Computer Vision Methods for Guitarist Left-Hand Fingering Recognition

Anne-Marie Burns

Input Devices and Music Interaction Lab
Schulich School of Music
McGill University
Montréal, Québec, Canada

January 2007

A thesis submitted to McGill University in partial fulfillment of the requirements of a degree of Master of Arts in Music Technology.

Abstract

This thesis presents a method to visually detect and recognize fingering gestures of the left hand of a guitarist. The choice of computer vision to perform that task is motivated by the absence of a satisfying method for realtime guitarist fingering detection. The development of this computer vision method follows preliminary manual and automated analyses of video recordings of a guitarist. These first analyses led to some important findings about the design methodology of such a system, namely the focus on the effective gesture, the consideration of the action of each individual finger, and a recognition system not relying on comparison against a knowledge-base of previously learned fingering positions. Motivated by these results, studies on three important aspects of a complete fingering system were conducted. One study was on realtime finger-localization, another on string and fret detection, and the last on movement segmentation. Finally, these concepts were integrated into a prototype and a system for left-hand fingering detection was developed. Such a data acquisition system for fingering retrieval has uses in music theory, music education, automatic music and accompaniment generation and physical modeling.

Sommaire

Cette thèse présente un système axé sur la vision informatique permettant la détection et l'identification du doigté d'un guitariste. Le choix de la vision informatique comme méthode de traitement est motivé par l'absence d'un système de détection du doigté en temps réel satisfaisant pour les musiciens. Le développement de cette méthode a été précédé d'analyses manuelles et informatisées d'extraits vidéos d'un guitariste. Ces analyses préliminaires ont permis d'établir les caractéristiques nécessaires au développement d'un prototype de reconnaissance visuelle du doigté : l'accent sur le mouvement effectif, la considération de l'action individuelle de chaque doigt et l'établissement d'un système de reconnaissance ne reposant pas sur la comparaison avec une base de connaissances. Motivées par ces résultats, trois études ont été conduites. Une étude porte sur la détection de la position du bout des doigts en temps réel. Une autre est sur la détection des cordes et des frettes dans l'image. Une dernière étude traite de la segmentation du mouvement. Les résultats de ces trois études ont ensuite été combinés pour le développement d'un prototype et la réalisation d'un système de détection du doigté. Un système d'acquisition de données sur le doigté comme celui-ci a des applications en théorie de la musique, en éducation, en génération automatique de musique et d'accompagnements et en modélisation physique d'instruments.

Acknowledgments

First I would like to thank Marcelo Wanderley, my supervisor, whose enthusiasm carried me throughout this Master's degree.

The study on finger localization was realized at the InfoMus Laboratory, D.I.S.T., Università degli studi di Genova, and has been partially supported by funding from the Quebec government (PBCSE), the Italian Ministry of Foreign Affairs, and by the EU 6 FP IST ENACTIVE Network of Excellence. Special thanks go to Barbara Mazzarino, Ginevra Castellano for her help in compiling the results, and Gualtiero Volpe for his contribution to the development of the EyesWeb blocks. I would also like to thank Antonio Camurri for welcoming me as a research intern, and Marcelo Wanderley for making this collaboration possible.

I would like to thank the guitarists who participated in the tests: Jean-Marc Juneau, Riccardo Casazza, and Bertrand Scherrer. Thanks to Richard McKenzie for his help with the video setup for the preliminary analyses. Thanks also to Donald Pavlasek, Electrical and Computer Engineering, McGill University, for the conception and implementation of the guitar camera mount.

Special thanks to Ichiro Fujinaga, David Birnbaum, Pascal Bélanger, Mélanie Burns, and Marcelo Wanderley for proofreading and constructively commenting on this thesis, and to Vincent Verfaille for his help with LaTeX formatting.

Special thanks to Riccardo Casazza, Yves Boussemart, Marcelo Wanderley and Guy Chagnon for their ideas about the future development of the application.

This thesis has been not only a school project but also an odyssey that completely changed my life. Although I didn't realize it at the time, my research began long before my enrollment in the Master's program. In that regard, I would like to thank my collegiate English teacher, Brent Davis Reid, who fought hard to convince me I could also write in English. Thanks to him I had the desire to pursue studies in English, to master that second language that I used to find so frustrating!

I am grateful to everyone in the Music Technology area who welcomed me in the Minor program. Thanks to the enthusiasm of Marcelo Wanderley, a new professor at the time who later became my thesis supervisor, I decided to continue with the Master's program. In a moment of doubt, Marcelo suggested the internship in Genova which oriented my research and my life.

Since that time, I have been traveling back and forth between Italy, France and Québec, managing my personal life and an exciting new international career in research. Thanks are due to the Italian and Quebec governments for funding my internship, McGill Alma Mater for helping me to present at conferences, Nicole and Paul Vieille, the École des mines de Paris à Sophia Antipolis for giving me the opportunity to stay in France for an internship, and most importantly to my family and friends all around the world who have supported me in moments of uncertainty and doubt. Thank you all for simply being there.

Thank you very much! Merci beaucoup! Grazie mille!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis presents a prototype to retrieve information about a guitarist left-hand fingering using computer vision. This chapter briefly discusses the motivations that led to the choice of the guitar as the instrument of study, the choice of gesture and especially left-hand fingering as the element of interest, and the choice of computer vision as the direct gesture acquisition method. Section 1.1 is a brief history of the guitar. It explains how the widespread usage of the guitar in many styles and cultures together with its affordable cost makes it an instrument of choice for this thesis. Section 1.2 discusses the study of gesture from two approaches, the Human-Computer Interaction (HCI) community approach and the music community approach. Section 1.3 explains the importance of the left-hand fingering gesture in the guitar sound production. Section 1.4 and section 1.5 outline the objectives of the thesis and its possible implications on various fields of research. Finally, section 1.6 presents the structure of the thesis.

1.1 The Guitar

The following definition of the guitar can be found on the Wikipedia's guitar webpage:

> The guitar is a fretted, stringed musical instrument. Guitars are used in a wide variety of musical styles, and are also widely known as a solo classical instrument. They are most recognized in popular culture as the primary instrument in blues, country, and rock music. The guitar usually has six strings but 12 strings instruments are also found, these are used mostly in classical and

> folk music. (Guitar section)

It is precisely the popularity of the guitar among a wide variety of musical styles, musicians and cultures and its use as a solo, orchestra, and accompaniment instrument that make it interesting to study. The musicians who play it are either professionally trained musicians or self-learners without formal musical training and they come from a variaty of styles: classical, flamenco, blues, pop, country, rock, jazz, etc. The guitar itself comes in many “flavors” (acoustic: classical, flamenco, 12-string, etc., and electric), it is made of different materials and varies considerably in price.

The guitar history and repertoire are rich. The first instruments resembling the guitar are believed to have appeared in the 2000-1500 BC and have been observed in ancient carvings and statues recovered from the old Iranian capitol of Susa. The modern six strings guitar is derived from the Spanish vihuela, dating from the antiquity. The earliest extant guitar is attributed to Gaetano Vinaccia an Italian luthiers from Naples and was built in 1779. The electric guitar was patented by George Beauchamp in 1936 (Wikipedia, 2006).

In conclusion, the guitar is an accessible and affordable instrument. It is played by a wide range of musicians in different styles. It is also a mature instrument with an established repertoire. All these characteristics motivated the choice of the guitar for this study.

1.2 The Study of Gesture

Gesture is studied in many fields including human-human communication, human-computer interaction (HCI), and music. As it will be discuss in chapter 3 each field and even each branch of research attribute a different meaning to the word gesture and use a different terminology to describe it. However, there exist some common points. Gesture is associated with the motion of the body or of a limb and carries a meaning. In the context of human-human communication, this meaning will mostly be communicative, but in the context of HCI and music this meaning can also be manipulative. In fact, one goal of the HCI community is to make the usage of computer intuitive by creating interfaces that human can manipulate like everyday objects. The musical context is also composed of both communicative (expressions, emotions, etc.) and manipulative gestures (plucking, fingering, bowing, blowing, etc.) (Cadoz & Wanderley, 2000).

In traditional instruments, there exists a close link between the musician and his instrument. As explained in Kvifte and Jensenius (2006), listeners are able to understand, at various degrees, the interplay between the musician, the instrument, and the resulting sound. The mechanical aspect of traditional physical instruments also provides the performers with various levels of feedback. An active branch of the music technology research is dedicated to the understanding of the gesture-sound relationship in both traditional and new instruments. A community of researchers is working to establish standards to describe gesture in music (Jensenius, Kvifte, & Godøy, 2006).

1.3 Fingering in Guitar

The sound produced by an instrument is influenced by its physical characteristics but also by the way the musician interacts with it. On an instrument like the guitar, both hands perform a distinct but complementary set of actions. The left-hand fingering gesture is the action performed by the musician to modify the string vibration length and, therefore, determines its pitch. The point where the musician presses the string against a fret is called the fingering point.

Generally, many different fingering points can be used to produce the same pitch. In fact, each pitch can be fingered at one to four fret positions, and theoretically each fingered position could be played by any of the four fingers. Consequently, for a score containing n notes, there can exist a maximum of 16^n combinations of $(string, fret, finger)$. However, a professional musician will only consider a few of these possibilities. The choice of the appropriate fingering will therefore be determined by many factors, including philological analysis (interpretation of a sequence of notes), physical constraints due to the musical instrument, and biomechanical constraints in the musician-instrument interaction (Radicioni, Anselma, & Lombardo, 2004b). Although the appropriate fingering might be obvious and intuitive for the experienced musician, beginners will often need external guidance because fingering indications are not always included in scores (Gilardino, 1975a, 1975b).

1.4 Objectives and Motivations

This thesis' main objective is the development of a prototype capable of investigating the feasibility and of exploring the potential of the use of computer vision to solve the guitar left-hand fingering problem in realtime during “real” playing situations (i.e. outside of a controlled environment). The choice of computer vision is motivated by the following constrains:

1. The system should account for all factors involved in the choice of a fingering. A computer vision system analyses the musician solution, it is a direct gesture acquisition system, therefore, no assumptions are used on a preferred choice.
2. The system should not need a priori information or analyses of the musical excerpt. A computer vision system takes only the musician’s image in input and outputs the fingering.
3. The musician should not have to adapt his playing style or to wear special devices for the system to output the fingering. Computer vision is a non-obtrusive way to obtain gesture information.
4. The system should be accessible in term of cost, ease to use and allow for the reproducibility of the results.

