch is closely related to the fundamental frequency, because generally the perceived pitch correlates with the fundamental – a high pitch sound corresponds to a tone with a high fundamental and a low pitch sound corresponds to a tone with a low fundamental. Throughout this thesis we will use fundamental frequency, fundamental, and pitch interchangeably, as is customary among musicians.

It follows that each musical note has a specific, distinct set of frequency components. This allows the human ear as well as computerized programs to identify and differentiate between notes with different fundamentals or pitch. For example, the middle C, C4, has a fundamental frequency of 261.63 Hz, and has harmonics of 523.26 Hz, 784.89 Hz and so on (see Figure 2.1). The amplitude of each partial determines the timbre of the note. For that reason different instruments that plays the same note (with the same tone) sound different, e.g., the different sound produced by playing C4 on a guitar and a flute. In this work we will assume that each of the guitars produces at most the first, second and third harmonics with some amplitude. This, since the higher a harmonic is the lower is its amplitude and in practice we found that amplitudes of harmonics higher than the third are as low as the noise.



(a) Middle C harmonics

(b) Middle C spectrum

Figure 2.1: Middle C harmonics and spectrum up to the 10th harmonic, produced by a plucked guitar string.

2.2 Fourier Transform and Analysis

Fourier Transform is a method to convert a signal, which is in the time domain, into the frequency domain, by decomposing it to its frequency constituent parts.

The Fourier transform can operate on continuous-time signals (which are often periodic, or constructs of periodic components) in order to create a continuous frequency spectrum. However, in practice, signals are discrete and finite, since they are reduced from real-world, continuous-time signals by sampling (see Section 2.2.1). For these signals, a discrete version of the Fourier transform, the *Discrete Fourier Transform (DFT)* is used. Many applications in digital signal processing use this transform, or rather its fast implementation *Fast Fourier Transform (FFT)*, and it is presumed to be the most important discrete transform.

Formally, DFT transforms a sequence of samples $x_1, x_2, ..., x_{n-1}$ to the sequence $X_1, X_2, ..., X_{n-1}$ which is defined by

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi kn/N}$$

2.2.1 Sampling and Aliasing

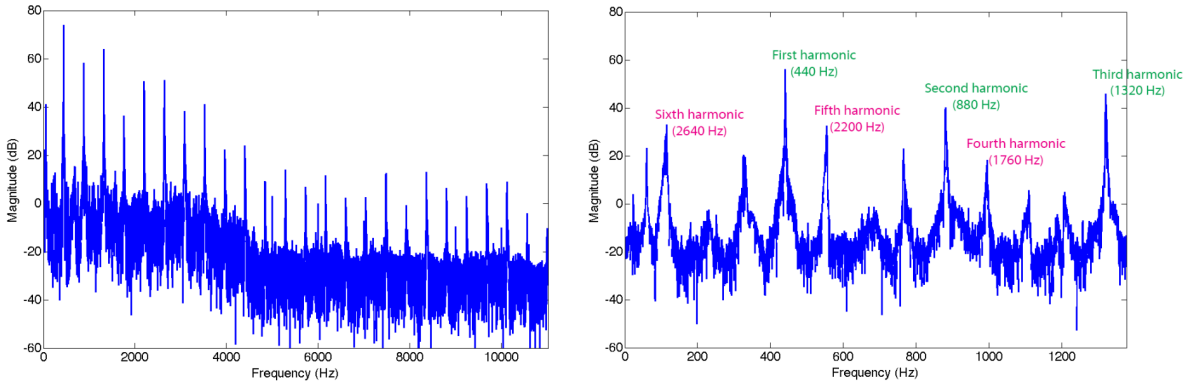
The process of reducing a continuous-time (analog) signal to a discrete-time signal is called *sampling*. The capturing of a sound wave by a microphone is an example of sampling, as well as capturing a scene by a video camera.

When a signal is sampled, it is inherently band-limited in frequency. That is, sampling a signal with a finite number of points, clearly cannot represent an infinite range of frequencies. Thus, every signal obtained by a sampling process is limited to a specific frequency range that is determined by the sampling rate, defined by the number of samples per second. Figure 2.4 in part illustrates the sampling process.

The Nyquist-Shannon sampling theorem [7, 8] establishes a sufficient condition for a sample rate that permits a discrete-time signal to entirely capture the information from a continuous-time signal. According to the theorem, the sufficient frame rate for a signal that has a maximum frequency of f_{MAX} should be at least $2f_{MAX}$ samples per second. In other words, the sampled signal must contain no sinusoidal component that is higher than half the sample rate. It follows that the frequency spectrum of a time-to-frequency transform will be limited to half the sample rate. This defines the typical audio sampling frequency to be 44.1 kHz which is slightly higher than double the maximal frequency of the human hearing range (20 kHz).

However, when a signal that does contain frequency components higher than $f_s/2$ is sampled with a sample rate of f_s, a phenomenon called *aliasing* occurs. Aliasing is an effect that causes different signals to become indistinguishable when sampled. Equivalently, such a signal will not correctly show its frequency components, as some can exceed the maximum frequency of the frequency spectrum. Any frequency component above $f_s/2$ is indistinguishable from a lower-frequency component, and is called an alias. In the case of an insufficient sample rate f_s, for a frequency above $fs/2$, f_{high}, there exists $m \in \mathbb{N}^+$ such that $f_a = |f_{high} - mf_s| < f_s/2$ that it aliases to. For example, when sampling with 240 Hz, two sinusoidal signals with frequencies of 110 Hz and 130 Hz are aliases of one another, as both will have a frequency component of 110 Hz. The 110 Hz sinusoidal will show a signal component in 110 Hz, since it is lower than half the sample rate. However, 130 Hz is above $f_s/2$,

thus aliases to 110 Hz, since $110 = |130 - (1)240|$. Although usually an anti-aliasing filter is used to suppress these high frequencies, in this work we will use these aliased frequencies to obtain additional information, since our sample rate is limited with respect to the notes played on the guitars. Figure 2.2 illustrates the phenomenon of aliasing.



(a) (b)

Figure 2.2: **(a)**: The spectrum of the note A4 ($f_0 = 440$ Hz), with a sample rate of over 20 kHz. **(b)**: The spectrum of the note A4 (440 Hz), with a sample rate of 2756 Hz. Frequencies above 1378 Hz, such as the fourth, fifth and sixth harmonics are aliased. For example, the alias of fourth harmonic $h_4 = 4f_0 = 1760$ is $996 = |1760 - 2756|$.

2.2.2 Temporal Information

As mentioned above, the musical tone is a complex signal that contains several frequencies. Thus, Fourier analysis is well-suited for the task of identifying a note, by braking it to its frequency components. Although this simple transform is sufficient for several music applications (such as a tuner), it does not give any temporal information. That is, when performed on a long signal, it will reveal the frequency components throughout the entire signal. In musical-analysis cases, if a signal represents an entire musical piece, this analysis might be practically meaningless, as the frequencies throughout a musical piece are many and vary frequently. Thus, a time–frequency analysis is the main practice in music related signal processing. Mainly, it involves braking the signal into small, usually overlapping signals and applying a Fourier transform (or other time-to-frequency transforms) to each one, to create a time-frequency representation.

Short-Term Fourier Transform

The most basic and well-used transform is *Short Term Fourier Transform (STFT)*, which breaks the discrete samples of the signal $x[t]$ of length M to equal-length segments (with length N), and computes FFT on each one to reveal the frequency components in a given time.

Each of those segments is first multiplied by a window function, $w[t]$ of size N [1], and then applied with FFT. To reduce artifacts, those windows are usually

[1] we take the number of DFT points in each segments to equal to N

applied with some overlap, r. Figure 2.4 shows the STFT applied on an analog signal following the sampling process.

The Spectrogram

A general spectrogram is a visual representation of a signal that shows its frequencies information variation over time (or some other variable). A spectrogram representation of the STFT applied to a signal correspond to taking the squared magnitude of the STFT. In this case, the spectrogram is a 2D map, $T \times f$, where each column of the map is the power spectral function of each segment. More specifically, the spectrogram is a $m \times k$ matrix, α. The number of columns is given by $k = (\lfloor (M–r)/(N–r) \rfloor)$. The frequency bin size f_{BS} is given by $f_{BS} = f_s/N$. It follows that the number of spectrogram rows is given by $m = (n/2 + 1)$ for an even N and $(N + 1)/2$ for an odd N, since the applicable frequencies ranges from zero to $f_s/2$. Finally, the value at an entry $\alpha(t_i, f_j)$ is the magnitude of the frequency f_j computed for the interval with size N of the signal starting from $(t_i - 1)(M - r)$ (see Figures 2.4 and 4.3a).

2.3 The Guitar

The guitar is a string instrument, usually fretted, that is played by strumming the strings or plucking them. For a standard guitar, the right hand is the strumming/plucking hand whilst the finger of the left are fretting it. Typically, all guitar strings have the same length. However, their thickness varies; higher strings are thinner, which causes the string to produce a higher set of frequencies. Additionally, Shortening a string using the left hand makes a string produce higher frequencies. See guitar structure and components in Figure 2.3

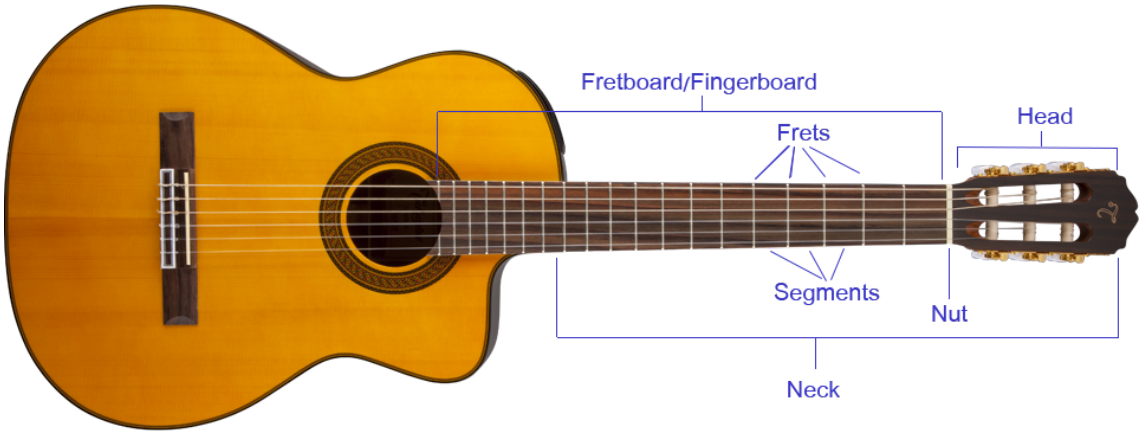


Figure 2.3: The key components and structure of a guitar.

The guitar's *frets* are metallic straight strips that divide the fretboard into *segments*. The ratio of the spacing of two consecutive frets is $\sqrt[12]{2}$. This ratio makes the difference in notes produced by playing consecutive segments to be a semitone. Pressing the string against a higher segment will shorten the string to the next closest fret and make it produce a higher pitched note. We will use the term fret to address the segment that is behind it, as is customary among musicians. For

example, when naming the second fret we mean the segment between the first fret and the second one.

2.4 Musical Notation

Musical notation is a representation of a played (or sung) music by instruments (or human voice) using some form of musical symbols. Musical notation include various properties of the musical piece. The most important ones include the notes temporal information and pitch information. Other properties may include tempo, key, dynamics, etc. Modern staff notation is the most known form of musical notation, which usually include the majority of these properties, and are usually used for classical instruments (like brass and bowed string instruments), instruments with complex polyphony (such as a piano) or for multiple instruments (like an orchestra or a quartet). An example of a modern staff is shown in Figure 2.5a. However, some instruments or different methods of playing require little understanding of these complex musical notation or some note properties are redundant. Guitar's common musical notation is the *tablature*. This notation includes merely the pitch information by notating the number of string and fret needed to be pressed in order to produce it. Temporally, the notes' orders is given, and usually some additional temporal information regarding the notes' onsets and offsets is added, either by the space between the these numbers (where a small space between notes indicated to play them closer together and vice versa), or by other simple temporal notation as in modern staff notation. This notation can be sufficient to describe a large majority of music for guitars. See example of a tablature in Figure 2.5c.

Music being computerized formed a new way of interfacing and communicating between electronic music instruments and computers. This formed *MIDI (Musical Instrument Digital Interface)* which is a technical standard that carries information about events of musical notes. For example, a MIDI keyboard controller can be connected to a computer and a software will identify each key press. MIDI mainly carries event messages that correspond to temporal and pitch information. MIDI representation will usually consist of a matrix, where the columns (x-axis) represents the time sequence and the rows (y-axis) represent the pitch. Every note in this representation is an horizontal bar. The bar's y-axis coordinates corresponds to the pitch of the note. The bar's x-axis coordinates corresponds its occurrence in time, where the left edge indicates the start of the note and the right one indicates its end (and accordingly its length corresponds to the note's duration).

In this work, we will use MIDI-like representation as it is closely related to the time-frequency representation obtained using the STFT and since it is suitable for guitar playing (especially for bass and for melody playing in guitars). Furthermore, this representation describes best the features obtained by NT. Since we analyze each string separately, the transition to a tablature is trivial. An example of a MIDI representation is presented in Figure 2.5b.

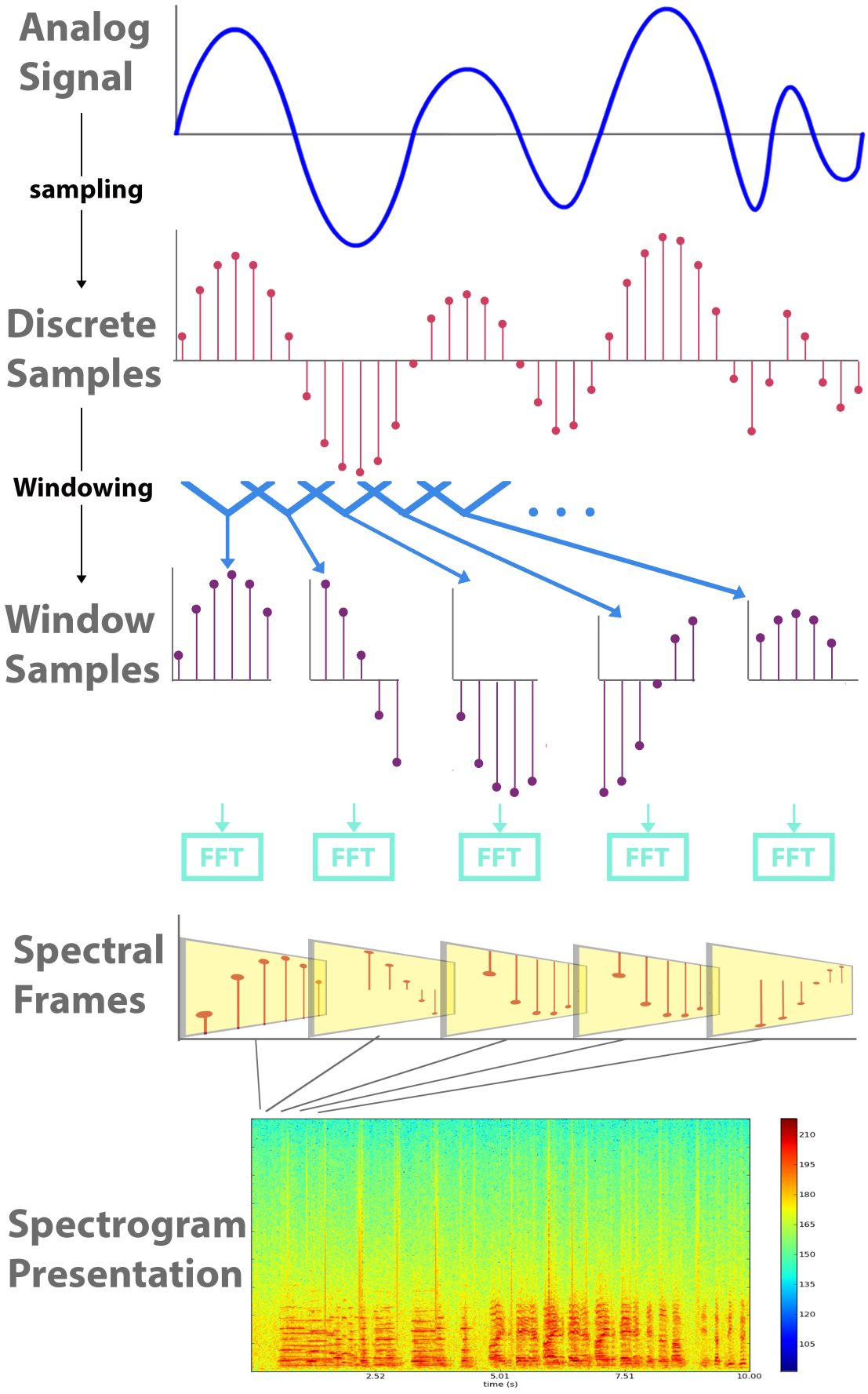
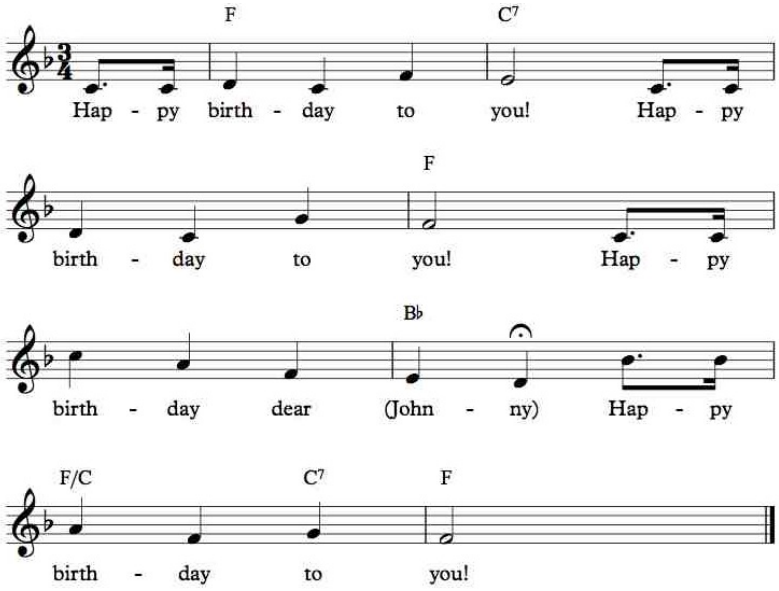
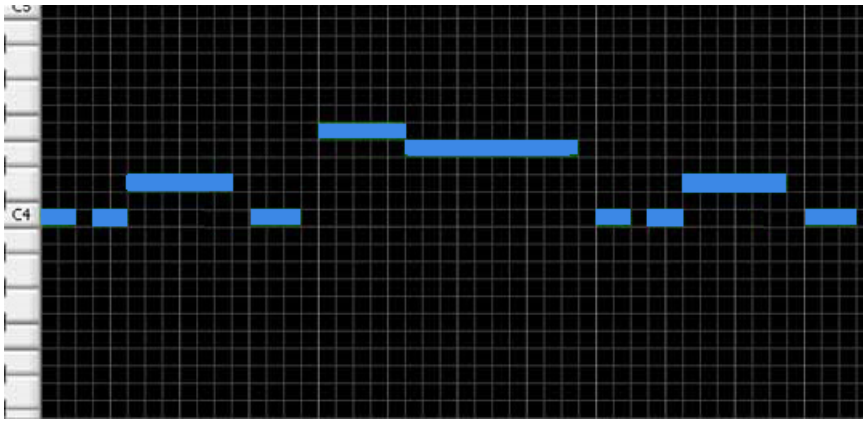


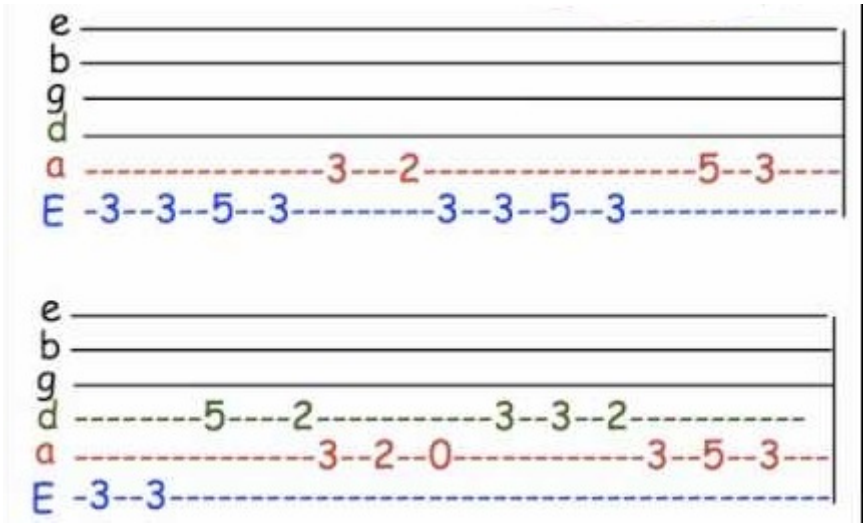
Figure 2.4: The short-term Fourier transform process.



(a) Modern staff notation of "Happy Birthday".



(b) MIDI representation of "Happy Birthday".



(c) Tablature representation of "Happy Birthday".

Figure 2.5: Different representations of the "Happy Birthday" song.

Chapter 3

Related Work

Our work was inspired by the innovative study by Davis *et al.* [4], where sounds were recovered from silent videos. In their study, they showed that when sound hits an object, its surface vibrates subtly, and those vibrations can be visually detected by a camera. Furthermore, it is possible to reconstruct the original sound by processing the signals obtained from those vibrations. However, NT was beyond the scope of their study, since they did not focus specifically on music signals or attempted to analyze them. Wu *et al.* [9] suggested to look at a certain location in the image and extract the temporal color changes, in order to amplify it to make it visible to the naked eye. Rubinstein [10] demonstrates that a temporal color change obtained in a certain location on a vibrating string, can reveal its vibration frequency solely from the video. These studies used an extremely high frame rate camera or made use of the rolling shutter properties to effectively increase the frame rate, which is inapplicable in our work, as our signal must be captured in a single pixel. Another notable study is that of Owens *et al.* [11], where supervised learning is used to predict the sound of an object being hit or scratched in a scene from a silent video. It demonstrates that audio information could be retrieved by visual data alone, but without any attempt to analyze the physics of the scene.

Naturally, most applications use auditory means to solve the AMT problem (see surveys [12, 13, 14, 15, 16]). These applications use signal processing methods to analyze the audio signal. But although AMT is a fundamental field in music information retrieval, only monophonic AMT is considered solved. AMT for polyphonic music is a more complicated problem and the state-of-the-art AMT systems still perform well-below the level of humans, or is restricted to the degree of polyphony or instruments type [12]. Furthermore, audio signals can partially obfuscate some essential information that can be observed almost only visually, such as fingering or even what string is used to play a certain note (as a note can be played on several strings). Although several works did address those issues, they are limited in several aspects, e.g., by the set of pre-defined structured chords or by the level of polyphony [17, 18], an isolated single note [19], or a single instrument playing solely [18, 20].

Visual and hybrid (visual combined with audio-based) approaches were attempted to solve the AMT problem, due to the limitations of the audio-based methods. As the visual information obtained usually consist of the physical manner the player is operating the instrument, visual methods are mainly instrument specific. Methods that focus of the guitar mostly follow the same technique of detecting the guitar's

strings and frets and then locating the guitarist's left hand. Most methods ignore the player's right hand (the plucking / strumming hand).

A key component of methods that use visual infomration is to locate the guitar in the video or frame. Straight forward approaches use the common structure of the guitar to locate it in the frame. For example, Pelari *et al.* [21, 22] and Quested *et al.* [23] detected the straight lines of the of frets. Common trackers were used to track some guitar-specific structural elements, such as the constant ratio of the distance between frets. In both these hybrid systems a rough location of the left hand was obtained to retrive the fingering information (string and fret) for each detected note by the audio analysis system. Similar but more comprehensive systems were presented by Cheung and Lee [24] and Scarr and Green [25]. These solely vision-based systems use similar characteristics of the guitar to locate the fretboard and more specifically the exact location of each string and fret. Locating of the fretboard is executed on each frame separately and no tracking is used. Zhang *et al.* [26] used the similar structural elements of a violin for an AMT application.

Applications that bypassed the challenge of tracking the fretboard or to relocate it in each frame used, as we do, a mounted camera [3, 27, 28]. In this case, the strings and the frets need to be detected only once, as they stay static in relation to the camera throughout the entire video. Others [2, 29, 30, 31, 32] used markers on the guitar in order to track it. Our method use a mounted camera, and the strings need to be only once detected. The detection is based on their vibration in a video rather than using the geometric structure of the guitar from a single image (see Section 4.3).

Given the location of the guitar, the frets, and the strings, the next challenge is to identify the left hand and/or fingers of the player. In this task as well, some method used marking on the fingers [2, 31]. Most method that avoid markings usually use either detecting of the skin color [1, 21, 23, 25, 22, 24, 26] and/or search for the circular shape of the finger tips by using circular hough transform [1, 3, 24, 26, 27, 28]. Hrybyk and Kim [29] trained a set of rectified images of the fretboard to identify a specific voicing of a chord.

Although visual and hybrid methods can diminish some limitations derived from solely audio-based applications, these methods are not without their shortcomings; first, since the detection of the frets is executed by measuring the distance from the nut, it is required to have a sufficiently large view of the fretboard that includes the frets that are played on, starting from the 1st fret. Some methods are limited either by the number of frets considered (e.g., [3]) or by a pre-known set of possible chords or voicing options (e.g., [29]). Our method requires only a small portion of the strings to be visible.

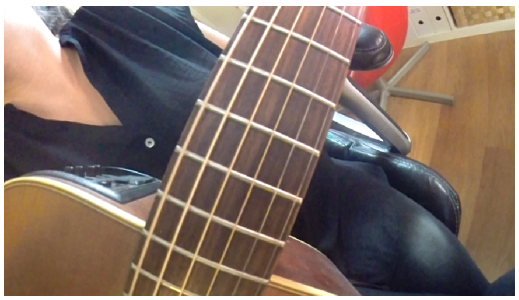
Another main limitation of visual-based method is the detection of when a note was played, if at all, as discussed earlier. Zhang *et al.* [26] attempt to overcome this limitation by measuring the distance between the strings to its closest finger, in an attempt to determine the finger that is pressing the string. However, they concluded that, due to poor results, a single 2D camera is insufficient to obtain this kind of information. Another notable exception is the work of Wang and Ohya [1] that detects "key frames" in which the player's hand is moving toward a new chord, thus making it possible to roughly determine when a chord is not being played and when it is. However, this approach does not solve the problem of a specific strumming or picking and is irrelevant if the player is playing a melody rather than chords.

Chapter 4

The Note Tracking System

In this chapter, we present our note tracking method. The input is a video taken by a camera mounted on the guitar, as shown in Figure 4.1. The output is the NT that consists of the temporal segmentation of the played notes as well as their pitch.

(a) A camera mounted on the guitar



(b) A frame captured by the mounted camera

Figure 4.1: Our system setup.

In Section 4.3 we present our method to detect a set of *string-pixels* for each string. A string-pixel q_s of string s is roughly located on the projection of s. The string-pixels should be located only once, because their locations are fixed with respect to the camera; the change in the strings locations is only due to their vibration. The intensity of q_s as a function of time, $q_s(t)$, is correlated with the frequency of the string vibrations (see Figure 1.1). For each string s, we compute the NT of the music played using a set of its string-pixels signals, $\{q_i^s(t)\}$, as an input. The temporal segmentation algorithm is described in Section 4.1.2. Then, for each segmented note, the algorithm for computing the fundamental frequency (the note's pitch) is described in Section 4.2. Robustness of the NT is obtained by voting from several string-pixels of the specific string. The pitch and temporal information of notes from all the strings can then be rearranged to form a coherent musical sheet (MIDI or tablature-like).

A full diagram of the proposed method is given in Figure 4.8.

4.1 Temporal Information

In this section we present our method to compute the temporal information of the played music on a given string. It consists of the onset and offset of each note. Our goal is to divide the set of input signals, $\{q_i^s(t)\}$, into a set of segments, each corresponding to a single note played on the string s. We represent each of the input signals in a time-frequency representation, a spectrogram, to achieve both temporal and pitch information. However, since temporal resolution and frequency resolution are complementary, we favor the temporal information and largely ignore the pitch information obtained by this representation. The computed temporal information is then used to compute the frequency of each played note (see Section 4.2).