1.5 Contributions to the Field

Fingering retrieval is an important topic in music theory and performance. Guitarist fingering has being studied for educational purpose, to help beginners or non-musically trained amateurs (Wang & Li, 1997; Miura, Hirota, Hama, & Yanagida, 2004) and as a compositional help for non-guitarists (Truchet, 2004). It has also been studied for producing more realistic sounds in guitar physical models (Cuzzucoli & Lombardo, 1999; Laurson, Erkut, Välimäki, & Kuushankare, 2001) and in image modeling of a guitarist playing (Elkoura & Singh, 2003). Also, it has impact in automatic music generation (Cabral, Zanforlin, Lima, Santana, & Ramalho, 2001), and in score and tablature generation.

A fingering retrieval system can be useful in education where knowledge of fingering positions as played by an expert could be compared to that of an amateur to identify potential difficulties faced by beginners. In addition, further analysis of the left-hand gesture can permit the use of the system as a controller for self-accompaniment and sound synthesis.

Finally, this research can benefit the HCI community since the testing of a selected group of finger-localization algorithms on a specific musical problem, as investigated here, can lead to improvements that would have been impossible in a more general context. It is also possible that knowledge gained on the guitar could later be applied to other string instruments and, with small adaptations, to various instruments for which the fingering problem exists.

1.6 Thesis Overview

The remaining portion of this thesis is divided in seven chapters. Chapter 2 is a review of the existing methods for guitar fingering retrieval. Chapter 3 presents a terminology and methodology used by the human-computer interaction (HCI) computer vision community. The following chapters describe the process that lead to the prototype. Chapter 4 describes a preliminary work using existing software components to solve the fingering recognition problem. Chapter 5 presents a study of four general purpose methods to locate fingertips. Chapter 6 explains how string and fret detection, fingertips localization, and movement segmentation were used together for the design and development of a guitarist fingering retrieval method based on computer vision. Chapter 7 and 8 discuss future work to be done to ameliorate the prototype and eventually make it widely usable and present the overall conclusions.

Chapter 2

Fingering Recognition

Fingering is an especially important aspect of guitar playing, as it is a fretted instrument where many combinations of string, fret, and finger positions can produce the same pitch. As discussed in the previous chapter, fingering retrieval is an important topic in music theory, music education, automatic music generation and physical modeling. Unfortunately, as Gilardino noted (1975a, 1975b), specific fingering information is rarely indicated in musical scores.

Fingering can be deduced at several points of the music production process. Three main strategies are:

- Pre-processing using score analysis;
- realtime using MIDI guitars;
- Post-processing using sound analysis;

This chapter will review existing approaches for each of these three strategies. The first section will present a category of approaches based on score analysis. The second section will present approaches based on MIDI guitars and guitar-like controllers. The third section will present an approach based on sound analysis. Finally, a realtime approach based on computer vision will be presented.

Fingering information can be retrieved through score analysis. The score is fragmented in phrases, and the optimum fingering for each phrase is determined by finding the shortest path in an acyclic graph of all possible fingering positions. Weights are assigned to each position based on a set of rules. The problem with this approach is that it cannot account for all the factors influencing the choice of a specific fingering, namely philological analysis (interpretation of a sequence of notes), physical constraints due to the musical instrument, and biomechanical constraints in the musician-instrument interaction. Outputs of systems using this approach are similar to human solutions in many cases, but hardly deal with situations where the musical intention is more important then the biomechanical optimum fingering.

Other systems retrieve the fingering during or after a human plays the piece. These approaches uses MIDI guitars or guitar-like controllers. Theoretically, using a MIDI guitar with a separate MIDI channel assigned to each string, it is possible to know in realtime which pitch is played on which string, thus determining the fret position. In practice however, MIDI guitar users report several problems, including a variation in the recognition time from one string to another and the necessity to adapt their playing technique to avoid glitches or false note triggers (Verner, 1995). Guitar-like controllers are instruments on their own and therefore musicians have to develop new playing styles that only mimic that of the traditional guitar.

An approach using the third strategy is the study of the guitar timbre. Traube (2004) suggested a method relying on the recording of a guitarist. The method consists of analyzing the sound to identify the pitch, finding the plucking point and then determining the string length to evaluate the fingering point. Shortcomings of this method are that it works only when one note is played at the time, and the error range of the string length evaluation (and therefore the fingering evaluation) is better than one centimeter in the case of open strings but can be as high as eight centimeters in the case of fretted strings (Traube & Smith III, 2000).

Fingering information can also be retrieve in realtime with the help of computer vision. The last section of this chapter presents an approach that has been used in concerts to control realtime sound effects applied to the live guitar sound output recorded by a microphone. This approach use a computer vision algorithm to locate the painted fingers of the musician captured throughout a video camera mounted on a tripod in front of the musician. This approach impose some constraints to the musician: that he paints his fingers and that he restrains his movement so that the guitar-neck is always at the same position in the captured image.

2.1 Pre-Processing: Analysis of the Score

This ensemble of techniques consists of analyzing a score as input and producing an optimum fingering for that score as output. Mostly all research in the field is based on representing all possible fingering triplets $(string, fret, finger)$ for a musical excerpt using nodes of a graph and on assigning a weight to each edge according to a pre-determined set of rules. The graph is then searched for the optimum path that corresponds to the computed optimum fingering. Fingering of chords is determined using a similar rule strategy but is based on the computer science theory of Constraint Satisfaction Problem (CSP). Recent research on guitar fingering extends that of Sayegh (1989) that proposes a computationally efficient solution to the fingering process in string instruments. It also has been influenced by the work of Parncutt, Sloboda, Clarke, Raekallio, and Desain (1997) with the extensions of Jacobs (2001) that proposed an ergonomic model for determining keyboard fingering from the analysis of short fragments of a score.

In this section, a summary of the techniques used by Radicioni et al. (Radicioni et al., 2004b; Radicioni, Anselma, & Lombardo, 2004a; Radicioni & Lombardo, 2005b, 2005a; Radicioni, 2005) will be presented as it is the most complete and representative work in this field to the author knowledge. As it can be seen in figure 2.1, Radicioni's method is based on a prior manual segmentation of the score into musical phrases. In a second step, he applies a cognitive model of the human-instrument interaction to the musical phrase. This model is based on a behavioral study of the complexity of a guitarist left-hand movements by Heijink and Meulenbroek (2002), theoretical score analysis explanations by Gilardino (1975a, 1975b) and from insights of a professional guitarist. Using this model, the fingering information can be computed and output to any other system. To validate his system, Radicioni proposed the coupling of his system with a sound processing guitar physical model developed by Cuzzucoli and Lombardo (1999), since his interest is in automatic music performance. He claims that his approach is more cognitively reliable and closer to a human expert's solution than Sayegh's (1989) global optimization approach.

2.1.1 Manual Score Analysis

Even if there exist some attempts to do automatic score partitioning in the literature (Bod, 2001, 2002; Pardo & Birmingham, 2000), Radicioni chooses to manually segment the score in musical phrases. His principal motivations for segmenting the score in phrases are the

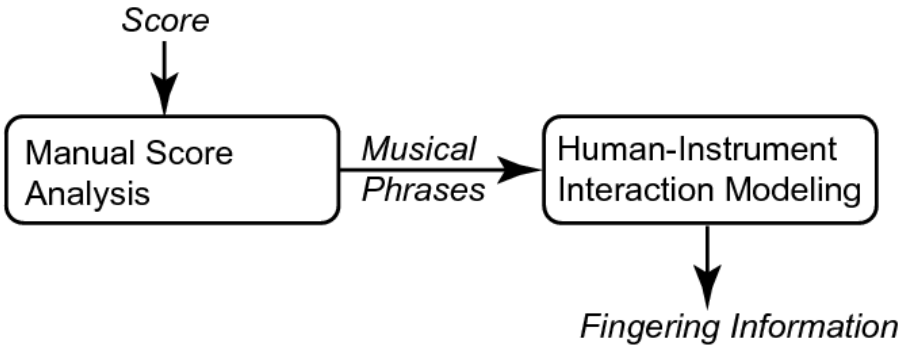


Figure 2.1 Fingering retrieval from score analysis algorithm (Radicioni et al., 2004b)

following:

- “A global optimization approach [...] is not cognitively reliable, since both human performers and listeners do break musical pieces into phrases, themes, motives, etc. . . . (Radicioni et al., 2004b, p.5).”
- “Performing a piece of music implies discerning and emphasizing structural features of the music such as structural boundaries (Radicioni et al., 2004b, p.5).”
- “ [...] If performers aims at clarifying the score structure by means of segmentation, then they will likely employ fingering too in order to mark further those boundaries, being mainly concerned in “optimizing” fingering inside these margins(Radicioni et al., 2004b, p.5).”

Therefore, using the assumption that the apparent natural tendency to break a musical piece or score into smaller structures also happens during the fingering process, their system takes in input musical phrases segmented from the score by a professional musician. The system optimizes the fingering inside these phrases, first, and then finds the best possible fingering at the boundaries between each phrase.

2.1.2 Human-Instrument Interaction Modeling

The choice of the “best possible fingering” is done inside each phrase and then between the phrases by applying a model of the interaction between the human and the instrument. Generally, different fingerings can be used to produce the same pitch (see figure 2.2). The choice of the appropriate fingering will therefore be determined by many factors, including philological analysis (interpretation of a sequence of notes), physical constraints due to the musical instrument, and biomechanical constraints in the human-instrument interaction (Gilardino, 1975a, 1975b). However, these constraints are sometimes conflicting and a model has to compromise while keeping the emphasis on the most significant constraints. To create a human-instrument interaction model, Radicioni relies on the bio-mechanic assumption that the performer will choose the easiest solution for fingering. This has the advantage that in many cases, as Sayegh (1989) explained, this will also satisfy one of the musical intentions, that is, to produce homogenous sounds. In fact, as Heijink and Meulenbroek (2002) illustrated, a strategy that is biomechanically advantageous is to keep the hand at the same position. Hence, it also happens that remaining at the same position permits the production of sound of uniform quality because the hand thereby placed has access to section of strings of approximately equal length, therefore, the relative damping of higher harmonics of the different notes is similar (Sayegh, 1989). According to Radicioni tests, modeling the human-instrument interaction on biomechanical constraints seems to be cognitively and musically acceptable. It is important to note, however, that this conclusion might only hold on a specific type of musical excerpts.