4.1.1 Spectrograms Generation

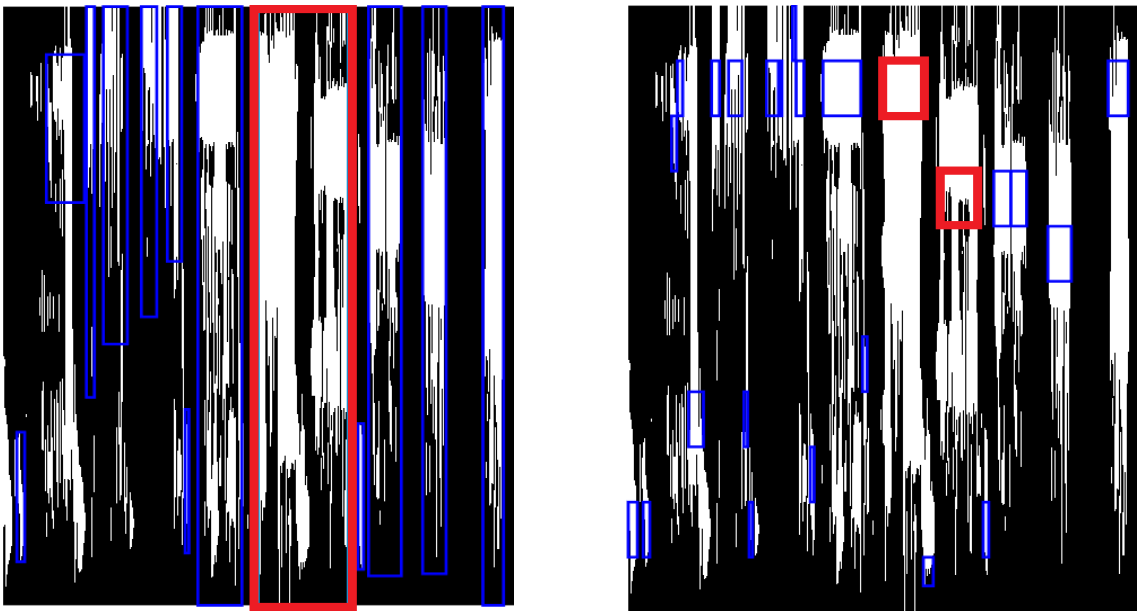
We calculate the short-term Fourier transform of each of the signals separately and represent it as a spectrogram (see Section 2.2.2). That is, an FFT is applied to time intervals of length N of the original signal. The size N should be sufficiently large for obtaining high energy at the fundamental frequency of the played note as well as for improving resolution of the computed frequencies. On the other hand, small N improves the time resolution, and it is more likely that only a single note is played during a short time interval. To improve the time resolution, overlapping intervals can be considered. In our implementation, we use overlapping time intervals and trade frequency resolution for temporal resolution, since the temporal information is more imperative in this stage. Thus, we use a rectangular window with $N = 20$, and the maximal overlap between the signal's time intervals ($r = 19$).

4.1.2 Note Temporal Segmentation

Since each $q_i^s(t)$ is a signal obtained from a single string, only a single note is played at a given time. Moreover, for a sufficiently small time interval N, only a single note is present. Hence, at the time interval during which a note is played, the spectrogram is expected to have a region with high energy roughly centered around the played frequency. High energy of the spectrogram at the same temporal interval is also expected at the visible frequencies that correspond to the note's harmonics, and the aliased frequencies of both the note and its harmonics (see Section 4.2). Other regions of the spectrogram are expected to have low energy, except for noisy regions, typically at low frequencies. In our case, we consider the noise range to be $\leq 20\,\text{Hz}$. To avoid the detection of noise, the region of the spectrogram that contains frequencies equal to or below 20 Hz is discarded.

We use a naive segmentation method – applying a threshold to the spectrogram energy. A bounding box is set around each region above the threshold. The bounding box coordinates in the temporal axis correspond to a temporal segment, denoted by $\tau_j = (t_j, t_j + \delta_{t_j})$.

The large frequency bins, the spectral leakage phenomenon, and the fact that some harmonics can appear as aliases and might overlap unaliased harmonics, can cause the high energies to smear across the frequency axis. This, in addition to noisy areas in the spectrogram, can result in temporally adjacent segments to be falsely merged to be included in a single bounding box.



(a) (b)

Figure 4.2: Spectrograms that were applied with bounding boxes. (a) Applying the bounding boxes on the entire spectrogram. Some bounding boxes can be falsely merged to include two notes, as in the red bounding box. (b) Applying bounding boxes on horizontal stripes of the spectrogram, which resulted in individually detection of the notes, in the red bounding boxes.

Hence, the segmentation is performed separately on each frequency bin of the spectrogram. That is, the spectrogram image is divided to horizontal stripes according to the frequency bins size, and each is set with the bounding boxes. This causes large energy areas representing a note to be divided to several, roughly temporally equal bounding boxes. Since we merely wish to extract temporal information in this stage, we can ignore the height of the bounding box which eventually be selected to represent the temporal segmentation of the note. An example of this sectioning if showed in Figure 4.2.

As a post-processing step, we discard overlapping temporal segments, by choosing the more dominant regions, as at each given time only one note can be played. Additionally, small segments are also discarded. The threshold is chosen to be the ranked 80% of the energy values in the spectrogram, since we assume that the played notes take roughly 20% of the spectrum at each given time (See Figure 4.3).

This algorithm is applied to each $q_i^s(t)$. As expected, the set of temporal segments obtained from the set $\{q_s^i(t)\}$ (for a gives string s) varies due to the quality each of the signals. This, since different locations along the string captures its vibration differently (see example in Figure 4.4). We use the results obtained from the signals of all string-pixels of a given string to vote for its temporal segmentation; the set of frames that appear in at least 65% of the signals are chosen.

Clearly, a more sophisticated segmentation algorithm can be used for this stage of our algorithm. It is left for future research to study how advanced segmentation methods can improve our results.

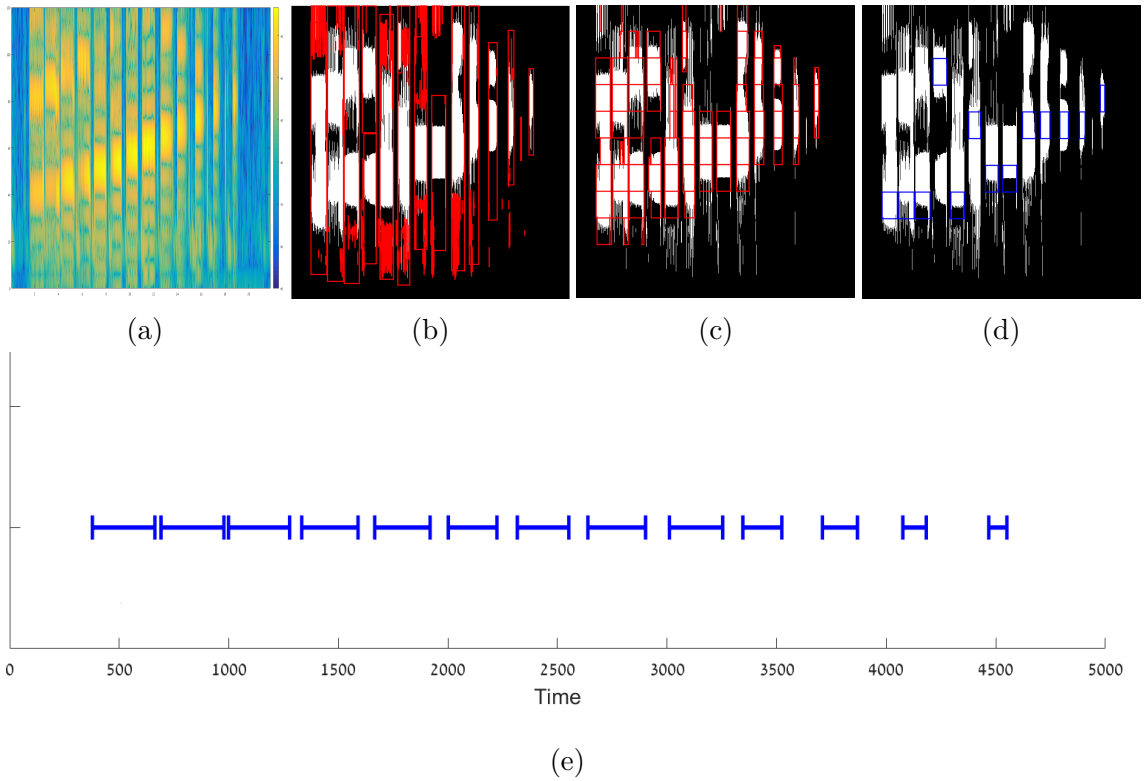


Figure 4.3: (a) The spectrogram computed for a chromatic scale played on E string of the bass guitar. (b) The initial bounding boxes placed around the thresholded spectrogram (c) The horizontal sectioning and discarding of noise.(d) The discarding of time-overlapped bounding boxes, before discarding of the bounding boxes with no meaningful signal. (e) The final temporal segments.

4.2 Pitch Detection

Computing the note's fundamental frequency is a main goal of NT. Many audio-based methods compute the note's pitch by selecting frequencies with high energy in the spectrogram of the audio signal and matching it to the known structure of the notes' frequency components (the corresponding harmonic series). In our case, these frequency components are largely above the spectrogram frequency range due to the low frame rate. More importantly, the frequency resolution of the spectrogram is poor thus frequency bins are too coarse to allow distinction between close frequencies (e.g., a frequency bin of 1 hertz will not allow distinction between frequencies of 81.4 Hz and 82 Hz assuming the fall in the same bin). The poor frequency resolution is caused by our choice of a small time interval size, N, which guarantees high temporal resolution at the cost of low frequency resolution.

To improve the frequency resolution, FFT can be applied to a longer time interval that consists of a single note computed by the temporal note segmentation (Section 4.1.2). Indeed, the temporal note segments are typically longer than the chosen N.

Our goal is to compute the fundamental frequency, f_0, for each pre-calculated note temporal segmentation. This, by applying FFT on each of those segments, which results in obtaining the power spectra function, $S(f)$, for each string-pixel. Finally, a majority voting is computed on all string-pixels for each note to determine

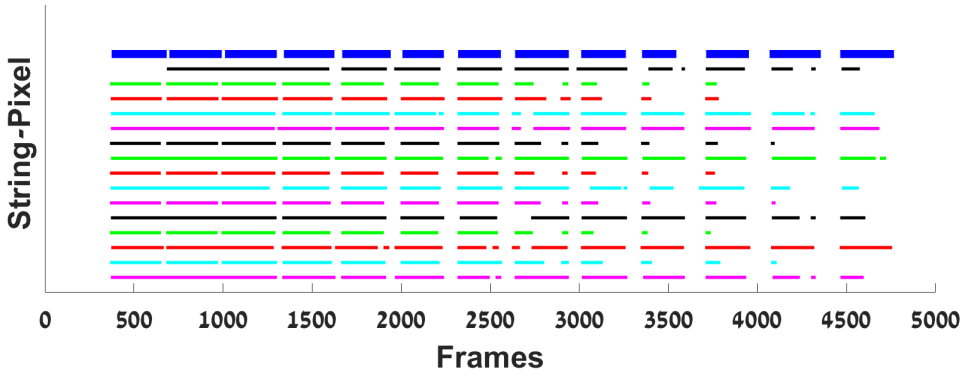


Figure 4.4: Several string-pixels' temporal note segmentation. Each row represents the temporal segmentation of the signal computed using a single string-pixel. The upper row (blue) is the ground truth.

each note's pitch.

The main challenge is the low sampling rate of the video (240fps) with respect to the fundamentals we wish to recover (up to 392 Hz), let alone for their second and third harmonics (up to 1176 Hz). According to the Nyquist-Shannon sample rate theorem [7, 8], 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 (see Section 2.2.1).

Peaks of $S(f)$: A peak of $S(f)$ is expected at the note's fundamental frequency, 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 (see definition in Section 2.2.1). That is, there exists $k \in \mathbb{N}^+$ such that $f_A(f_0) = |f - kf_s| < f_s/2$. We define the visible frequency of f (given f_s) to be

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

Hence, if $S(f)$ has a single peak at f', we could infer that it is either at f_0 or that it is an aliased frequency of f_0 given by the set $\{|f_0 \pm k \cdot fps|\}_{k \in N}$ (as long as $f_v(f_0)$ is not in the noise range).

Additional peaks are expected at the note's harmonics (or their visible aliased frequencies). Formally, let the harmonic series set of f_0 be given by $\{h_i | h_i = if_0\}_{i \in \mathbb{N}^+}$ (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 energy of a harmonic usually decreases as i increases (see Figure 2.1b); hence we consider only the fundamental frequencies and two lowest 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 ambiguities, since one needs to discriminate between f_0, its aliased frequency, 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, observing a frequency of 110 Hz can be either because a note has a fundamental

frequency of 110 Hz, because it is a (second) harmonic of a note with fundamental frequency of 55 Hz (as $2 * 55 = 110$), or because it is an aliased frequency of 130 Hz (as $240 - 130 = 110$). Moreover, our limited bandwidth could cause peaks to be very close together and thus harder to distinguish. For example, there is only 0.8 Hz difference between the peak of the note A2, where $f_0 = f_v(f_0) = 110hz$, and the peak of C3, where $f_0 = 130.8$ and $f_v(f_0) = 109.2hz$. Also, a slight out of tune guitar or a hard picking of the string can cause some variation of the note's pitch. Figure 4.5 illustrated this property.

In addition to the above-mentioned expected peaks, some peaks will not be visible because they are within the noise range (≤ 20 Hz), and peaks may be shifted due to $S(f)$ resolution.

Using Prior Knowledge: Our goal is to compute f_0 despite the expected above-mentioned ambiguities. The most significant information we have is that $S(f)$ consists of only a single note. Moreover, the set of possible fundamental frequencies that can be played on a given string s, $F_{string}(s)$, is also known (as long as the

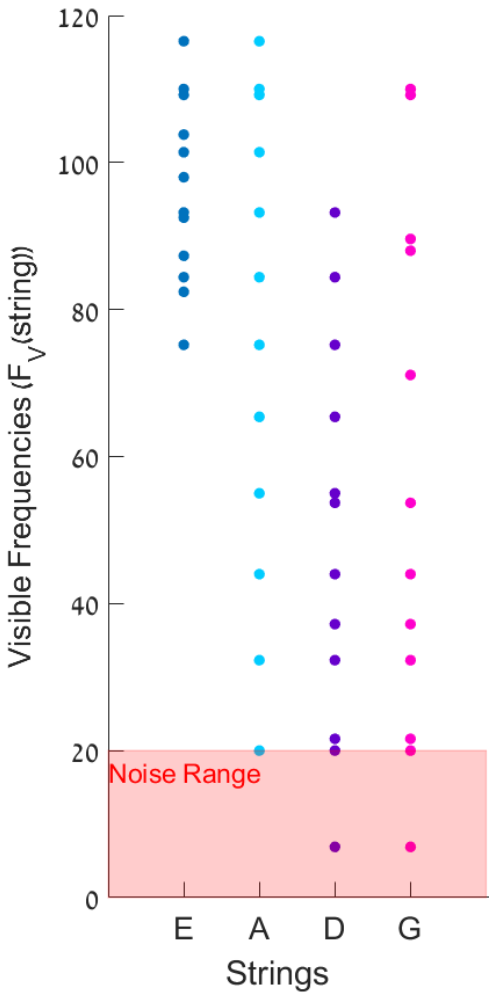


Figure 4.5: The visible fundamentals of each string for the standard guitars. Note that the visible fundamentals on string E are relatively dense while the ones of A string are sparse.

instrument is tuned). 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 the computed set of peaks, and we choose the note with the highest score.

Observe that in audio analysis, no aliasing exists since the bandwidth is sufficiently large or an anti-aliasing filter can be used. However, harmonics frequencies are visible, and it might be hard to discriminate, for example, whether a single note is played or both the note and its harmonics are played. We do not have to handle these challenges that arise from such ambiguity, since only a single note is played at a given temporal segment on a given string.

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 $f_h \in F_V(f_0)$ from its nearest peak in $F_{peaks}(S(f))$. We normalize the distances between f_h 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 $d(f, f^p) = |(f - f^p)|/|(f - f')|$. In a similar manner 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

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

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 has a fundamental that is the second 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 of the played note, or vice versa. To avoid such errors, we reexamine which one of them 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%.

Post-Processing After obtaining the score of each string-pixel for a specific temporal note we use a majority vote to determine the note's fundamental. Furthermore, some temporal segments can show no meaningful frequency information (i.e., no peaks are visible in the corresponding $S(f)$, see Section 4.2). This means that

no pitch was detected in the temporal segment. Thus, if the majority of the string-pixels indicated that no pitch was detected, the temporal segment is assumed to be a false-positive and is discarded.

4.2.1 Expected Failures

Given the limited frame rate used, it is possible to predict the failures of the pitch detection for each instrument.

Let us consider two notes played on the same string of a given instrument. These notes are defined to be *indistinguishable* if their corresponding harmonic series are observed identically. Strictly, let f' and f'' be two notes with fundamentals of f'_0, f''_0 such that $f'_0 \neq f''_0$ (and accordingly $H_n^{f'_0} \neq H_n^{f''_0}$), then f' and f'' are indistinguishable if $F_v(f'_0) = F_v(f''_0)$.

For example, the fundamentals of the notes B♭2 and B2 are 116.5 Hz and 123.5 Hz, respectively. For B♭2: $f_v(116.5) = 116.5$, $h_2 = 233\,\text{Hz} \Rightarrow f_v(h_2) = |233 - 1 \cdot 240| = 7$, $h_3 = 349.5\,\text{Hz} \Rightarrow f_v(h_3) = |349.5 - 1 \cdot 240| = 109.5$, and or B2: $f_v(123.5) = |123.5 - 1 \cdot 240| = 116.5$, $h_2 = 247\,\text{Hz} \Rightarrow f_v(h_2) = |247 - 1 \cdot 240| = 7$, $h_3 = 370.5\,\text{Hz} \Rightarrow f_v(h_3) = |370.5 - 2 \cdot 240| = 109.5$ (See 4.2 for computation explanation). Example of such notes on the classic guitar is given in Figure 4.6.

We clearly cannot recover a frequency below the defined noise range. Hence, notes that both f_0 and h_2 are in the noise range their pitch cannot be detected. Moreover, neither their temporal information can be detected. Theoretically, h_3 can be detected but in practice the peak of the high harmonic is too weak for detecting such notes.

Specifically, there are two pairs of indistinguishable notes. The first pair, B♭2 and B2 (116.5 Hz and 123.5 Hz) can be produced in the bass guitar in strings D and G and, and by all other guitars in strings E and A. The second pair, B♭3 and B3 ($f_0 = 233.1\,\text{Hz}, 246.9\,\text{Hz}$), which are played on all guitars but the bass guitar, are also indistinguishable but more importantly, both the fundamental and second harmonic are within the noise range. This make them practically impossible to detect since, as mentioned above, the peak in h_3 is usually insignificant. Figure 4.6 illustrates the indistinguishable notes in the standard guitar.

The note A3 (f_0 =220 Hz, $f_v(220) = 20$) has its fundamental in the noise range, but its second harmonic, $f_v(2f_0) = 40$, is detectable. Some notes has some of their harmonics in the noise range, which might cause some detection issues, mainly in the pitch detection phase. Tables 4.1 and 4.2 summarize the visibility properties of the notes per string and fret, respectively. Tables A.2 and A.1 show the visibility of each note and its string and fret location on each of the instruments.

Since the frequency resolution is relatively low, we also expect ambiguities for similar frequencies rather than identical ones. For example, A2 ($f_0^1 = 110\,\text{Hz}$) and C3 ($f_0^2 = 130.8\,\text{Hz}$) have very close visible frequencies of the fundamental, that is $f_v(f_0^1) = 110, f_v(f_0^2) = 109.2$. These notes second and third harmonics are also relatively close ($f_v(h_1^1) = 20$ and $f_v(h_1^2) = 21.6$. Also $f_v(h_2^1) = 90$ and $f_v(h_2^2) = 87.6$).

4.3 Strings Detection

We present two algorithms for locating the string-pixels. The first is a classic, geometric approach for string detection which is based on a single frame. The

String	% f_0 (only) in noise range	% f_1 (only) in the noise range	% both f_0 and f_1 in the noise range
E(1)	0	17	0
A(2)	6	20	0
D(3)	6	6	12
G(4)	6	12	12

Table 4.1: Notes visibility properties, per string, up to the 12th fret.

Fret	% f_0 (only) in noise range	% f_1 (only) in the noise range	% both f_0 and f_1 in the noise range
0	0	16.67	0
1	0	16.67	0
2	16.67	25	0
3	0	8.34	16.67
4	0	8.34	16.67
5	0	16.67	0
6	0	16.67	0
7	16.67	25	0
8	0	8.34	16.67
9	0	8.34	16.67
10	0	0	0
11	0	16.67	0
12	16.67	8.34	0

Table 4.2: Notes visibility properties, per fret.

second is a temporal-spectral approach, using the temporal data from the video. Note, that we assume that the string-pixels obtained are not obfuscated throughout the entire video. However, if only few of the string-pixels are partially invisible, this should not effect our results, as multiple string-pixels are processed and a majority

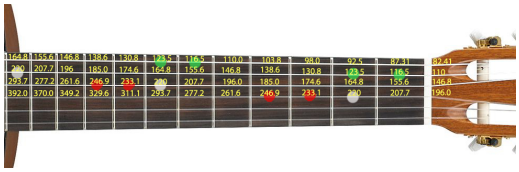
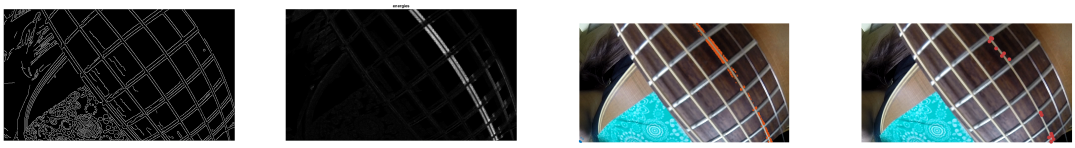


Figure 4.6: The fundamental frequencies of the standard guitars. The pairs of red and green dots mark a pair of indistinguishable notes. Marked with white dots are notes whose fundamental frequency is within the noise range. Frequencies 233.1 and 246.9 are marked as indistinguishable although both their f_0 and h_2 are within the noise range, which practically makes the undetectable.



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

Figure 4.7: The process of locating the string-pixels. **(a)** The edge map of the guitar image (without dilation). **(b)** The energy map of 53.7 Hz (which is the aliased frequency of 293 Hz, note D4 on string G). **(c)** The pixels with highest energy for the computed frequency. **(d)** The random chosen subset of string-pixels from the set presented in (c).

voting is used, thus making our method more robust. Moreover, a small adjustment of our method can assure that the number of visible string-pixels at a given time of the video is sufficient, for example by using more string-pixels.

4.3.1 A Geometric-Based Algorithm

The guitar's strings and frets are approximately straight lines. It follows that the strings should be selected from a set of image lines. We assume that a region in the video frame contains the guitar neck, although part of it can be obfuscated by the players hand. The guitar's neck consists of frets and strings, which form a grid-like structure that can point us to the location of the strings. Multiple methods where attempted: Hough transform was used to detect straight lines the image, than only roughly perpendicular lines where chosen; Harris corner detector was attempted to find the corners created in the intersection between a fret and a string; and more. However, these methods offered only limited success and are not applicable unless strong assumptions regarding the scene and the visible part of the guitar are made.

4.3.2 A Temporal-Spectral Based Approach

We propose to use frequency information available from the video, and detect each string according to expected frequency (e.g., by an initialization video that captures pre-known visible notes). When a string vibrates in a known frequency, some pixels around or along the string are expected to have high energy in their frequency spectrum at that known frequency. Thus, those pixels capture the string vibration, and can be detected by searching high energy pixels after filtering the frequency spectrum to the desired frequency. Practically, an FFT is computed on the dilated edge image (Figure 4.7a) to reduce the computation time, and the string-pixels are detected by a filter on the expected frequency (see example Figure 4.7b). The string-pixels with the highest energy are chosen (Figure 4.7c) and then selected randomly to for a set of 31 string-pixels (Figure 4.7d) .

Real-data experiment revealed that even a manual locating of the strings yields inferior results to using our temporal-spectral algorithm. Thus, a geometric approach will be inferior as well (see Section 5.3). As a result, the geometric approach was abandoned and was not used for the experiments.

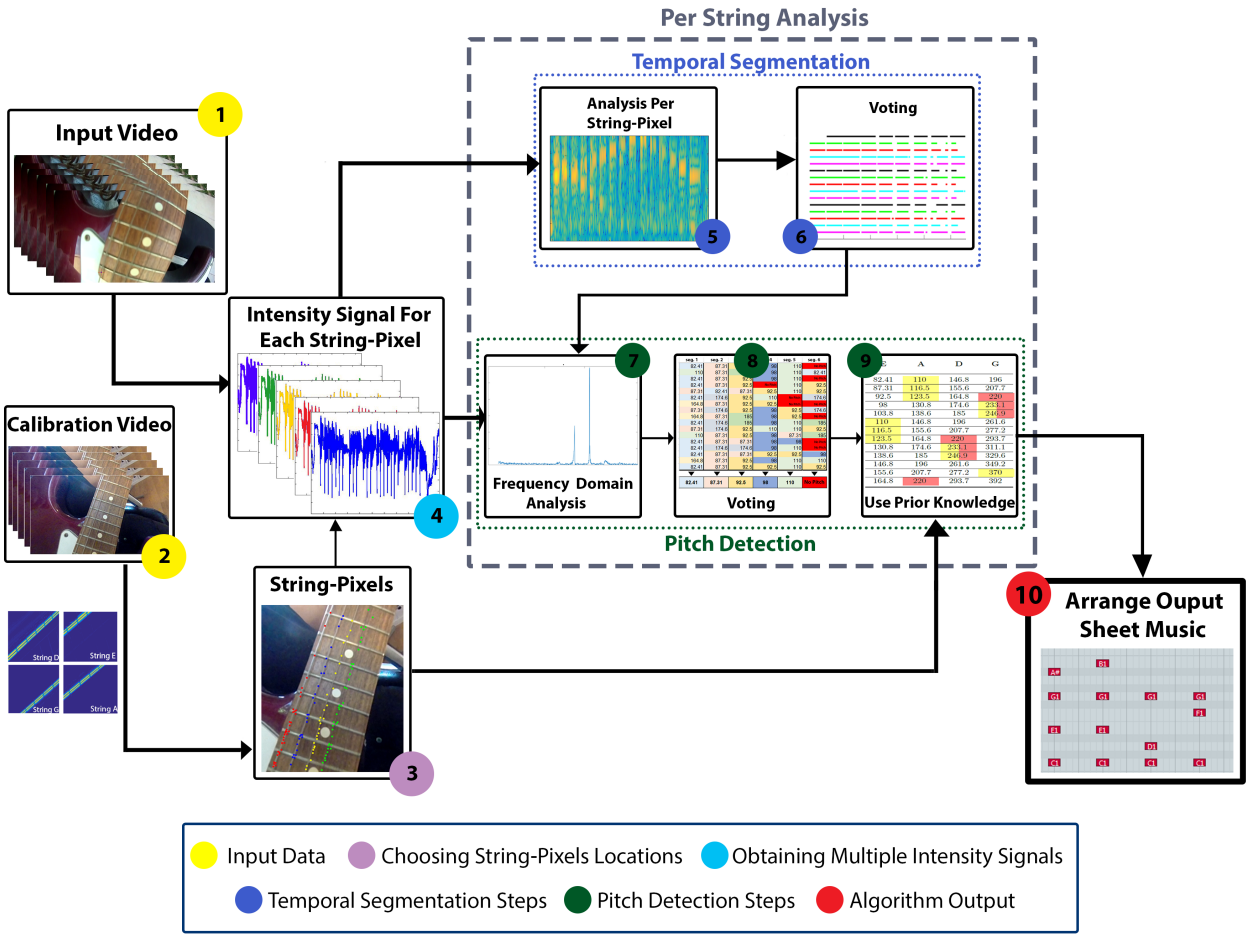


Figure 4.8: The proposed algorithm's pipeline.