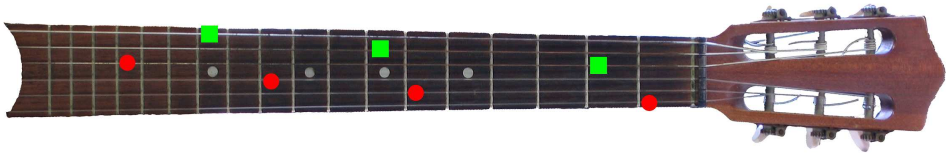


Figure 2.2 Fingering ambiguity problem a) dots illustrate all the possible fingering for F4 (fa4) and b) squares illustrate all the possible fingering for E3 (mi3)

To model the human-instrument interaction (figure 2.3), Radicioni relies on Heijink's and Meulenbroek's (2002) study on the complexity of the task of classical guitar playing. First, he defined two categories of movement of the hand: moving *ALONG* the fingerboard, hence horizontally, and moving *ACROSS* the fingerboard, hence vertically. He determined that since both motions have to be performed in the same amount of time, moving along the neck should be more penalized, since it requires hand repositioning, than moving across the neck that simply implies finger displacements. The difficulty between two fingering positions i and j is therefore computed using the following formula:

$$Difficulty(i, j) = ALONG + ACROSS \tag{2.1}$$

Where $ALONG$ is computed by assigning a difficulty weight to each possible position and $ACROSS$ mainly by avoiding unnatural postures and by assigning higher weights to positions that do not contribute to maintain a close vertical span. Radicioni has established a table of the maximum allowed distances, expressed in frets, between each finger pair (Radicioni, 2005). Using this table he assigned difficulty weights to $ALONG$ positions according to the following rules:

- No weight is added if the fingering is inside the boundaries of the maximum span;
- Increasing weight is added if the fingering is outside the maximum span boundaries due to the need of repositioning.

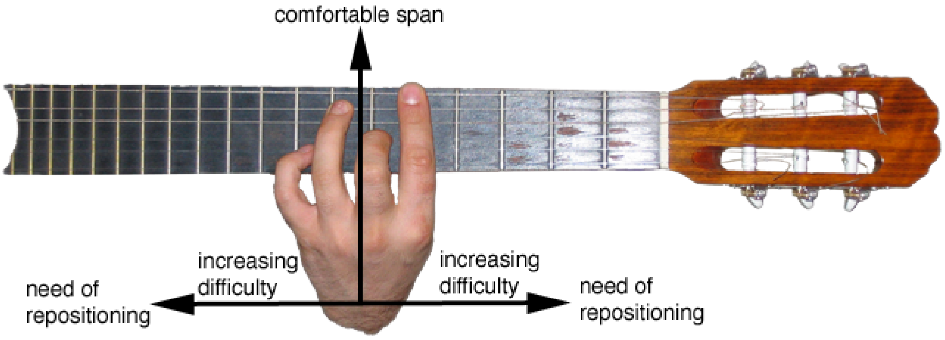


Figure 2.3 Biomechanical constraint: A maximum guitarist left-hand span model

2.1.3 Computing the Minimum-Weight Path

Once all possible fingering positions for each note have been determined and once weights have been assigned to each transition between notes, the problem has been transformed to a minimum-weight path search in a directed acyclic graph. By fragmenting the score in phrases, Radicioni's method first creates and resolves graphs for each phrase and then solves the fingering at the boundaries between phrases. As figure 2.4 displays, the graph has a layered structure where each note has at least four possible fingerings (with the exception of notes played on open strings), i.e., one possible $(string, fret)$ position on the neck played with any of the four fingers. This layered structure ensures that the system can resolve the minimum-weight path searching problem in linear time ($O(n)$) relative to the number of notes (n) and to a hidden linear complexity constant relative to the number of edges between each layer (256 in the worst case).

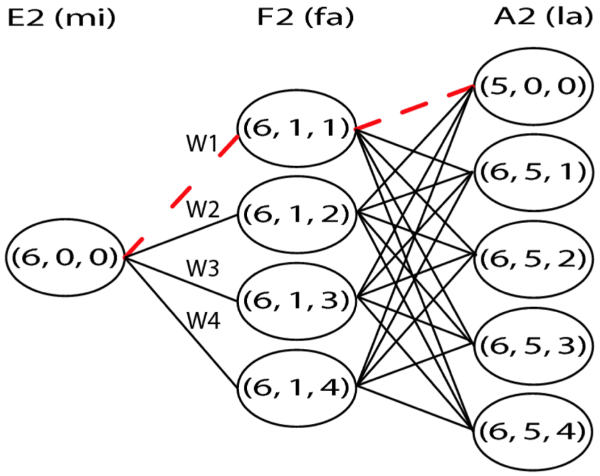


Figure 2.4 A simple example of the graph generated for the sequence of notes E2-F2-A2 (weights are omitted on edges between F2 and A2 for clarity). The dash line represents the optimum path.

2.1.4 Computing Chord Fingering

The case of the fingering of chords is slightly different from the one of single note explained in the precedent paragraphs since it involves many fingering positions at a time. The solution to solve it is also different. Radicioni (Radicioni, 2005; Radicioni & Lombardo, 2005b) and Truchet (2004) both demonstrated that the chord fingering problem could be solved using the constraint satisfaction problem theory (CSP). The constraint satisfaction problem assigns values to variables satisfying a set of constraints (or rules, to keep the same language as in the previous paragraphs). In this case, the variables are the individual notes that compose the chord, the values are all the possible triplets of $(string, fret, finger)$ (up to 16 per note) that can solve the note, and the constraints are the set of rules (biomechanical constraints in the case of Radicioni) used to determine the optimum fingering. In the case where many solutions satisfy the set of rules, Radicioni developed a comfort ranking to choose the optimum solution. To solve the fingering problem in case of a succession of chords or of a mixed passage of one note melody and chords, chords are included in the search graph and are solve as a sub-problem using the CSP technique.

2.1.5 Conclusion

This section presented a pre-processing solution to the fingering problem. This solution is advantageous when the score is the only information available about a musical piece. This method searches an optimum fingering based on a pre-determined set of rules and therefore does not necessarily satisfy the musical intent of the original composer of the piece. According to Gilardino's (1975a, 1975b) comments on manually determining guitar left-hand fingering from score analysis, it is not possible to establish a set of rules that will satisfy all musical genres. This, therefore, implies that an a priori study of the piece could be necessary to determine which set of rules needs to be applied, assuming that such sets can be created. However, Radicioni's comparative results between a professional musician and the computerized solution were satisfying. The system has demonstrated some difficulties at the boundaries and also at places where the musician also hesitated between two solutions. This method is potentially useful in domains like automatic music performance, instrument physical modeling, and music theory but cannot be applied directly in live performances to control sound synthesis, sound effects, or accompaniment. It can have some utility in education (Wang & Li, 1997; Miura et al., 2004) but cannot be used to analyze the evolution of a beginner or to compare his fingering choice to that of a professional.

2.2 Realtime-Processing: MIDI Guitar and Guitar-shaped Controllers

Realtime guitarist gestures have been used to control sound synthesis for approximately thirty years either by capturing these with devices added to traditional guitars or by using guitars-shaped controllers. In the 1970's, when the popularity and availability of keyboard synthesizers were growing, "it came to pass that guitar players would have the same performance potential as keyboard players" (ARP instruments advertisement cited by Verner (1995)). The desire to capture guitarist gestures in realtime to control sound effects and sound synthesis was born and is still an active research topic today. Commercial solutions exist for the acquisition of the left-hand fingering gesture of the guitarist. These solutions solve the $(string, fret)$ component of the problem. The most widely used solution is the pitch-to-voltage technology. Its popularity is due to its relatively low price and its simple setup. Other commercial solutions exist, for example, wired frets and ultrasonic pitch sensor scanning. These tend to be more accurate and to have shorter response time, but their high production cost make these hardly accessible to most musicians. Finally, recent research also proposes some solutions using advances in the sensor technology field.

2.2.1 Pitch-to-Voltage Converter

In 1976, the ARP Instruments Avatar analog monophonic synthesizer was the first commercial solution to use the guitar as a sound synthesis controller (Vintage Synth Explorer, 2005). Left-hand guitarist gestures were captured from a traditional guitar with the use of a pitch-to-voltage converter. With the advent of the MIDI protocol in the 1980's, pitch-to-voltage converter has been renamed pitch-to-MIDI but uses the same principles:

1. The strings' vibrations are captured via a hexaphonic pickup.
2. Six electric signals (one for each string) are then sent to a converter that determines their frequencies (pitches).
3. Signals are also analyzed to retrieve information about the velocity (amplitude).
4. In the case of a MIDI instrument, these information are then converted to digital messages according to the MIDI protocol.

Using this method it is possible to retrieve information about the $(string, fret)$ component of the fingering problem.

2.2.2 Other MIDI Solutions

Although pitch-to-voltage is still in use and is the MIDI guitar industry bestseller, other technologies have also been used. Launch in 1984, the SynthAxe (Rojas, 2005), a guitar-like MIDI controller, is the oldest representative of the wired frets group. Today, StarrLabs (2006) produces the Ztars guitar-like controller series that are inspired by this technology. The principle of this technology is that each fret is wired and divided in six regions. The contact of the string on the fret triggers the note.

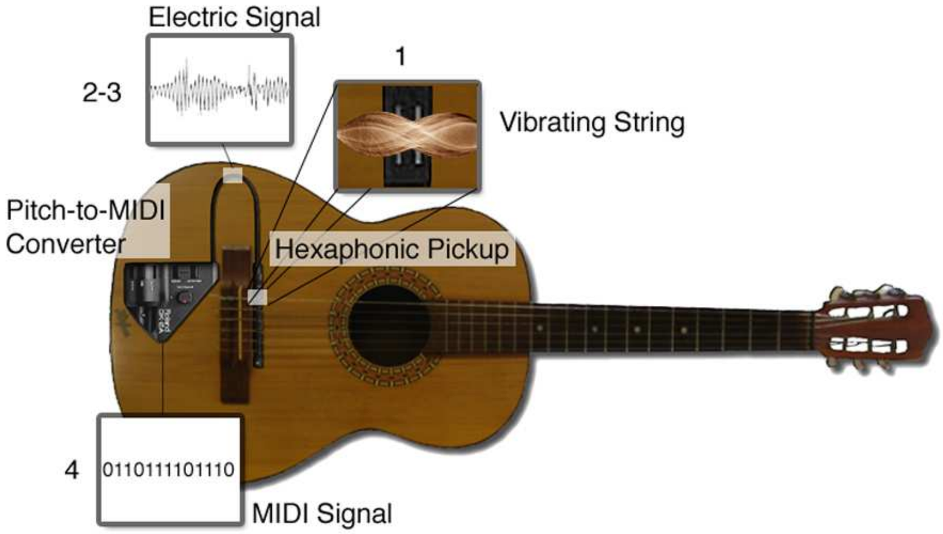


Figure 2.5 A typical MIDI guitar setup

Ultrasonic pitch sensor scanning guitar-like controllers were produced in the late 1980's by Yamaha and Quantar. In this technology, high-frequency ultrasound signals are transmitted across the strings and the fingered fret position is determined by analysing the reflected wave (Verner, 1995). These two technologies address the $(string, fret)$ component of the fingering problem only.