Chapter 5

Experimental Results

5.1 Experiments

We tested our method on data generated by us. Existing data available on the Web is insufficient for our method since it is either captured with a low frame rate or by a camera that is not mounted on the instrument. The algorithm was implemented in Matlab, and run on a standard PC. We evaluate the performance of each component of our algorithm separately, as well as the performance from start to end of the entire system.

Since this problem has not been addressed before, there are no existing solutions with which to compare our results. Existing methods that use audio signals face challenges due to polyphonic music, which is trivial in our data, but they avoid the challenges of the low sampling rate of a camera. On the other hand, visual-based methods are not comparable to ours, since they detect only the pose of the left hand. The left hand pose is used to detect a chord that can be played, but no onset and offset are computed.

An visualization of the tests results for pitch detection and frame-by-frame evaluation is shown in Figure 5.1.

5.2 Data

We applied our method to all data described below. The videos were captured by a camera mounted to several guitars - an electric bass guitar, a classic guitar, an electric guitar, and an acoustic guitar. These guitars have strings made from different materials, and consists of a representative sample of all common strings available for guitars. We use a GoPro (model: HERO3+) camera and SJCam (similar to GoPro, with similar specifications), with 240 FPS in WVGA resolution (see Figure 4.1).

The instrument was tuned using a simple electronic tuner before data collection. The music was played by a guitar player. The videos were captured in a well-lit room, without flickering lighting (such as fluorescent). In addition, all strings were visible in the video with high enough contrast to the background, although sometimes the string shadow can also be used to obtain the same signal with higher contrast. The strumming was strong enough for obtaining a large amplitude of the string's vibration.

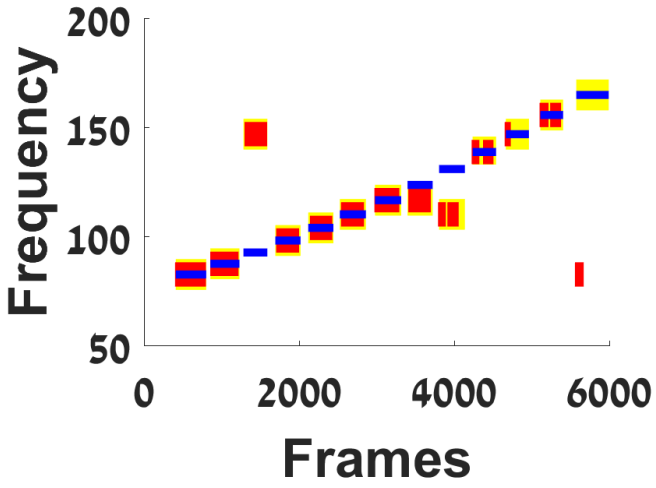


Figure 5.1: A visualisation example of the different tests results. The blue segments are the GT, the yellow ones are the pitch detected on the given GT intervals (as in Test 2),and the red ones are the automatically temporally segmented notes (as in Test 1), and their detected pitch (as in Test3 - evaluation of pitch detection). The frame-by-frame evaluation (as in Test3 - frame-by-frame evaluation) is calculated by counting only frames where the blue segments and the red ones are present and show the same frequency.

For the bass guitar we consider notes played up to the 14th fret (including, out of 20), that is, 15 notes played on each string. Higher fret notes are rarely used in practice. For the other guitars, we considered only the first 13 notes on each string, that is up to the 12th fret (out of 20), since the strings are shorter than those of the bass and thus the amplitude when strumming is smaller. Unfortunately, the resolution of our cameras does not allow to reliably capture the signal of the two highest strings of these guitars, which are very thin and/or made of a partially transparent material (i.e, nylon). Hence, we tested our method only on the lower 4 strings of these instruments out of total of six. Note that the bass guitar has only four strings. It remains to be seen whether a camera with higher resolution may allow to deal with all strings.

On each of the strings (E,A,D,G) of each of the guitars, we played four times the chromatic scale (consecutive semitones on all the considered frets) up to the 14th fret in the bass guitar and up to the 12th fret in all other three guitars. All together we played $864 = 15 \times 4 \times 4 \times 1 + 13 \times 4 \times 4 \times 3$ notes. We made no assumption on the music played, hence, this data represents well all musical pieces played in the considered range. Additionally, we demonstrate a polyphony example of a chord playing in a guitar.

5.3 Calibration

The temporal-spectral algorithm was used to detect the string-pixels in each of the videos (see Section 4.3.2). For calibration, for each video, we used a known temporal interval with a known played note. In practice, a calibration video can

also be processed separately, by providing a video that captures the strumming of all strings with predefined notes (e.g., open string or a chord).

In the next experiments we will use the temporal-spectral string detection algorithm with a calibration performed using notes that where played on a high (6th,7th, or 8th) fret. In general, we found that calibrating using notes played on a higher fret gives more reliable results than that obtained on an open string or with manually chosen string-pixels (see Test 4. The geometric approach was abandoned since it yielded inferior results in the detection of the string.

Example of the manually detected pixels, the low-fret and the high-fret detected pixels are shown in Figure 5.5.

Test 1: Temporal Note Segmentation

Here we test the temporal segmentation of the notes. The input is a set $\{q_i^s(t)\}$, for a given string, and the output is a set of temporal intervals $\{\tau_i(s)\}$.

The ground truth is computed manually for each string s, by manually marking the starts and ends around the high-energies areas of a spectrogram. The spectrogram is computed for a single pixel of each string, $q_i^s(t)$. The known number of notes visible in the spectrogram and their frequencies are used for the manual marking. This process yields temporal intervals (start and end frames) for each played note. It is important to note that this manual extraction of the ground truth is considerably inaccurate, as the exact frame a note is starting and ending is often unclear. In particular, notes tend to fade out. We do not use existing audio methods for ground truth extraction since they often use priors on the played music, which does not reflect the actual signal. Such priors could also be used in our method as a post-processing. However, our goal here is to study the information that can be extracted directly from the visual signal.

For quantitative evaluation of the onsets, we use the same methodology as in MIREX [33]. That is, an onset is considered if it is within a tolerance time window around the ground truth defined by a threshold. Note, that this test merely requires a general note to be detected regardless of the correctness of the pitch. Given a detected onset, we consider its offset to be correct if it is within a tolerance time window around the ground truth. When more than a single onset is detected for a given note, we use the one which is closest to the ground truth for the offset evaluation. The others are considered as false positives for the onset detection. Note that errors in the offset values may be due to the choice of the matched interval based on the onset.

The temporal interval detection may miss a note with low energy or with frequency components in the noise range. It may also split a note into two or more intervals, which causes false positive onset detection and an offset error. Noisy data may cause false positive detection of a note. Finally, fading out of a note may cause offset errors.

Four different thresholds are considered, 12, 24, 36 and 80. The 12 frames threshold corresponds to 50ms. The 50ms threshold is typically used in the MIREX onsets detection [33]. It corresponds to 2205 samples in audio recorded at 44.1 kHz. An 80 frames threshold (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 chords playing. Hence, a threshold of 80 frames is acceptable in the data

resolution available by the considered camera.

The recall of the onset detection, using 80 as a threshold, is 0.88 (true-positive). The precision with the same threshold is given by 0.74. This results in F-measure of 0.8. In addition, 70% of the notes which onsets were detected were also successfully matched with offsets. The detailed results for onset and offset for the bass guitar are presented in Table 5.2 and for each of the other guitars in Table 5.1. In addition we present the sum of onset errors (false-negatives) per fret and string over all guitars in Table 5.3. Note that in this table, only the first 13 notes (out of 15) for each string of the bass are considered, for compatibility with the tests on the other guitars. These results are computed using 80 as a threshold. For completeness the results of the other thresholds for the onset detection are presented in Appendix B.

The results of our method cannot be compared to those obtained by the state-of-the-art MIREX 2016 [34], since the data reported consists of different musical instruments (including unpitched ones), different musical pieces, and a different annotation technique. Most importantly, the signal's sources are different (audio vs. video). However, it is interesting to note that the F-measure reported by the MIREX (2016) is 0.87 which only 0.07% better than our results which are obtained using naive segmentation.

Generally lower strings and lower frets yield better results (see Table 5.3). The two lower strings achieved around 4% errors (of played notes on both strings), while the higher two yielded around 19%. Frets under the 9th fret had less than 8% errors and 9-12 frets had more than double that, with close to 20% errors. Lower strings has several advantages for notes extraction over higher ones. First, lower strings are thicker, which makes the visual signal clearer and stronger. In particular, the classic guitar G string is made from nylon which has some transparency. Additionally, notes with f_0 in the noise range are less likely to be detected, since the first harmonic is typically the one with the highest energy. Moreover, some notes have both their f_0 and h_2 in the noise range, which make them almost impossible to detect, as the third harmonic alone is unlikely to have a high energy. These imperfectly seen notes are more present in the two higher strings. Table 4.1 summarize the imperfectly seen notes for each string of the instruments. As can be seen, the two higher strings have 17% of the notes with (at least) f_0 in the noise range, as opposed to 3% in the two lower ones. Note that notes with merely h_2 in the noise range should be detected by the onset detector, as their f_0 is not within the noise range and is ordinarily visible.

The extracted signal from high fret notes are inferior to the one extracted from lower frets. This is due to the fact that higher frets corresponds with shorter strings, thus vibration amplitude is smaller (assuming same force is applied for the displacement (strumming or picking) of the string).

Test 2: Pitch Detection with GT Temporal Intervals

We evaluate the pitch detection part of our method, given the temporal intervals of the notes. The input is the set of string-pixel signals, $\{q_i^s(t)\}$ for each string s, the set of possible frequencies of s, $F_{string}(s)$, and a set of temporal note segmentations $\{\tau_j^s\}$. In this test we wish to focus merely on the pitch detection, hence we used GT temporal segmentation as input. For evaluation, we used the ground truth pitch

	Acoustic			**Classic**			**Electric**		
String	Onset TP (%)	Onset FP	Offset (%)	Onset TP (%)	Onset FP	Offset (%)	Onset TP (%)	Onset FP	Offset (%)
E(1)	51 (98%)	12	37 (73%)	47 (90%)	25	28 (60%)	52 (100%)	2	42 (81%)
A(2)	49 (94%)	12	35 (71%)	45 (87%)	27	21 (46%)	52 (100%)	2	49 (94%)
D(3)	37 (71%)	21	15 (41%)	30 (58%)	19	18 (55%)	48 (92%)	7	39 (81%)
G(4)	39 (75%)	43	13 (33%)	39 (75%)	33	28 (72%)	50 (96%)	17	26 (52%)

Table 5.1: Classic, acoustic and electric guitars onset and offset detection results out of 52 notes played per string on each instrument. True-positive (TP) detections and their respective percent out of the played notes, false-positive (FP) detections and offsets detection and the respective percent of the TP detection.

	Bass		
String	Onset TP(%)	Onset FP	Offset (%)
E(1)	60 (100%)	2	58 (97%)
A(2)	57 (95%)	3	45 (79%)
D(3)	57 (95%)	27	45 (79%)
G(4)	51 (85%)	20	39 (76%)

Table 5.2: Bass guitar onset and offset detection results out of 60 notes played per string. True-positive (TP) detections and their respective percent of the played notes, false-positive (FP) detections and offsets detection and the respective percent of the TP detection.

according to the known played notes.

Overall, 66% of the notes' pitches were detected correctly. We present the percent of incorrectly detected pitches. The errors per string and per fret are summarized in Tables 5.5 and 5.6, respectively. The results per instruments are shown in Table 5.7.

We classify these the errors into the following seven categories:

- Indistinguishable Notes (IN): The detected note is indistinguishable from the GT note (see Section 4.2.1 and Figure 4.6).
- Notes with f_0 in Noise Range (N_{f_0}):
- Notes with h_2 in noise Range (N_{h_2}):
- Notes with both f_0 and h_2 in Noise Range (N_{h_2}):
- Semitone Error (SE): The detected note is one semitone away up or down from GT.
- Harmonic Error (HE): The detected note is a harmonic of the GT one, or the GT is harmonic of the detected note.
- Others: all other errors.

Fret	Err. St. 1	Err. St. 2	Err. St. 3	Err. St. 4	Total Err.	% Total Err.
0	0	1	4	1	6	9%
1	0	0	0	1	1	2%
2	0	0	0	5	5	8%
3	0	0	0	7	7	11%
4	0	1	1	2	4	6%
5	0	0	1	5	6	9%
6	0	0	2	1	3	5%
7	0	0	4	2	6	9%
8	0	2	6	0	8	12.5%
9	0	1	11	2	14	22%
10	3	0	6	1	10	16%
11	1	2	4	4	11	17%
12	2	4	5	2	13	20%
Total Err.	6	11	44	33	94	**11%**
% Total Err.	3%	5%	21%	16%	**11%**	

Table 5.3: Summary of the errors of the temporal note segmentation (true-positive) per string and fret. The last two columns and rows show the total error per each fret and string respectively and the corresponding percent from the notes played per fret / string.

String	**Onset TP(%)**	**Offset (%)**
E(1)	210 (97%)	165 (79%)
A(2)	203 (94%)	150 (74%)
D(3)	172 (80%)	117 (68%)
G(4)	179 (83%)	105 (59%)

Table 5.4: Summary of all instruments temporal segmentation results, per string. The correctly detected notes (TP) are shown with their respective percent of the played notes. The correctly detected offsets are shown with their respective percent of the notes that their onsets were correctly.

We divide the first six error types to two sets. The first, $S_{expected} = HE \cup SE \cup IN$ of expected errors caused by several reasons: notes that are a semitone apart from one another and thus very close in frequency components(SE), notes that are octave apart thus share frequency components (HE) and indistinguishable notes (IN). The second, $S_{noise\_range} = N_{f_0}, N_{h_2}, (N_{f0}\ \&\ N_{h2})$, which are notes that are imperfectly seen by our method, as one of their $f_v(f_0)$, $f_v(h_2)$ or both are in the noise range. Note that HE, SE, IN are pairwise disjoint, as are $N_{f_0}, N_{h_2}, (N_{f0}\ \&\ N_{h2})$. However, $S_{expected}$ and $S_{noiserange}$ are not disjoint. For example, B2 has a fundamental frequency of (123.5 Hz), which is indistinguishable from B♭2 ($f_0 = (116.5\,\text{Hz})$) and also a note with h_2 in the noise range.

Tables 5.5 and 5.6 summarize the results per string and fret, respectively. Note that 13% of the 34% errors was in notes that has at least one harmonic (out of the first two) in the noise range. Harmonic errors account for only 2% of the 34%. This is expected, as those errors are applicable only for 2 (out of 13) frets for the classic,

Err. / Str.	Total Err.	$S_{expected}$			S_{noise_range}		
		HE	SE	IN	N_{f0}	N_{h2}	N_{f0} & N_{h2}
E(1)	70 (32%)	10 (7%)	0	12 (6%)	0	13 (6%)	0
A(2)	54 (25%)	0	6 (3%)	15 (7%)	5 (2%)	19 (9%)	0
D(3)	86 (40%)	4 (2%)	17 (8%)	4 (2%)	12 (6%)	4 (2%)	24(11%)
G(4)	82 (38%)	0	27 (12.5%)	4 (2%)	11 (5%)	4 (2%)	24 (11%)
Total	292 (34%)	14 (2%)	50 (6%)	35 (4%)	28(3%)	40 (5%)	48 (6%)

Table 5.5: Pitch detection errors given GT temporal intervals, per string. The first column summarize the total errors of each string and the respective percent from the total number of played notes, 216 (per string). The other columns present the number of errors for each type and set defined above, and the respective percent from the total played notes.

Err. / Fret	Total Err.	$S_{expected}$			S_{noise_range}		
		HE	SE	IN	N_{f0}	N_{h2}	N_{f0} & N_{h2}
0	4 (6%)	1 (2%)	0	0	0	3 (5%)	0
1	0	0	0	0	0	0	0
2	23 (36%)	0	0	12 (19%)	8 (12.5%)	12 (19%)	0
3	23 (36%)	0	0	0	0	0	12 (19%)
4	21 (33%)	0	7 (11%)	4 (6%)	0	4 (6%)	12 (19%)
5	11 (17%)	0	2 (3%)	0	0	1 (2%)	0
6	11 (17%)	0	3 (5%)	0	0	0	0
7	28 (44%)	0	2 (3%)	12 (19%)	12 (19%)	12 (19%)	0
8	27 (42%)	0	1 (2%)	0	0	0	12 (19%)
9	34 (53%)	0	10 (16%)	4 (6%)	0	4 (6%)	12 (19%)
10	34 (53%)	0	14 (22%)	0	0	0	0
11	27 (42%)	0	6 (9%)	0	0	0	0
12	37 (58%)	13 (20%)	1 (2%)	0	5 (8%)	0	0
13*	2(12.5%)	0	1 (6%)	0	0	0	0
14*	10 (62.5%)	0	3 (19%)	3 (19%)	3 (19%)	4 (25%)	0
Total	292 (34%)	14 (2%)	50 (6%)	35 (4%)	28(3%)	40 (5%)	48 (6%)

Table 5.6: Pitch detection errors given GT temporal intervals, per fret. Note, *frets 13 and 14 are only applicable for the bass guitar. The first column summarize the total errors of each fret and the respective percent from the total number of played notes, 64 for frets 0-12 and 16 for frets 13-14. The other columns present the number of errors for each type and set defined above, and the respective percent from the total played notes.

acoustic and electric guitar (0, 12) and for 6 (out of 15) frets for the bass guitar (0,1,2,12,13,14). 6% of the 34% are notes that were identified as a semitone above of below the GT pitch.

Analyzing the results per fret uncovers the relation between the errors caused in frets with imperfectly seen notes and the total amount of errors, as is shown in Figure 5.2. Up to the 8th fret, a steady ratio between the total errors and those of the imperfectly seen frets is kept. Starting from the 8th fret, the number of errors increases in frets with no noise issues. This is expected since, as explained above, higher frets notes are harder to detect and analyze. Moreover, this is consistent with the findings in Test 1, where the higher frets showed inferior results. Failure to temporally segment a note will usually cause failures in the pitch detection as well,

since occasionally a temporal segmentation errors is a result of an undetected note and not merely failure in detecting the onset within the desired threshold.

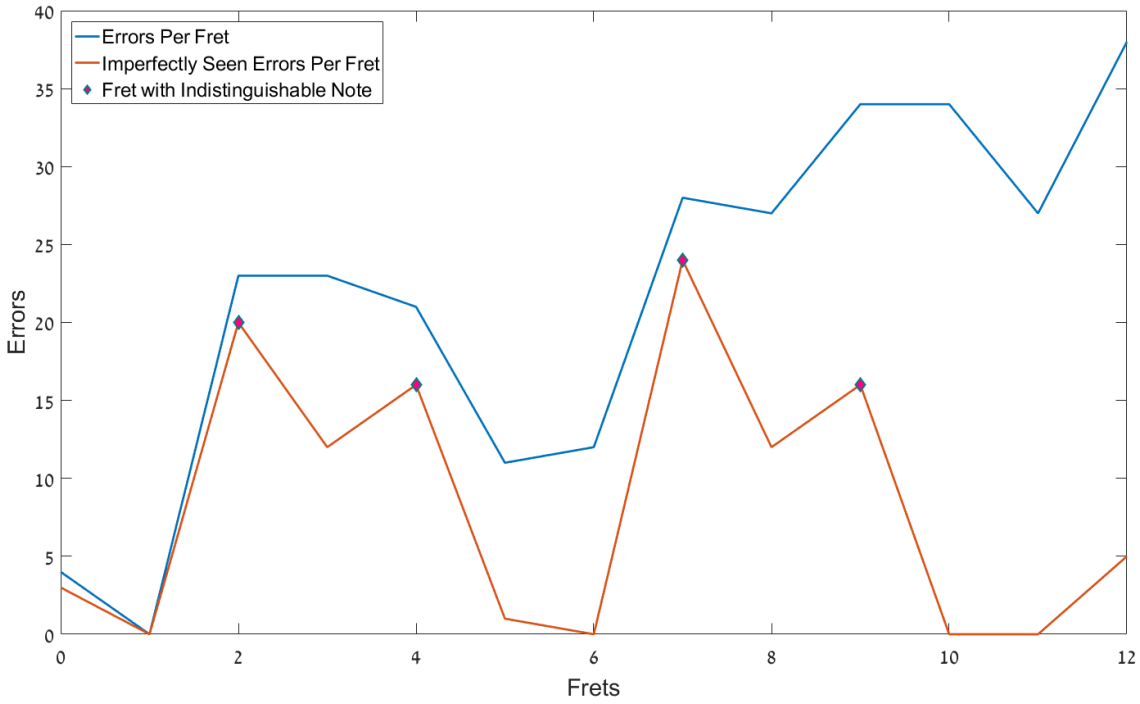


Figure 5.2: The errors using GT temporal intervals (blue), and the errors in imperfectly seen notes (orange), on each fret. The peaks (diamonds) correspond to frets that have indistinguishable notes, in which errors are certain.

Observing the results per string shows that the two lower strings are again performing better than the two higher ones. However, the best results was listed for the second lowest string (A), rather than the lowest one (E). This can be explained by the sparse fundamentals of the notes on the A string, as opposed to the relatively dense fundamentals of the E string (see Figure 4.5). Clearly, this property affects the pitch detection and not the temporal segmentation, thus the E string presented the best results in the temporal note segmentation stage.

The results per instrument are presented in Table 5.7. As expected, the bass guitar results are superior to all other guitar, due to the strings thickness and length, and since less notes are imperfectly seen.

Instrument	% **Success**
Bass	86
Acoustic	64
Classic	56
Electric	55
Total	**66**

Table 5.7: The pitch detection success percentage using GT temporal intervals, per instrument, counting the number of correctly detected pitches of the total notes played on each instrument - 240 for the bass guitar and 208 for all other guitars.

Err. / String	Matched	Total Err.	$S_{expected}$			S_{noise_range}		
			HE	SE	IN	N_{f0}	N_{h2}	N_{f0} & N_{h2}
E(1)	210 (97%)	68 (32%)	10 (5%)	0	12 (6%)	0	13 (6%)	0
A(2)	206 (95%)	55 (27%)	0	5 (2%)	13 (6%)	6 (3%)	18 (9%)	0
D(3)	196 (91%)	74 (38%)	5 (3%)	5 (3%)	4 (2%)	12 (6%)	4 (2%)	13 (7%)
G(4)	197 (91%)	74 (38%)	0	24 (12%)	4 (2%)	7 (4%)	5 (3%)	17 (10%)
Total	809 (94%)	271 (33%)	15 (2%)	33 (4%)	33 (4%)	25 (3%)	40 (5%)	30 (4%)

Table 5.8: Pitch detection errors given automatically segmented temporal intervals, per string. The first column is the number of intervals that were matched with a GT interval and corresponding percent of the total number of the GT notes, 216 (per string). The second column presents the errors per string in pitch detection only for the matched detected notes. The other columns present the number of errors for each type and set defined above, and the respective percent from the total played notes.

Test 3: End-to-end

In this test we apply our algorithm to compute the pitch of all temporal intervals segmented by our method. This allows evaluation of the entire system. We first evaluate only the pitch detection, and then frame-by-frame errors.

Evaluation of Pitch Detection We now evaluate the pitch detection given the temporal intervals calculated by our method. The pitch errors are computed only for detected temporal segments that overlap the GT. Results per string and per fret are given in Tables 5.8 and 5.9 respectively. The first column is the number of intervals that where matched with a GT interval and corresponding percent of the total number of the GT notes. The second column presents the errors in pitch detection only for the matched detected notes. We also present, as in Tables 5.5 and 5.6, the classification of the errors, and the overall results per instrument, similarly to the Table 5.7.

Since notes are matched by choosing the note with largest interval overlap to GT interval, notes that are not matched suggest that no note (nor noise) was detected in the GT temporal interval. The better results of matching in the two lower strings in consistent with the results in Test 1.

Roughly, the pitch detection results are consistent with the results of Test 2. This suggests that the note with closest temporally onset is usually also the one with the most temporal overlap to GT.

Similarly to the results in the previous section, more errors occur in higher frets, and from the 8th fret an on, less of these errors occur due to noise issues. Again, better pitch detection is obtained the string (A) showing better results than the lowest one (E) as previously in Test 2.

The results per instrument are presented in Table 5.10. Again we see the bass guitar superiority.

Frame-by-frame Evalutation We use frame-by-frame evaluation of the results compared to ground truth. We compare the output data and the ground truth in each frame. The algorithm output is assumed to be correct if in a given frame the GT and the output shows the same frequency (or no frequency, e.g. no note is played

Fret \ Err.	Matched	Total Err.	$S_{expected}$			S_{noise_range}		
			HE	SE	IN	N_{f0}	N_{h2}	N_{f0} & N_{h2}
0	64 (100%)	4 (6%)	1 (2%)	0	0	0	3 (5%)	0
1	64 (100%)	1 (2%)	0	0	0	0	0	0
2	62 (97%)	21 (34%)	0	1 (2%)	11 (18%)	6 (10%)	12 (19%)	0
3	59 (92%)	18 (31%)	0	1(2%)	0	0	0	7 (12%))
4	62 (97%)	17 (27%)	0	6 (10%)	4 (6%)	0	4 (6%)	10 (16%)
5	61 (95%)	9 (15%)	0	1 (2%)	0	0	1 (2%)	0
6	64 (100%)	12 (19%)	0	1 (2%)	0	0	0	0
7	64 (100%)	27 (42%)	0	1 (2%)	12 (19%)	12 (19%)	12 (19%)	0)
8	60 (94%)	23 (39%)	0	0	0	0	0	8 (13%)
9	57 (89%)	30 (53%)	0	3 (5%)	4 (7%)	0	4 (7%)	5 (9%)
10	59 (92%)	39 (66%)	0	17 (29%)	0	0	0	0
11	54 (84%)	22 (41%)	0	0	0	0	1(2%)	0
12	53 (83%)	38 (72%)	14 (26%)	1 (2%)	0	6 (11%)	1(2%)	0
13*	15 (94%)	3(20%)	0	1 (7%)	0	0	0	0
14*	11 (69%)	7 (64%)	0	1 (9%)	2 (18%)	1 (9%)	2 (18%)	0
Total	809 (94%)	271 (33%)	15 (2%)	34 (4%)	33 (4%)	25 (3%)	40 (5%)	30 (4%)

Table 5.9: Pitch detection errors given automatically segmented temporal intervals, per fret. Note, *frets 13 and 14 are only applicable for the bass guitar. The first column is the number of intervals that where matched with a GT interval and corresponding percent of the total number of the GT notes (64 for frets 0-12 and 16 for frets 13-14). The second column presents the errors per fret in pitch detection only for the matched detected notes. The other columns present the number of errors for each type and set defined above, and the respective percent from the total played notes.