2.2.3 Recent Research Solutions

Recent advances in sensors technology have permitted the development of novel guitar-like or augmented guitar controllers, for example, the GXtar was presented in NIME06 (Kessous, Castet, & Arfib, 2006). It is a guitar-like controller on which the fingerboard has been replaced by two FSRs (Force Sensing Resistor) capable of measuring the finger position and pressure. This instrument used two silent strings to guide the "guitarist" and help him reproduce traditional left-hand gestures. Right-hand gestures are produced using a three-dimensional joystick mounted on a slider, but a plectrum-like sensor has also been evaluated to extract right-hand gesture information from the "plucking" signal. This instrument is fretless and limited to two "strings" due to FSR dimensions but this is simply a technological limitation. Other sensor technologies are actually tested to create new augmented guitars or guitar-like controllers, for example InfoMus laboratory (University of Genoa) is working on an infrared solution to measure the distance between an infrared source mounted on the instrument and the fingered position (Camurri, personal communication, June 7, 2006).

2.2.4 Conclusion

This section presented several solutions to solve the fingering problem in realtime. All of these solutions are able to retrieve information about the $(string, fret)$ component but none solves the complete $(string, fret, finger)$ triplet of the fingering problem. MIDI guitar controllers using pitch-to-voltage converter suffer from several problems inherent to the technology. The pitch detection time varies from string to string since lower frequencies necessitate more periods to be identified. Each note also has to be played in a clean and detached (*staccato*) way in order to avoid glitches or false note triggers. Guitarists consequently have to adapt their natural playing style or to perform post-processing editing to get satisfying results with pitch-to-MIDI guitar controller (Glatt, 1999; Pollock, 1999, 2000). Recent advances in artificial intelligence might bring solutions to these problems, for example Blue Chip Music Technology (n.d.) claims to have developed a neural network solution capable of determining the pitch even before the attack ends. Industrial guitar-like controllers seem to offer better solutions in terms of the accuracy, precision, and realtime response. However, they are unique instruments and musicians have to develop an appropriate playing style that only mimic that of the traditional guitar. Their high cost is also a limitation to their wide use. Infrared sensor technology could offer an interesting alternative to the pitch-to-MIDI solution but it will also only answer the $(string, fret)$ component of the fingering problem, and would probably need to be combined with other techniques to determine when a string is played. At the moment of writing this thesis, this technology is only at the conceptual step and will be tested on the violin first (Camurri, personal communication, June 7, 2006).

2.3 Post-Processing: Sound Analysis

This last method is based on the analysis of sound recording of a guitar. It has been inspired by the observation of the timbre space of the guitar (Traube, Depalle, & Wanderley, 2003). The major point of this method is that comparison between the physical model of a guitar and the sonic parameters of a recording permits to determine some of the guitarist playing techniques. The main task is to identify the plucking point with the highest degree of accuracy. Once a relative string length ratio has been determined for the plucking point, it is possible to compare it with ratios of a fixed plucking point with different string lengths and therefore find the fingering point (the point where the left hand of the guitarist shorten the string).

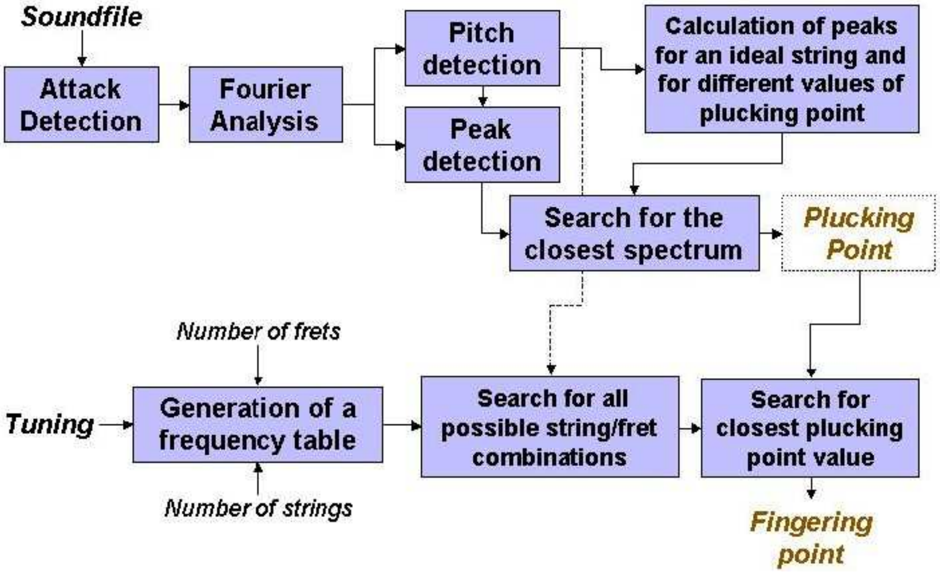

Soundfile
Attack Detection
Fourier Analysis
Pitch detection
Peak detection
Calculation of peaks for an ideal string and for different values of plucking point
Search for the closest spectrum
Plucking Point
Number of frets
Tuning
Generation of a frequency table
Number of strings
Search for all possible string/fret combinations
Search for closest plucking point value
Fingering point

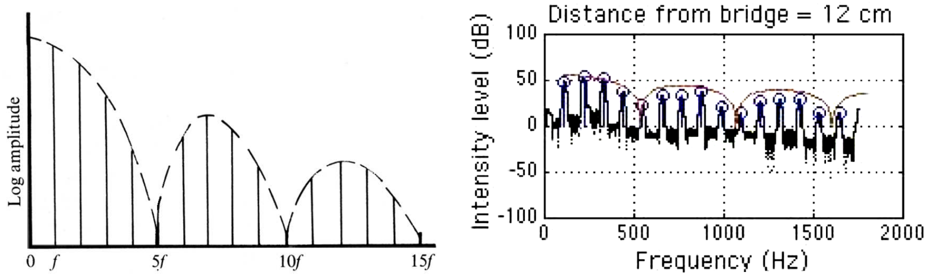
Figure 2.6 Block-diagram for the estimation of the plucking and fingering points reproduced with permission from (Traube & Smith III, 2000)

2.3.1 Finding Notes and Pitches

The first step of this method consists in finding note beginnings. This is done by observing the energy of successive samples blocks and by detecting sections of the recording where the energy increased by a factor of 2. When the notes' attacks are found, stationary parts between two attacks are taken in order to determine the notes' pitches. To determine the fundamental frequency of notes, Fast Fourier Transforms (FFT) are performed on windows of the waveform taken at approximately 1/8th of the distance between two attacks. Pitches are determined by searching for a maximum in each spectrum. In case the maximum is not the first frequency of a spectrum, previous peaks are evaluated to determine if their height is significant enough to be fundamentals. This is done knowing that generally the fundamental of a guitar note is the first and highest peak of the spectrum.

2.3.2 Estimating the Relative Plucking Point

In the first implementations of this method (Traube & Smith III, 2000, 2001), the next step was to generate spectra of an ideal string pluck at different positions for each detected pitch. These spectra were then compared to the real spectrum of each note. The spectra minimizing the error between the real data and the theoretical one were assumed to be a good approximation of the real notes and plucking points. This method has been refined in subsequent articles (Traube & Depalle, 2003; Traube et al., 2003; Traube, 2004). In its latest implementation it uses a log version of the autocorrelation function, that the author named the log-correlation. The log-correlation of the theoretical model and the real data produced a first approximation of the *(plucking point / string length)* ratio that is then refined iteratively using a weighted least-squared estimation. These estimations are based on the fact that plucking a string created an effect similar to that of a comb filter by eliminating harmonics that have a node at the plucking position. This resemblance is exploited in many synthesis methods based on physical models of the guitar (Cuzzucoli & Lombardo, 1999; Laurson et al., 2001), explanations on its physical foundations can be found in *The Science of Sound* (Rossing, Moore, & Wheeler, 2002) and in Traube "The physics of Plucked String" doctoral thesis chapter (2004). Traube also provides detailed information on how the plucking effect can be reproduced with comb filter in her "The Plucking Effect as Comb Filtering" chapter.



(a) Theoretical spectrum

(b) Real spectrum

Figure 2.7 Spectrum of a plucked string: (a) Theoretical model of a string pluck at 1/5 of its length (Rossing et al., 2002) reproduced with permission from (Traube, 2004). (b) Real spectrum of a string pluck at approximately 1/5 of its length reproduced with permission from (Traube, 2004).

2.3.3 Determining the Fingering Point and the Absolute Plucking Point

One of the problems encountered when trying to retrieve the fingering of a guitarist is that the same pitch can be produced with different fingerings (see figure 2.2). The choice of a particular fingering is influenced by multiple factors that are not strictly physical, therefore, an educated guess is not possible. A frequency table can be generated by knowing the number of strings and frets and by multiplying the tuning frequency by 2 F/12 (a semitone). Once the pitches are known, possible ($string, fret$) combinations are retrieved from the table. For each ($string, fret$) combination a relative *(plucking point / string length)* ratio is computed using a fixed plucking point and knowing that the vibrating string length is inversely proportional to the fundamental frequency. These *(plucking point / global length)* ratios are then compared with the one computed at the relative plucking point estimation step. The *(plucking point, fingering point)* combinations that minimize the error between the two ratios are assumed to represent the reality.

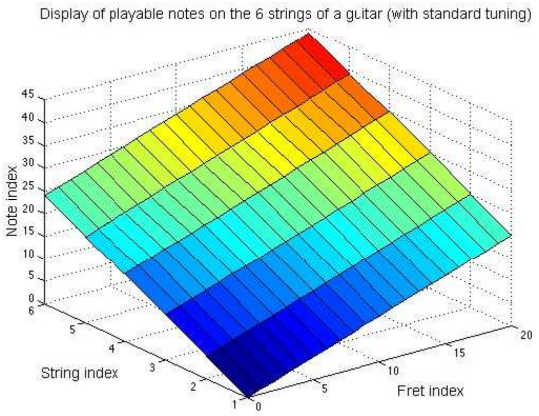


Figure 2.8 Frequency table based on the tuning, the number of strings and the number of frets reproduced with permission from (Traube & Smith III, 2000)

2.3.4 Conclusion

This section has introduced a method to detect guitar fingering using sound recordings. The principal advantage of this method is that it could be used to retrieve a guitarist fingering when only sound recording of the piece are available. This method is efficient but its accuracy to detect fingering is dependent on the detection of the plucking point.

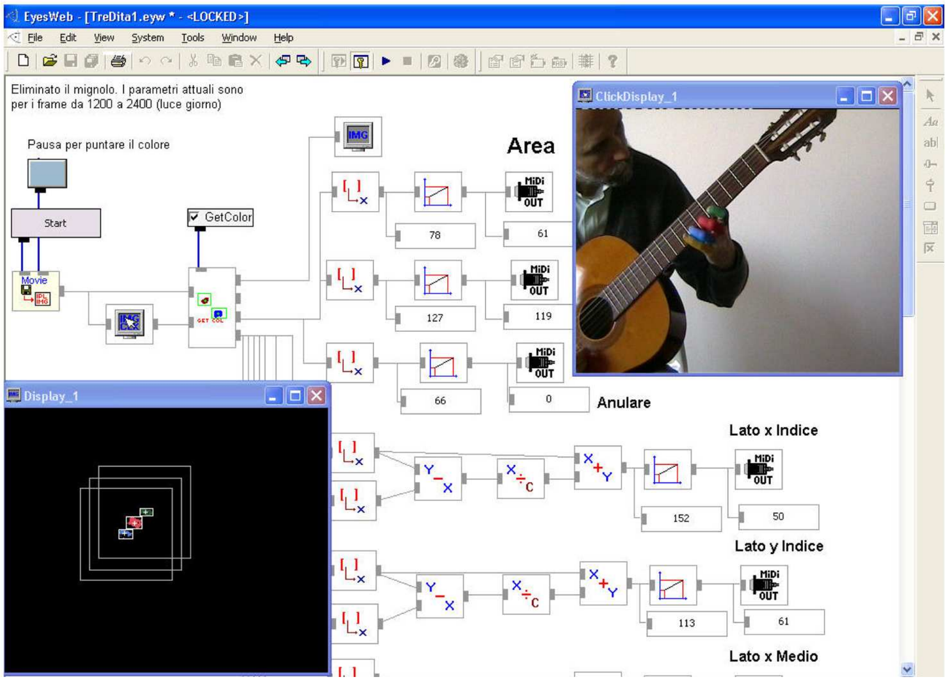
The plucking point detection algorithm has shown accuracy results in the range of one centimeter for notes played on open strings. Unfortunately, the results for fretted notes, the most important ones in the fingering problem, are not as good. In the worst case, the accuracy goes down to errors between 3.8 and 8.3 centimeters. These errors might be due to the friction of the string on the fret and to sympathetic resonances with open strings. These two factors introduce distortions that make the analysis more difficult and less accurate. Moreover, this method can only be applied if one note is played at the time. Also, it cannot be directly applied to electric guitar because in that case, the sound is recorded by pick-ups that introduce another comb-filter effect. In fact, since pick-ups record the sound at a particular point, it will miss all the harmonics that have a node at this point. Since the combination of pick-ups is not always the same depending on the guitar this might introduce complications and reduce the principal advantage of this method. This method therefore needs further development to refine the results with fretted notes and to be applied on any guitar and in real playing situations where multiple notes are played at the time.

2.4 An Application Example: A Realtime Guitar Performance Relying on Computer Vision

This section presents an example of the use of computer vision to retrieve gestural information from a guitarist's live performance. This computer vision system was developed at InfoMus Laboratory, University of Genova and has been used by the composer Roberto Doati for a piece requested by the guitarist Elena Casoli (Doati, 2006). The live electronic version of "L'apparizione di tre rughe" has been performed in concert in 2004.

2.4.1 Finger-Localization Setting

The setting is simple, the musician is sitting in front of a camera and a microphone mounted on a tripod. The camera is focused on the guitar neck region. Three of the musician's fingers are painted, the index is green, the middle finger is red, and the ring finger is blue. An EyesWeb color localization patch is used to detect the painted left-hand finger positions on the guitar neck. From this information, 15 different parameters are extracted and are translated into MIDI messages that are sent to MAX/MSP patches that apply live digital effects to the captured sounds. Different patches and mappings are used at different moments of the piece.



Eliminato il mignolo. I parametri attuali sono per i frame da 1200 a 2400 (luce giorno)

Pausa per puntare il colore

Figure 2.9 EyesWeb detecting the colored fingers of Roberto Doati reproduced with permission from (Doati, 2006)

2.4.2 Conclusion

This method does not retrieve precise finger positions nor relate these to exact $(string, fret)$ coordinates. The system only have an idea of the position of the whole finger in the camera view window. The musician is required to paint is fingers and should restrain his movements in order to keep the guitar neck stable inside the camera view. However, Doati comments that “the results in terms of articulation are much more “natural” than with a normal sliders MIDI controller (Doati, 2006, p.18).” This example demonstrates that there is a desire for controllers that use the natural guitarists’ skills and gestures to control parameters external to the guitar. This also shows that computer vision might offer solutions for these kind of controllers.

Chapter 3

A Methodology for Gesture Processing Using Computer Vision

The advances in technology and the widespread usage of computers in almost every field of human activity call for new interaction methods between humans and machines. The traditional keyboard and mouse combination has proved its usefulness but also, and in a more extensive way, its weaknesses and limitations. In order to interact in an efficient and expressive way with the computer, humans need to be able to communicate with machines in a manner more similar to human-human communication (Picard, 1997).

In fact, throughout their evolution, human beings have used their hands, alone or with the support of other means and senses, to communicate with others, to receive feedback from the environment, and to manipulate things. It therefore seems important that technology makes it possible to interact with machines using some of these traditional skills.

The human-computer interaction (HCI) community has invented various tools to exploit human gesture, the first attempts resulting in mechanical devices. Devices such as data gloves can prove especially interesting and useful in certain specific applications but have the disadvantage of often being onerous, complex to use, and somewhat obtrusive.

The use of computer vision can consequently be a possible alternative. Recent advances in computer vision techniques and the availability of fast computing have made the realtime requirements for gesture recognition in HCI feasible. Consequently, extensive research has been done in the field of computer vision to recognize hand postures and static gestures, and also, more recently, to interpret the dynamic meaning of gesture (Kohler, n.d.; Pavlovic, Sharma, & Huang, 1997). Computer vision systems are less intrusive and impose lower constraints on the user since they use video cameras to capture movements and rely on software applications to perform the analysis.

This chapter presents a model of gesture processing using computer vision. It is mostly based on Pavlovic et al. (1997) review of the field. The first section discusses the difficulty to come to a single definition of gesture and proposes various vocabulary words that will be used throughout the rest of this thesis. The subsequent sections suggest a methodology to recognize gesture based on three steps:

- Gesture modeling;
- Gesture analysis;
- Gesture recognition.

The modeling step implies the representation of gesture in time and in space. The analysis step is concerned with the algorithmic mechanisms necessary to transform the captured images into data corresponding to the chosen model. The recognition step confronts the data with the model to associate these to a gesture. Finally, the conclusion section will explain how this methodology can be used in the context of the guitar fingering problem.

3.1 Gestures Definition

Although human gesture is widely studied in various fields, it seems impossible to find a single and simple definition of it (Cadoz & Wanderley, 2000). However, a common denominator relates it to human physical behavior. In all cases, gesture is associated with the idea of motion of the body or of a limb. What is not agreed upon is whether this motion should convey information or whether manipulation and expressive movements can be considered gestures. In the domain of HCI, where the aim is to control the computer using gestures, it is desirable to use body and limb movements that mimic both manipulation (also called practical gestures (Kendon, 1986)) and communication gestures.

The music technology community tends to define gesture as the physical representation of the communication between the musician and his instrument. It is the mean by which the musician's effort and expressiveness are transmitted to the instrument and converted to mechanical energy. Part of gesture is essential to sound production while the rest is related to feeling and emotion. Gesture can therefore be divided into two categories: effective and accompanist (Delalande, 1988). Effective gesture is necessary for sound production. It is the action performed by the musician to produce the mechanical energy that permits his instrument to generate sound. On the other hand, accompanist gesture, which is also sometimes called expressive or ancillary gesture, is used by the musician to express the emotive content of the piece. Even if this second component does not directly affect the sound production, experiments in which musicians limited their expressive movements have tended to prove that accompanist gesture also contains meaningful information (Wanderley, Vines, Middleton, McKay, & Hatch, 2005). A complete computer system for music should consequently consider both of these types of gestures.

From an HCI perspective, Pavlovic et al. (1997) define gesture as a motion originating from a gesturer's mental concept and perceived by an observer as a stream of visual images. These images are then interpreted by the observer based on his knowledge of their communicative and expressive content. In the case where the observer is a computer, the gesture is captured by one or more cameras. The image stream is analyzed with respect of the gesture model parameter space during a defined time interval. The extracted parameters are then compared using a grammar and a class of known gestures in the recognition phase. Finally, the computer converts the recognized gesture into commands or data required to control an application.

3.2 Gesture Modeling

As there is no absolute definition of gesture, there is no general model that solves all the human-computer interaction gestural problems. However, the following sections present one methodology that can be applied to recognize gesture using computer vision. As figure 3.1 illustrates, gesture from the gesturer is captured by a camera. The captured images are analyzed to localize the region of the image and the moment in the sequence of images where the gesture takes place. The gesture is segmented in time according to the temporal model and in space to create the region of interest (ROI). Parameters are then extracted from the segmented image according to the chosen spatial model. Finally, depending on the type of model, gesture is recognized either by comparing these parameters with a knowledge-base of previously learned gestures or with an alphabet and a grammar establishing the set of possible gestures and sequences of gestures.