Instrument	% **Matched**	% **Success**
Bass	96	88
Acoustic	88	65
Classic	93	55
Electric	98	55
Total	**94**	**67**

Table 5.10: Total pitch detection success percentage using automatically obtained temporal intervals, per instrument. The first column is the percent of matched intervals from the notes played - 240 for the bass guitar and 208 for all other guitars. The second column is the percent of the successful pitch detection from the total notes matched.

in the frame). This allows us to evaluate the system for both frequency and temporal information combined. Using this evaluation allows us to largely ignore cases that are counted as errors in previous test, but reflect some success of the methods in some aspects. Figure 5.4 shows some of those errors. The first error (1) is a case were a note is split, where the first, temporally closer to GT, detected note was detected with the wrong pitch, and the second was detected with the correct pitch. In this case the first detected note will be accounted for in the temporal segmentation detection and the other will be considered a false-positive. In the auto-computed pitch detection, the same detected note will be considered and since its pitch was

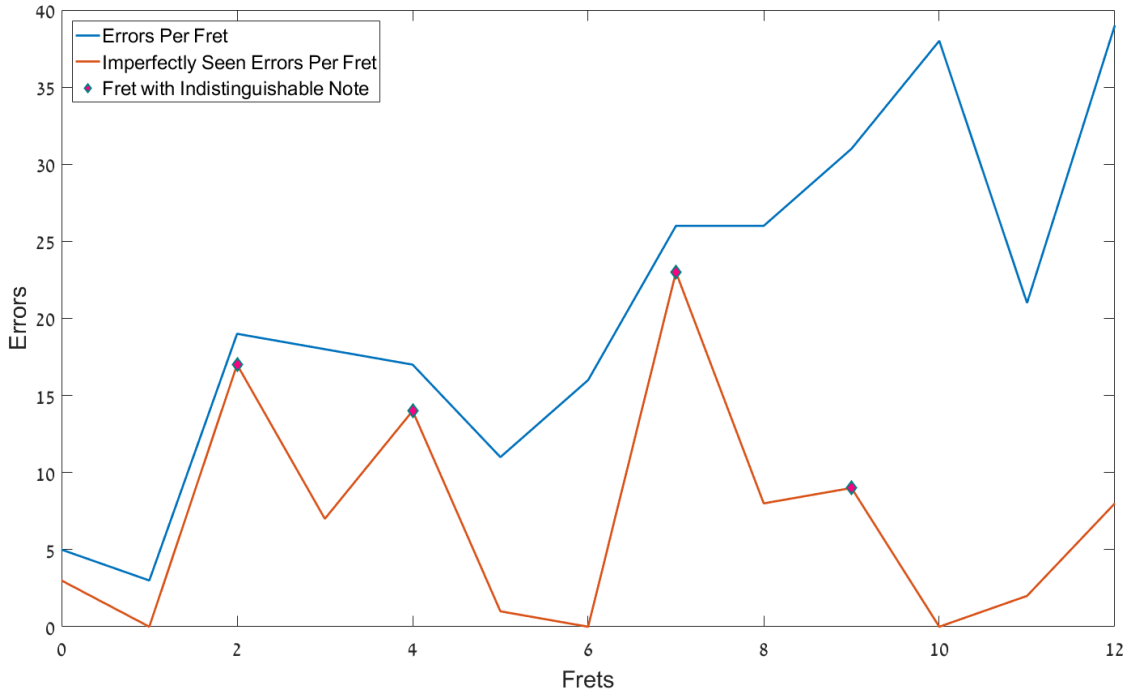


Figure 5.3: The errors using automatically obtained temporal intervals (blue), and the errors in imperfectly seen notes (orange), on each fret. The peaks (diamonds) correspond to frets that have indistinguishable notes, in which errors are certain.

detected wrongly, the pitch test will record an error. However, using the frame-by-frame evaluation, the overlapping frame of the second detected notes and the ground truth will be considered as success. In the second error (2), a splitting occurred and both detected notes was detected with the right pitch. For the temporal interval test, the second detected note will be considered as false-positive. For the pitch detection using automatic intervals, the first detected note will be considered and the other ignored, and the offset detection is a failure. Frame-by-frame evaluation will consider the entire note as successfully obtained, except from the gap between to detected notes. The third and fourth errors (3,4) are cases where the detected onset is not within the tested threshold, either too late (3) or too early (4). For the temporal segmentation test, the detected notes in both cases will be considered false positives. For the auto-segmented pitch detection test, the detected notes will be matched and considered true-positive since the pitch detected is correct. The frame-by-frame evaluation will consider all area that overlaps both the GT and detected note as success and any other area as error.

This evaluations is roughly similar to f_0-estimation problem (see MIREX 2016 [35]) that involves detecting all active fundamentals in a given time frame. In our case, the problem diminishes to a single f_0-estimation, as we consider a single string each time, that can only produce a single note simultaneously. This evaluation is extremely robust, does not make any assumptions on the data (such as thresholds), it is not dependant of sequential steps and provides good estimation of the system performance. Note that we do not allow tolerance in the note onsets and offsets here. Furthermore, this evaluation disregards the nature on an error. For example, undetecting a note temporally and a correctly temporally detected note with an incorrectly detected pitch will both be considered as a frame with an error. This ex-

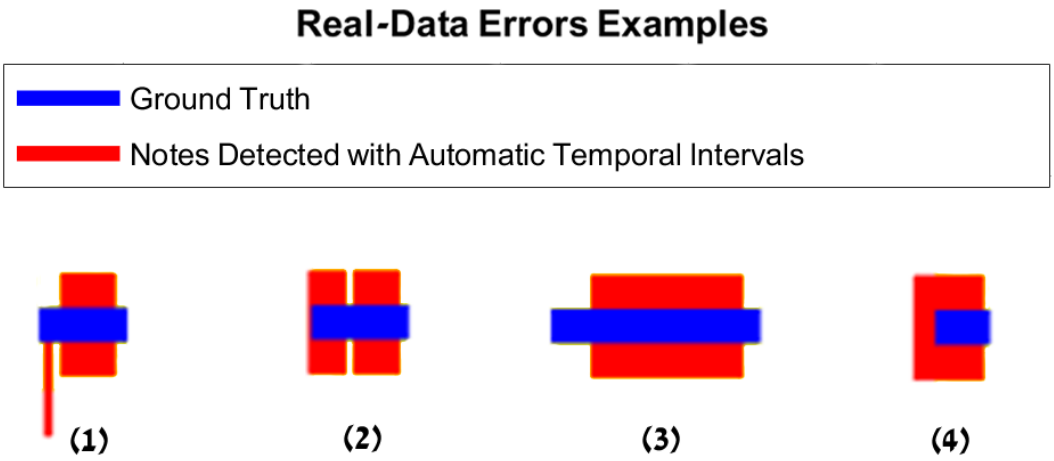


Figure 5.4: Errors examples obtained by testing our method on real-data. **(1)** A split note, where the first detected note is detected with the wrong pitch and the second with the correct one. **(2)** A split note, where both detected notes are detected with the correct pitch. **(3),(4)** A case where the detected onset is not within the tested threshold, either too late (3) or too early (4).

plains the slight variation is results per string and instrument comparing to previous tests.

Table 5.11 summarizes this evaluation method. Overall, 68% of the frames were correctly detected. As was observes before, when considering the entire data on all guitars, lower strings yield better results. However, this observation does not hold when considering each instrument separately, which encourage us to collect more data to get more conclusive results. In this test, as in was shown in the other tests, the bass guitar outperforms other guitars.

String	Bass	Acoustic	Classic	Electric	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
Total	79	64	63	68	**68**

Table 5.11: A frame-by-frame evaluation presenting the percent of frames that were successfully detected from the total frames in the video, calculated per string and instrument.

Test 4: String-pixels Selection

Here we test the string-pixels that yields the best results. Three sets of string-pixels were considered; string-pixels that were extracted using high fret (6th, 7th or 8th) note calibration, open string note calibration, and manual marked pixels on the string. Our method was then applied on each video of the bass guitar. Several of

(a) High-fret string-pixels (b) Open string string-pixels (c) Manual string-pixels

Figure 5.5: The string-pixels obtained using three different methods: **(a)** String-pixels obtained by using the temporal-spectral algorithm with a high-fret note (6, 7 or 8). **(b)** String-pixels obtained by using the temporal-spectral algorithm with an open string note. **(c)** String-pixels that were marked manually upon the string.

the mentioned above measurements were used to determine the best selection of string pixels. Those include frame-by-frame evaluation (as in Test 3 - frame-by-frame evaluation), onsets f-measure (as in Test 1) and pitch detection using GT temporal intervals (as in Test 2). Results are summarized in Table 5.12. The set of string-pixels obtained by our automatic string detection using high fret note calibration surpassed other methods (our performed equally well) in each of the tested measurements.

	Frame-by-Frame	Onsets F-measure	PD with GT int.
Manual	**79 %**	0.85	85 %
Auto. Open-String	73 %	0.85	84 %
Auto. High Fret	**79 %**	**0.87**	**86 %**

Table 5.12: Different string-pixels detection methods and their performances, evaluated by the different evaluation methods.

Test 5: Polyphony Demonstration (Chords Playing)

In this section we present a demonstration of a possible application of our method. In this demonstration we show a polyphony example of an acoustic guitar playing 5 chords: Cmaj (open) - Gmaj (open) - Fmaj (Barred) - Dmaj (open) - Gmaj (open). This means that at the same time, 5 or 6 strings are playing together. In this case, we assume that the guitarist is playing in "chord mode", meaning only chords are played and considered. Appropriately, we add some post-processing to the output of our algorithm:

- Only temporal intervals that overlap in at least 3 strings are kept.
- Temporal intervals lasting less than 48 frames (200 ms.) are discarded.

The results are shown in Figure 5.6. Note, that in this case the offset is without significance. The time-overlapping temporal segments where shortened to the shortest interval in the chord to provide better visualisation. In practice, when playing

chords, a letter representing the chord will appear in the musical sheet (for example, Am7), and no offset notation is used.

This demonstration points out two important strengths of our method. First, polyphony does not affect the signal extracted from each string separately. This validates our approach and tests. Secondly, since every string is analyzed separately, an error on one or more strings may not affect the correctness in identifying a chord. It remains for future work to develop a chord-oriented system that could analyze the played notes as chords.

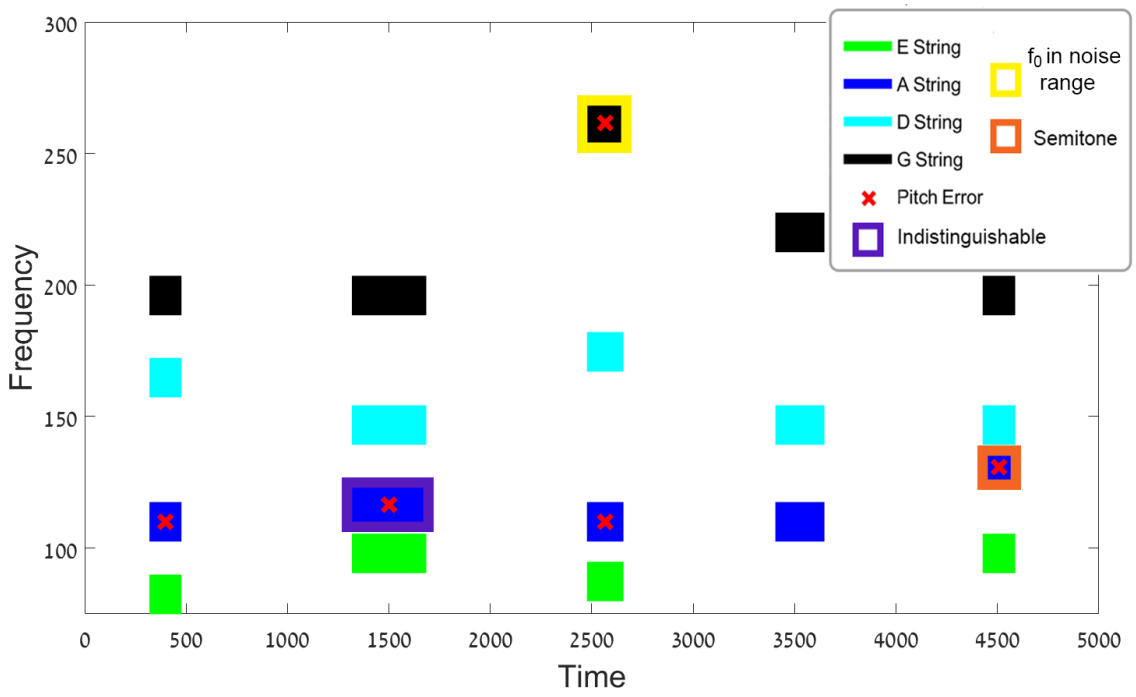


Figure 5.6: The method's output in "chord mode", for a video capturing the playing of the chords Cmaj - Gmaj - Fmaj - Dmaj - Gmaj, as explained in Section 5.3. The X's mark notes that were detected with the wrong pitch. The rectangles around the detected notes indicate the type of error, if applicable. In this case the offsets are disregarded, since generally in chords notation no offsets are mentioned.

Chapter 6

Discussion and Future Work

In this thesis we present the first steps toward solving Note Tracking for various guitars, using a video capturing the vibration of the strings. We clarify the challenges involved and suggest algorithms for handling them. We present our algorithm which includes thee main steps – string-pixels detection, temporal note segmentation, and pitch detection. Our method for detecting string-pixels is based on the strings vibrations rather than using classic detection techniques that are based on geometry. The pitch detection algorithm uses priors on the possible notes played on each string as well as their expected visible aliasing and harmonics. The temporal segmentation uses multiple signals obtained from a single video to attain robustness. In addition, we present a set of methods for evaluating these algorithms. We next discuss our results, and suggest how our method can be further extended in a future work.

Discussion

As a rule, a successful capturing of signals of the played note depends on two main factors. The first is the camera's frame rate. The frame rate makes some notes partially or completely obscured by noise, which induces errors. Additionally, some notes are ambiguous in relation to others. For the standard guitars other than the bass guitar, the method of harmonics scoring had limited success in detecting notes that their fundamental frequency was in the noise range. In particular, the notes B♭3 and B3 that both their 1st and 2nd harmonics are in the noise range were undetectable by our method.

Second, the physical characteristics of the string can also influence the success rate of our method. Notes that are produced by playing a lower fret, are usually easier to detect, since the longer the string is, the bigger its vibrating amplitude. Additionally, lower strings are thicker, which seldom makes the intensity change signal captured stronger and less noisy.

The bass guitar results are better than the other guitars throughout all tests. This is expected, due to the instrument characteristics as mentioned above: the bass guitar has longer strings and thicker strings and the bass guitar produces lower frequencies notes, thus a note's fundamental frequency is always in not in the noise range.

Since our method is unique and incomparable to other NT methods, we presented a diverse variety of testing techniques to evaluate our method. Most results are roughly in the same scale as audio, and although our method is limited in some

aspects such as frets higher than the 12th (in standard guitars) and the invisibility of the two highest strings, it offers an innovative approach - analyzing a complex polyphonic signal by transforming it to multiple monophonic signals.

Future Work

Extracting notes from a vibrating string for the task of note tracking or other music information retrieval tasks, is a brand new field. Thus, the options for future work are innumerous.

As commercial cameras keep improving, higher frame rates and higher resolution are expected to be available at reasonable price. We expect that an increase in frame rate will massively improve our performances. Not only will a higher frame rate solve ambiguities and remove notes from noisy range, it will provide a better time resolution. Additionally, errors that are unaffected by the noise range are also expected to be reduces, as a higher frame rate expands the unaliased frequency range, thus making notes more distinguishable from one another. Moreover, a better resolution camera or one with zooming abilities can also contribute to better results, since the extracted signal can be stronger and clearer.