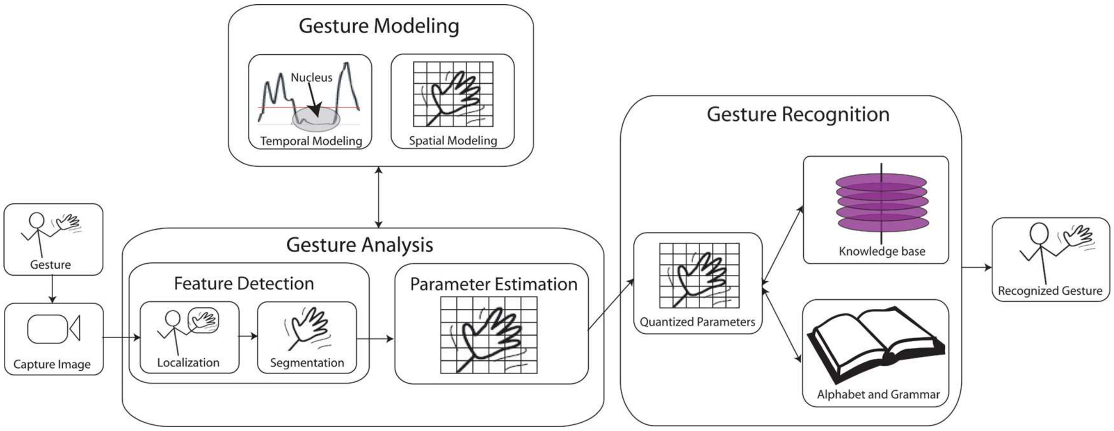


Figure 3.1 A methodology for gesture processing using computer vision

3.2.1 Temporal Modeling of Gestures

In 1997, Pavlovic et al. stated that although gestures are dynamic actions with temporal characteristics rich in meaning and information, most of the past researches focused on the recognition of static gestures or postures. This was mostly due to the lack of established software tools to analyze dynamic gestures and to their high computational cost.

In order to analyze the dynamic component of gesture it is important to understand its mechanism. A good understanding of the different phases of gesture is also essential for its temporal segmentation. Gestures can be divided in three phases:

- Preparation;
- Nucleus;
- Retraction.

The preparation phase is the moment where the movement is initialized from a resting position. The nucleus is the significant part of a gesture; it is the part we want to analyze to recover the meaning of the observed movement. The retraction phase is the moment where the limbs implied in the gesture return to a rest position or enter a new preparation phase. Preparation and retraction phases are normally characterized by rapid motions; during the nucleus phase the motions are generally slower. Due to the complexity of gestural interpretation, these phases are not always easily detectable.

3.2.2 Spatial Modeling of Gestures

Pavlovic et al. (1997) note two major categories of spatial model of gestures. In a first approach, gestures are inferred directly from the observed images. This approach is called *appearance-based*. The second approach is named *model-based* and implies that gestures are inferred from the parameters of models of motion and postures.

The principles of appearance-based models are simple, they use the parameters derived from the images captured by the cameras and compare these to parameters of a set of predefined template gestures. This class of parameters is called *image property parameters*. Image property parameters may be derived using many techniques including but not restricted to: binary silhouettes, edges, contours, signatures, histograms, image moments and eigenvectors. Templates can be obtained by averaging image property parameters of a group of training data representing each of the desired gestures. This approach can also be used for dynamic recognition; in this case the trajectory of the parameters is used. A subgroup of these models uses fingertip positions as parameters with the assumption that fingertip positions are sufficient to describe uniquely a finite group of gestures (Pavlovic et al., 1997).

The model-based approach can itself be divided in two main groups: volumetric models and skeletal models. Volumetric models are three-dimensional representations of the body or limbs with varying degree of realism ranging from fully articulated 3-dimensitional surfaces to cylindrical models with reduced number of joint and restricted degree of freedom. A high degree of realism will have a dramatic impact on computation time, and is not necessarily required to recognize gesture. Therefore, although these complex 3D surfaces are really useful in computer animation where realtime output is not always a requirement, simplified versions are often sufficient in the field of gesture analysis and recognition. The volumetric models approach is also named *analysis-by-synthesis* tracking and recognition. Its underlining concept is to analyse gestures by synthesizing 3D models of the body or limbs and varying its parameters in order to obtain a match between the model and the observed data. Skeletal models use a similar concept but with a toothpick representation of the body. Depending on the application, a reduced number of joints and segments are chosen with established restrictions on the segments possible angles and degrees of freedom (Pavlovic et al., 1997).

3.3 Gesture Analysis

Once a model parameter space has been chosen, the next task is to perform the necessary steps to extract the required parameters from the captured images. The analysis process is performed in two steps. The first step is to extract the images' features. The second is to compute these features so that they correspond to the chosen model. The extraction step is itself dual and consists of localizing the region of the image where the gestures take place and segmenting the features that need to be extracted from the rest of the image. The computation step simply consists of transforming the data into a format corresponding to the model parameter space.

3.3.1 Feature Extraction

3.3.1.1 Localization

The localization step consists in finding the region of interest, therefore, the region of the image where the gestures take place. Traditionally, two types of localization methods have been used to perform this task:

- Color detection;
- Motion detection.

Color detection relies on the identification of human skin in complex background images or on the use of specific color markers or uniform background color. The first case would be ideal since the use of color markers or uniform background color limits the generality of the applications and impose restrictions on the users and environment setup. Unfortunately, color identification is sensitive to illumination change and human skin color range is large and often similar to other environment elements color (wood for example). The use of the hue-saturation space instead of the standard RGB color space provides a solution since it is less sensitive to lighting condition but is still error prone. Some assumptions or educated guesses on the size and potential locations of the searched regions, for example, can be used to enhance the use of color detection. In many cases color detection algorithms are computationally intensive and, therefore, hard to perform in realtime.

Motion detection is based on the assumption that the background, gesturer and cameras are stable and, most of the time, that only one hand gesture is performed, therefore, the analyzed gesture is the only changing component in the image sequence. It uses background subtraction algorithms between a reference image and the actual image to determine the regions of the image where motion is happening. Motion detection is sensitive to phenomena like shadows but its major drawback resides in its assumptions. These assumptions are true in a vast majority of cases but there exist situations where they are constraining (Pavlovic et al., 1997).

Detection of the gesture is a major issue in gesture recognition systems. Since both of the common methods present some limitations, active research is done in this field. Two lines of research are hybrid systems and prediction systems. Hybrid systems rely on a combination of detection methods and prediction systems try to estimate future locations of the regions of interest based on the model dynamics and the previously known locations.

3.3.1.2 Segmentation

When the region of interest has been localized, the next task is to prepare the image so that parameters corresponding to the parameter space model can be extracted. Segmentation of the body or limb from the background is generally the main and more complex part of the process. Background subtraction algorithms use similar mechanisms as gesture localization. The body limb doing the gesture can be segmented from the background by using color segmentation or motion segmentation. As it was the case in the localization phase, color segmentation can be performed based on skin color distribution or with the help of color markers. Background subtraction algorithms relying on motion use difference of pixels between the current image and a reference image and will therefore only keep pixels that have changed more than a given threshold. Background segmentation algorithms suffer from the same drawbacks than their localization equivalent, namely sensitivity to illumination change and shadow, and strictness and limitation imposed by the necessary assumptions.

Another common parameter used in 2D and 3D applications is the fingertip location. Fingertips can be found easily with the help of color markers and a color segmentation algorithm. More complex but less constraining techniques imply the use of pattern matching where the template can be an image of a fingertip or a fingertip generic model. Some techniques also rely on the characteristic properties of the fingertips in the image, for example, the specific curvature of the fingertip can be used for feature detection. However, fingertips are susceptible to occlusion and, therefore, cannot always be detected. In these cases, solutions are to use multiple cameras or to develop estimation techniques to determine the position of occluded fingers.

3.3.2 Parameter Estimation

Parameter estimation is the last step of the analysis process. It is the step where the detected image is transformed into the chosen model format. In the case of appearance-based approach this will correspond to the format required to compare the actual data with that of the knowledge-base. It can be in the form of pixel information like in the case of binary images, contours, and edges, or in a numerical form like in the case of signatures, eigenvectors, or moments. In the case of model-based approach, the information will be in the form of coordinates of points and angles of segments. This information will be applied to the model to recreate the gesture and to estimate missing parameters, for example, occluded fingertips. Joints between each segments and points will often be found using inverse kinematics.

3.4 Gesture Recognition

Recognition of the gesture is the final phase of a complete gesture recognition system. It is the phase in which the data analyzed in the previous stage is recognized as a given gesture. Pavlovic et al. (1997) identified two tasks associated with the recognition process:

- Optimal partitioning of the parameter space;
- Implementation of the recognition procedure.

Optimal partitioning of the parameter space deals with data quantization. In the context of appearance-based models, high-resolution image models require a lot of storage space and make the comparison process slower. On the other hand, too low-resolution image models may lose details important for recognition. A compromise between detail level, storage space and process speed must therefore be done. The choice of the appropriate quantization can only be done through testing of the different possibilities. In the case of model-based and in some appearance-based models that rely on parameter comparison instead of image comparison, a choice must be made on the number of parameters. Parameters must be chosen to help with the recognition process and to discriminate the different classes of gesture. In the case where dynamic gesture is considered, similar choices have to be made to quantize time. Furthermore, parameters should be chosen to be invariant to certain conditions like rotation, translation, and scale in the case of spatial parameters and time instance and time scale in the case of temporal parameters.

Implementation of the recognition procedure can also be seen as an elimination process. In fact, it is the step at which the recognized gesture is compared to a set of plausible gestures and is accepted or rejected. This is particularly important in the case of a sequence of gestures where a gestures do not make sense after another one. In the case of static gestures it is useful to reject impossible hand shapes, angles or finger positions, for example. In computing theory, the finite set of plausible gestures would be called the alphabet and a sequence of gestures would be called a string. The set of rules that are used to determined plausible sequences of gestures is called a grammar (Sipser, 1997). Certain systems concentrate only on recognizing the alphabet or string, while more sophisticated ones also consider a grammar. Once again, in this case, the computational complexity of the recognition procedure is important. Compromises must be done between the model complexity, the richness of the gesture alphabet and grammar, and the computation time. That is why researchers concentrate on specific tasks where it is possible to determine a finite alphabet and a grammar and not on a general model of all the possible human-computer interactions.

3.5 Conclusion

This chapter presented vocabulary words and a methodology to solve general fingering problems. In the subsequent chapters, this methodology will be applied to the guitar fingering problem, first, during preliminary analyses that are presented in chapter 4 and then in a prototype presented in chapter 6. Guitarist left-hand fingering gestures are instrumental gestures and can be categorized as effective manipulative gestures (Cadoz would further categorized these as modification instrumental gesture (Cadoz & Wanderley, 2000)). The fingering gesture recognition process will follow the same step as the general gesture recognition process. A model will need to be set for the temporal segmentation of the gesture in phases (preparation, nucleus, and retraction) and for the spatial representation of the gesture (Hu moments in the preliminary analyses, $(string, fret, finger)$ coordinates in the prototype). The gesture analysis will also follow the same process of localization of the gesture with the determination of a region on the fretboard around the guitarist left hand and of segmentation of the gesture in time and space. The resulting isolated gesture will then be converted according to the chosen model in order to send the appropriate parameters to the gesture recognition module. The recognition process will be based on a knowledge-base of pre-processed chord images in the case of the preliminary analyses and on an alphabet of triplets $(string, fret, finger)$ in the case of the prototype.