A clear aspiration of future work is to relinquish the need of a mounted camera. The vision is to capture a musician playing his instrument from a far, non-intrusive camera, and to be able to extract the played notes. Simple tacking of the instruments or strings might not suffice, as our method extracts the signal by measuring the change in intensities a string generates when vibrating, and a large-scale movements of the guitar may interfere with this signal. Nonetheless, preliminary tests show that a video of a guitar player sitting fairy still in front of a stationary camera, can be used to extract such signal and the note played. Furthermore, there are additional changes when using an unmounted camera; the distance from the instrument may cause the signal to be fairly weak, and it may be harder to distinguish between adjacent strings. This could be partially solved by using a more advanced camera.

As our retrieved signal is very similar to an audio retrieved signal, future work can also include implementing the great work done on audio signals, on our visually obtained signal. Time domain, as well as frequency domain methods for audio signals such as filter banks and cepstrum, can yield improvements if operated on the visual signal.

Moreover, a hybrid approach, combining both visual signal and audio signal, should be attempted. Some methods combining both audio and video were previously addressed, but only for left hand tracking methods that doesn't obtain the signal from the vibrating string. As our visual signal and the audio signal represent the same physical event, more information could be gathered on the played note, as opposed to a visual signal describing the movement of the instrument or the player. Each of the signals has its own strengths and weaknesses, e.g., audio has a higher resolution that evades low frame rate ambiguities but captures a polyphonic signal that presents other ambiguities and that is harder to process. Infusion of both signals can create a clearer, easier to process signal.

Future work should also include some knowledge on the instrument and the manner it is played. For example, tracking of the left hand and positioning it on the guitar, can eliminate ambiguities for the currently played note, as the hand should be placed roughly around the pressed fret. Another way to use instruments

or musical information is to estimate a likely trajectory of the left hand, which can also eliminate unlikely played notes and improve pitch detection accuracy. Although one of our method strengths (compared to other visual methods) is the ability to detect all kinds of melodies and non-trivial chords, providing a pre-set of chords, even a vast one, could eliminate ambiguities in pitch detection and also overcome the invisibility of some notes, as a chord with some detected notes could suffice to identify it.

More string instruments should be tested our method, mainly bowed string instruments, as the string vibration is slightly different.

To conclude, we believe that the proposed solution to the note tracking problem, can be a first step toward a more reliable NT with better results, as we present a novel approach to obtain the musical information. Combining our method with other existing audio or vision based method may be present a big leap in the NT results and other MIR applications.

Appendix A

Fret \ String	E	A	D	G
0	82.41	110	146.8	196
1	87.31	116.5	155.6	207.7
2	92.5	123.5	164.8	220
3	98	130.8	174.6	233.1
4	103.8	138.6	185	246.9
5	110	146.8	196	261.6
6	116.5	155.6	207.7	277.2
7	123.5	164.8	220	293.7
8	130.8	174.6	233.1	311.1
9	138.6	185	246.9	329.6
10	146.8	196	261.6	349.2
11	155.6	207.7	277.2	370
12	164.8	220	293.7	392

Figure A.1: Classic, acoustic and electric guitars fundamental frequencies. Yellow markings indicate notes that their h_2 are in the noise range, red markings indicates notes that their f_0 are in the noise range, mixed red and yellow markings indicates notes that both their f_0 and h_2 are in the noise range.

Fret \ String	E	A	D	G
0	41.2	55	73.42	98.00
1	43.65	58.27	77.78	103.8
2	46.25	61.74	82.41	110
3	49	65.41	87.31	116.5
4	51.91	69.3	92.5	123.5
5	55	73.42	98	130.8
6	58.27	77.78	103.8	138.6
7	61.74	82.41	110	146.8
8	65.41	87.31	116.5	155.6
9	69.30	92.5	123.5	164.8
10	73.42	98	130.8	174.6
11	77.78	103.8	138.6	185
12	82.41	110	146.8	196
13	87.31	116.5	155.6	207.7
14	92.5	123.5	164.8	220

Figure A.2: Bass Guitar fundamental frequencies. Yellow markings indicate notes that their h_2 are in the noise range, and red markings indicated notes that their f_0 are in the noise range.

Appendix B

Threshold = 12			**Threshold = 24**			**Threshold = 36**		
Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
0.36	0.43	0.39	0.56	0.68	0.61	0.64	0.79	0.71

Table B.1: Precision, recall and F-measure evaluations for onset detection for all instruments, using different thresholds.

Bass guitar

	Threshold = 12		Threshold = 24		Threshold = 36	
String	**TP (%)**	**FP**	**TP (%)**	**FP**	**TP (%)**	**FP**
E (1)	37 (62%)	25	49 (82%)	13	59 (98%)	3
A (2)	48 (80%)	12	55 (92%)	5	56 (93%)	4
D (3)	20 (33%)	64	38 (63%)	46	48 (80%)	36
G (4)	20 (33%)	51	33 (55%)	38	41 (68%)	30
Total	125 (52%)	152	175 (73%)	102	204 (85%)	73

Electric guitar

	Threshold = 12		Threshold = 24		Threshold = 36	
String	**TP (%)**	**FP**	**TP (%)**	**FP**	**TP (%)**	**FP**
E (1)	25 (48%)	29	46 (88%)	8	50 (96%)	4
A (2)	25 (48%)	29	43 (83%)	11	49 (94%)	5
D (3)	25 (48%)	30	44 (85%)	11	48 (92%)	7
G (4)	29 (56%)	38	41 (79%)	26	47 (90%)	20
Total	104 (50%)	126	174 (84%)	56	194 (93%)	36

Acoustic guitar

	Threshold = 12		Threshold = 24		Threshold = 36	
String	**TP (%)**	**FP**	**TP (%)**	**FP**	**TP (%)**	**FP**
E (1)	26 (50%)	37	47 (71%)	16	50 (96%)	13
A (2)	26 (50%)	35	39 (75%)	22	43 (83%)	18
D (3)	15 (29%)	43	25 (48%)	33	30 (58%)	28
G (4)	11 (21%)	71	21 (40%)	61	24 (46%)	58
Total	78 (37.5%)	186	132 (63%)	132	147 (71%)	117

Classic guitar

	Threshold = 12		Threshold = 24		Threshold = 36	
String	**TP (%)**	**FP**	**TP (%)**	**FP**	**TP (%)**	**FP**
E (1)	15 (29%)	57	28 (54%)	44	38 (73%)	34
A (2)	20 (38%)	52	30 (58%)	42	33 (63%)	39
D (3)	14 (27%)	35	19 (37%)	30	24 (46%)	25
G (4)	13 (25%)	59	18 (35%)	54	26 (50%)	46
Total	62 (30%)	203	95 (46%)	170	121 (58%)	144

Table B.2: Onset detection results for all instruments, using different thresholds. A total of 240 notes are played on the bass guitar (60 per string) and 208 on each of the other guitars (52 per string).

Bibliography

[1] Z. Wang and J. Ohya, "Tracking the guitarist's fingers as well as recognizing pressed chords from a video sequence," *Electronic Imaging*, vol. 2016, no. 15, pp. 1–6, 2016.

[2] 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*. Springer, 2007, pp. 625–638.

[3] A.-M. Burns and M. M. Wanderley, "Visual methods for the retrieval of guitarist fingering," in *Proceedings of the 2006 conference on New interfaces for musical expression*. IRCAM—Centre Pompidou, 2006, pp. 196–199.

[4] 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*, 2014.

[5] 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)*, vol. 31, no. 4, 2012.

[6] M. Rubinstein, "Analysis and visualization of temporal variations in video," Ph.D. dissertation, Massachusetts Institute of Technology, Feb 2014.

[7] H. Nyquist, "Certain topics in telegraph transmission theory," *Transactions of the American Institute of Electrical Engineers*, vol. 47, no. 2, pp. 617–644, 1928.

[8] C. E. Shannon, "A mathematical theory of communication, part i, part ii," *Bell Syst. Tech. J.*, vol. 27, pp. 623–656, 1948.

[9] H.-Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand, and W. Freeman, "Eulerian video magnification for revealing subtle changes in the world," *ACM Transactions on Graphics*, 2012.

[10] M. Rubinstein *et al.*, "Analysis and visualization of temporal variations in video," Ph.D. dissertation, Massachusetts Institute of Technology, 2014.

[11] A. Owens, P. Isola, J. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman, "Visually indicated sounds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2405–2413.

[12] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: challenges and future directions," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 407–434, 2013.

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

[14] B. 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 (SCOPES), 2016 International Conference on*. IEEE, 2016, pp. 140–152.

[15] T. F. Tavares, J. G. A. Barbedo, R. Attux, and A. Lopes, "Survey on automatic transcription of music," *Journal of the Brazilian Computer Society*, vol. 19, no. 4, pp. 589–604, 2013.

[16] A. P. Klapuri, "Automatic music transcription as we know it today," *Journal of New Music Research*, vol. 33, no. 3, pp. 269–282, 2004.

[17] 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*, vol. 20, no. 3, pp. 915–921, 2012.

[18] C. Dittmar, A. Männchen, and J. Abeber, "Real-time guitar string detection for music education software," in *Image Analysis for Multimedia Interactive Services (WIAMIS), 2013 14th International Workshop on*. IEEE, 2013, pp. 1–4.

[19] J. Abeßer, "Automatic string detection for bass guitar and electric guitar," in *International Symposium on Computer Music Modeling and Retrieval*. Springer, 2012, pp. 333–352.

[20] 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 *DAFx*, 2014, pp. 219–226.

[21] M. Paleari, B. Huet, A. Schutz, and D. Slock, "A multimodal approach to music transcription," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. IEEE, 2008, pp. 93–96.

[22] M. Paleari, B. Huet, A. Schutz, D. Slock, N. J. Kern, G. Guarascio, and B. Mérialdo, "Audio-visual guitar transcription," in *Jamboree 2008: Workshop By and For KSpace PhD Students, July, 25 2008, Paris, France*, 2008.

[23] G. Quested, R. Boyle, and K. Ng, "Polyphonic note tracking using multimodal retrieval of musical events." in *ICMC*, 2008.

[24] M. Cheung and K. M. B. Lee, "A vision based autonomous music transcription software for guitarists," *University of Sydney*, 2006.

[25] J. Scarr and R. Green, "Retrieval of guitarist fingering information using computer vision," in *Image and Vision Computing New Zealand (IVCNZ), 2010 25th International Conference of*. IEEE, 2010, pp. 1–7.

[26] 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*. ACM, 2007, pp. 521–524.

[27] A.-M. Burns and M. Wanderley, "Computer vision method for guitarist fingering retrieval," in *Proceedings of the Sound and Music Computing Conference*, 2006.

[28] A.-M. Burns, "Computer vision methods for guitarist left-hand fingering recognition," Ph.D. dissertation, McGill University, 2006.

[29] A. Hrybyk, "Combined audio and video analysis for guitar chord identification," Ph.D. dissertation, Drexel University, 2010.

[30] M. Cicconet, "The guitar as a human-computer interface," Ph.D. dissertation, D. Sc. Thesis. National Institute of Pure and Applied Mathematics. Rio de Janeiro, 2010.

[31] C. Kerdvibulvech and H. Saito, "Real-time guitar chord estimation by stereo cameras for supporting guitarists," in *Proceeding of 10th International Workshop on Advanced Image Technology (IWAIT 07)*, 2007, pp. 256–261.

[32] ——, "Vision-based detection of guitar players' fingertips without markers," in *Computer Graphics, Imaging and Visualisation, 2007. CGIV'07.* IEEE, 2007, pp. 419–428.

[33] "Music Information Retrieval Evaluation eXchange: audio onset detection," http://www.music-ir.org/mirex/wiki/2016:Audio_Onset_Detection.

[34] "Music Information Retrieval Evaluation eXchange 2016: audio onset detection results," http://nema.lis.illinois.edu/nema_out/mirex2016/results/aod/summary.html.

[35] "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.

תקציר

אחזור מידע מוזיקלי (Music Information retrieval) מהווה תחום מחקר חשוב במוזיקה, ומשלב את תחומי מדעי המחשב, עיבוד אותות, פיסיקה, פסיכולוגיה ועוד. אחת הבעיות החשובות ביותר תחום זה היא תעתיק אוטומטי של תווים מוזיקלים (Automatic Music Transcription), אשר כוללת כתיבת תוים עבור יצירות מוזיקליות להן לא קיימים תווים. לבעיה זו חשיבות רבה בתחום מחקר המוזיקה, אך מורכבותה גדולה ועבור מוזיקה פוליפונית ביצועיהם של אנשי מקצוע עולים על ביצועיהן של מערכות אוטומטיות. אנו מציגים גישה חדשה לפתרון בעיה חלקית של תעתיק אוטומטי של תווים מוזיקלים, המוגבלת למספר מרכיבים הכוללים את גובה הצליל ותזמונו, הנקראת מעקב אחר תווים (Note Tracking). אנו מתמקדים בעבודה זו בצלילים המנוגנים על גיטרות, כאשר הקלט הוא וידאו המצולם ללא קול על ידי מצלמה המורכבת על גבי כלי הנגינה. נשתמש בתנודת המיתרים בתור אות חזותי וננתח אותו בעזרת שיטות של עיבוד אותות וראיה ממוחשבת. הניתוח של האינפורמציה מכל מיתר בנפרד מאפשר את צמצום הסיבוכיות של הבעיה והפיכתה מבעיה של מוזיקה פוליפונית למספר בעיות של מוזיקה מונופונית. בנוסף, ניתן לנתח את השגיאות הצפויות בפתרון המוצע, בהנתן סוג כלי הנגינה, המיתר, וקצב הדגימה של המצלמה. הפתרון שלנו, שנבדק על 4 גיטרות שונות, יכול לתרום רבות לתחום המעקב אחר תווים.

עבודה זו בוצעה בהדרכתה של פרופ' יעל מוזס מבי"ס אפי ארזי למדעי המחשב, המרכז הבינתחומי, הרצליה.

בית ספר אפי ארזי
למדעי המחשב
הבינתחומי
הרצליה

המרכז הבינתחומי בהרצליה
בית-ספר אפי ארזי למדעי המחשב
התכנית לתואר שני (M.Sc.)

זיהוי אוטומטי של תום מוזיקליים לכלי מיתר על ידי ראיה ממוחשבת

מאת
שיר גולדשטיין

בהדרכתה של פרופ' יעל מוזס

פרויקט גמר, מוגש כחלק מהדרישות לשם קבלת תואר מוסמך M.Sc.,
בית ספר אפי ארזי למדעי המחשב, המרכז הבינתחומי הרצליה

ספטמבר 2017