Chapter 4

A Preliminary Computer Vision System for Guitar Chords Recognition

This section presents the author's first attempt to solve the guitar fingering problem using computer vision. It is presented as a preliminary study on the use of computer vision to retrieve and analyze guitarist gesture and more precisely guitarist fingering. It is based on already available blocks of the EyesWeb software platform (`http://www.eyesweb.org/`) and its aim was to explore the potential and limitation of these existing resources. Since the system deals with visual input, one important aspect to consider is what and when it has to observe. It may seem simple for humans to focus on the important information from the observation of someone playing guitar, but it is not obvious for a computer. As figure 4.1 illustrates, many types of information can be extracted from the visual image of a guitarist. Consequently, the first thing to do is to find the more appropriate viewpoint for the camera. Then, the region of interest in the image has to be identified. When this is done, the algorithm must know when to look, i.e., to distinguish between stable and transitory parts of the playing process. Finally, the application has to compare the information it receives against the analyzed information stored in its knowledge-base during the training phase. The information must therefore be converted to a format understandable by the algorithm.

4.1 Viewpoint Choice

The choice of a viewpoint is a preliminary step and does not involve the computer directly. This step is based on the hypothesis that the computer will be able to see at least what the human eyes can see, i.e., it is based on human observations of the images captured by the camera. The aim is to find a viewpoint that allows the retrieval of the desired information with the desired degree of accuracy and precision. As it will be shown, these two objectives are conflicting. Accuracy and precision necessitate a close viewpoint focusing on one point of interest, therefore, losing complementary information.

4.1.1 Global View

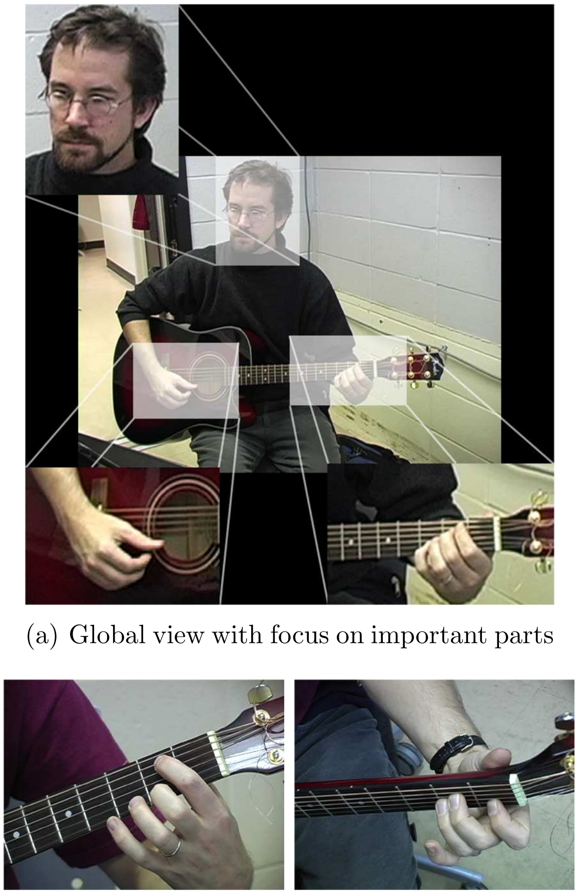
As it can be observed in figure 4.1(a), the global view is ideal for its richness in gestural information. This view allows to see the overall posture of the guitarist and also the action of both hands on the guitar neck and near the sound hole. This view is also rich in information about the expressive content since the face can be observed. Unfortunately, using this view it is impossible to obtain a detailed image of the hands (e.g., fingering or plucking information). To solve the fingering problem, a close-up on the neck region is necessary.

4.1.2 Front View

By focusing on the left hand as seen in figure 4.1(b), it is possible to obtain a more detailed image of the neck. Of course, using this view, right hand, postural and facial gesture information are lost. On the other side, this view provides a clear vision of the fingers in most of the situations, although some occlusion may happen with specific finger positions. Frets and strings are also visible and consequently could be detected to help with the estimation of the finger position on the neck. However, a drawback of this view is that it is not possible to visually know if a string is pressed or not.

4.1.3 Top View

Figure 4.1(c) presents a different perspective on the region observed with the front view. This view presents characteristics similar to the front view, namely a detailed view of the fingers, the possibility to detect the strings and frets and the potential occurrence of the



(a) Global view with focus on important parts

(b) Front view of the left hand

(c) Top view of the left hand

Figure 4.1 Three different views of a guitarist playing captured from a camera on a tripod placed in front of the musician: (a) Global view with focus on different important zones for gesture analysis, namelly facial expression and front view of the left and right hand. (b) Front view of the left hand. (c) Top view of the left hand.

finger occlusion problem. Moreover, this view permits to observe the fingers' proximity to the string; it may therefore be possible to know if the string is pressed or not by the guitarist. Another potential interest of this view is that it is close to the view the musician has of the neck when playing. This may or may not have influence in the computer system but it could be interesting, perhaps, in a system designed for educational purposes.

4.2 Knowledge-Base Creation

In this prototype, the fingering is determined by the evaluation of the global shape of the hand. The system is therefore appearance-based and need to compare the hand shape image of a musician playing chords against previously processed hand shape images of the same chords. To test the efficiency of this technique applied to the guitar fingering problem it is necessary to build a knowledge-base with a small group of selected chords. As seen in figure 4.2, eleven chords grouped in six sets were selected to test the system. The chosen chords are of two types:

- Chords with distinct fingering
- Chords with similar fingering

Distinct chords (set 4: G, set 5: B7 and set 6: G7/D) were chosen to determine the range of hand shapes the system can distinguish. Similar chords (set 2: A and Dm, set 3: C and G7, and set 1: D7, E, Em and Am) were chosen to test the complexity of detail level in the hand shape the system can achieve. The proximity factor used to determine the similarity is the visual shape of the hand as appearing to the human eyes only and not the tablature or any other musical characteristic.

During the training period of the system, a guitarist was asked to play separately each chord twice. One frame of the stable part of each chord was extracted by manual inspection of the video image. These images were then treated following the same procedure the system uses to treat the input images, namely manual selection of the region of interest, threshold of the grayscale image, edge detection of the filtered image, and computation of the edge image Hu moments (Hu, 1962) (figure 4.3 lines 2 and 4). Information about the hand shape is stored in a knowledge-base in the form of a vector of Hu moments. During the recognition phase, this knowledge-base will be consulted by the system to identify chords.

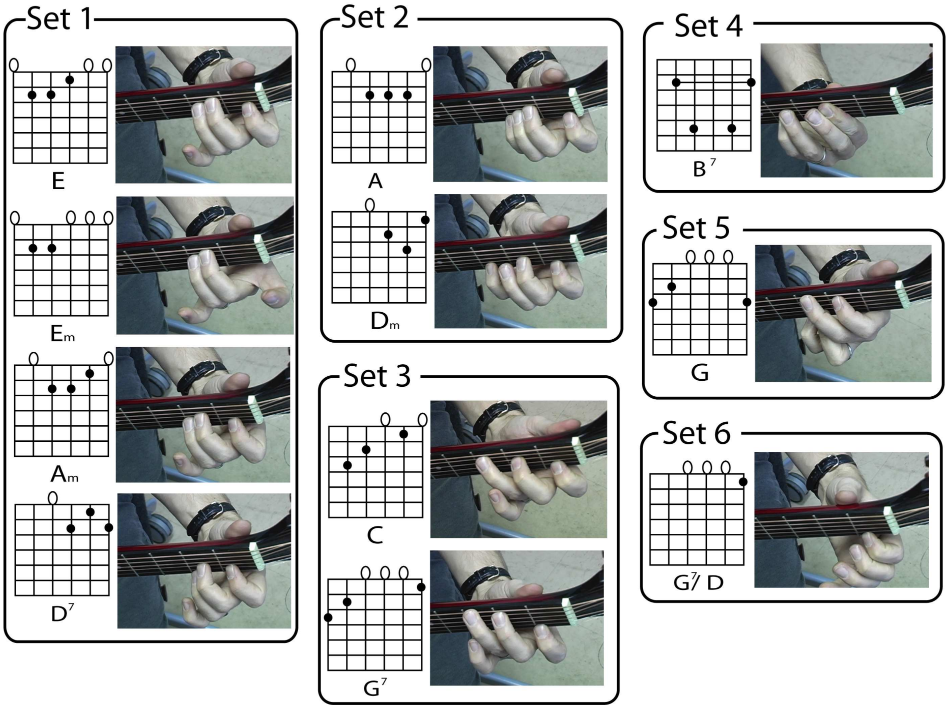


Figure 4.2 Image and tablature of the test chords subdivided in similarity sets

4.3 Algorithm Design

A prototype of the desired system was developed using the EyesWeb platform. The main requirements for the system were:

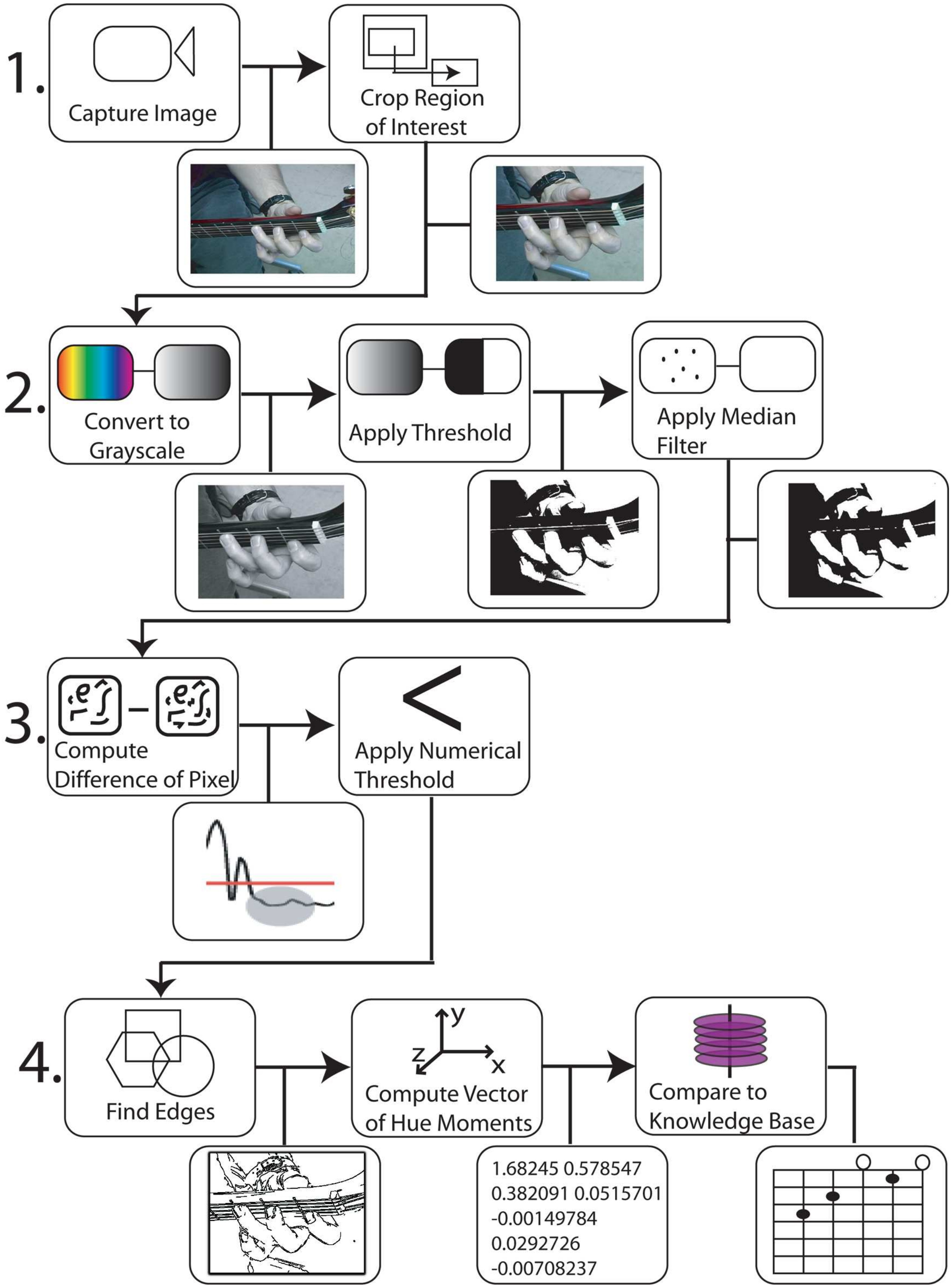
- To process a video signal (live or defered) in realtime;
- To capture the video image of the guitarist playing from a camera on a tripod placed in front of the guitarist;
- To create a reproducible system. In other words, another guitarist using a different guitar and a different camera with a similar setup (camera-to-guitarist distance and camera angle and viewpoint) should be able to use the system with similar recognition level.

From the video signal, the system has to recognize chords that it learned during the training session. The system has to be able to extract hand shape representations of chords and compare these with previously analyzed ones stored in a knowledge-base. As shown in figure 4.6, to perform the analysis, the system has to focus on the hand region in the image and create a shape using Canny edges detection algorithm (Canny, 1986). These edge images then had to be analyzed by computing their Hu moments (spatial modeling, figure 4.4). The system has to distinguish between stable and transitory parts of chords (nucleus versus preparation / retraction) in order to analyze stable parts only (temporal modeling, figure 4.4). Finally, as shown in figure 4.7, the algorithm has to compare Hu moments of the stable part of chords with the one analyzed during the training session. Figure 4.3 summarizes the algorithm.

4.3.1 Gesture Modeling

4.3.1.1 Temporal Modeling

In this system, temporal modeling is necessary to ensure the stability of the gesture recognition process. Assuming that the three dynamic phases of gesture: preparation, nucleus, and transition, are present in the chord playing process, the aim of the temporal segmentation is to prevent the system from analyzing and recognizing chords during the transitory phases of preparation and retraction. The movement analysis is done by computing pixels difference between frames of the video recording of a guitarist playing a sequence of chords



```
1.68245 0.578547
0.382091 0.0515701
-0.00149784
0.0292726
-0.00708237
```

Figure 4.3 Preliminary chord recognition algorithm. Note: For better print-out results, black and white pixels are inverted in the edge image (line 4)

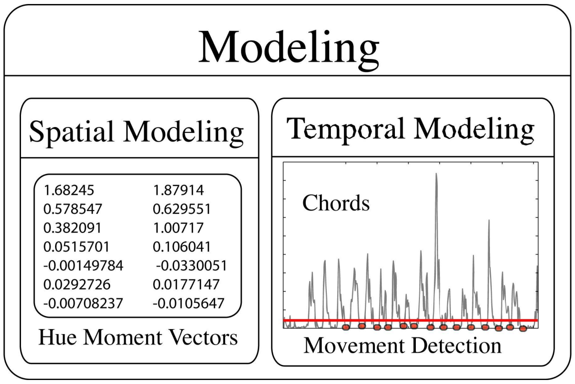


Figure 4.4 Spatial and temporal modeling

to generate a motion curve. The result for a sequence of sixteen chords is displayed in figure 4.5 where the number of pixels that have changed between each frame of the video recording can be observed. In the case studied with this prototype, namely plucked chords, the characteristic low motion of the nucleus can be observed clearly. Consequently, stable parts can be segmented from transitory parts by the application of a threshold. Temporal segmentation is performed early in the algorithm to prevent the unnecessary execution of the subsequent analysis and recognition step. The temporal segmentation step can be seen on line 3 of figure 4.3. Line 4 will be executed only when the motion curve value is lower than the threshold value. As a result, only stable parts of the video are analyzed and tested for recognition.

4.3.1.2 Spatial Modeling

Spatial modeling is the choice of representation of the image data that will be input to the recognition module of the system. In this prototype, the recognition process is performed using the Hu moments representation of the chords. This representation has been chosen considering many factors, namely:

- Storage space:
  - Considering that the amount of different chords that can be played on a guitar is large, the constitution of a complete knowledge-base based on the Hu moments vector of the image (a numerical sequence of seven floating-point numbers) is more economic than a knowledge-base that would rely on the pixel representation

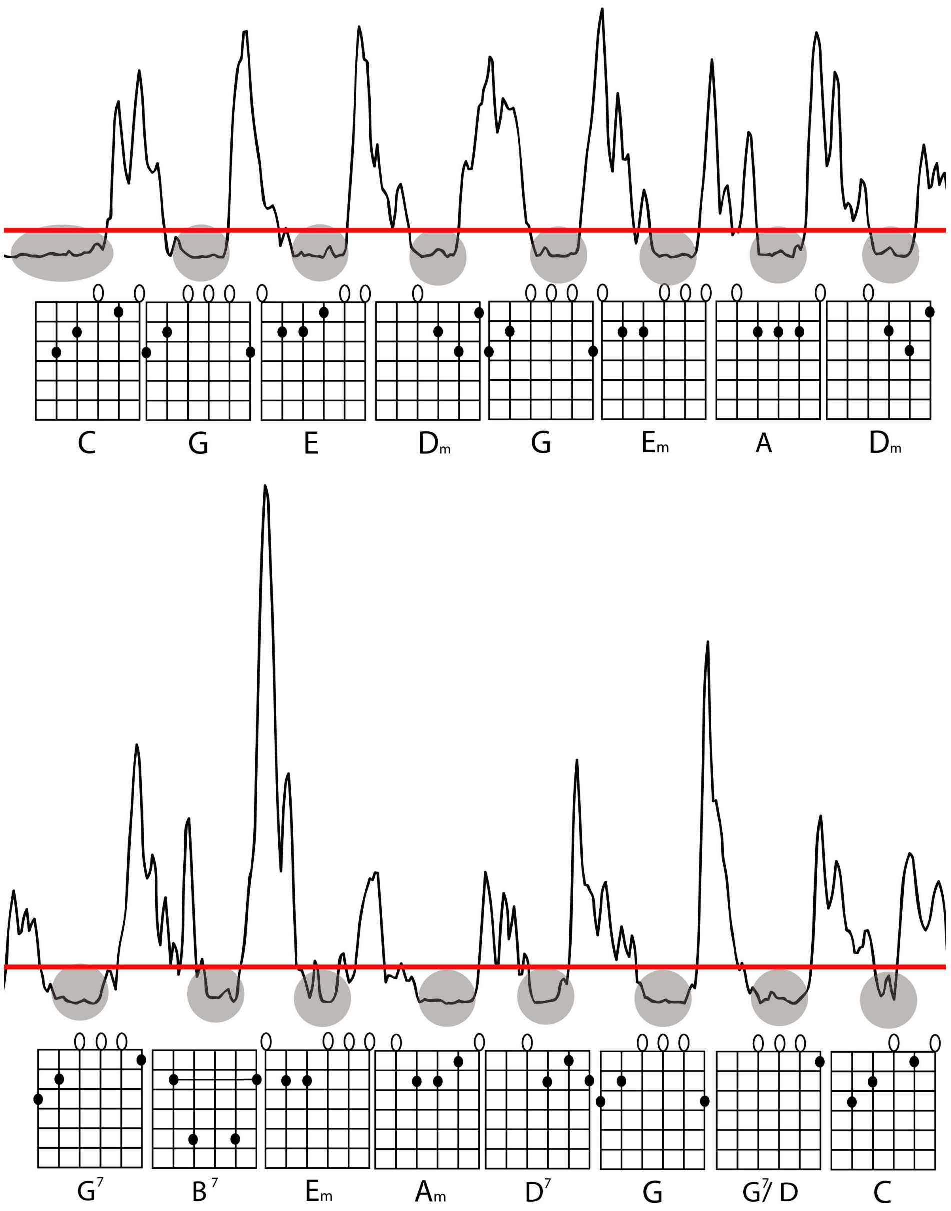


Figure 4.5 Motion curve of the guitarist left-hand playing the sixteen chords sequence

of the chords.

- Computation time:
    - The computation of the difference between two Hu moments vectors is faster than the computation of the correlation between two images (computation time is important to satisfy the real time requirement).
- Invariance to size, rotation, and translation:
    - The Hu moments vector is invariant to size, rotation, and translation, these three characteristics are important to satisfy the reproducibility requirement of the system:
        * Invariance to size, rotation, and translation is assumed to allow the system to be tolerant to small variations of the camera angle and position and musician to camera distance, and to work with different sizes of guitars and hands.
        * Invariance to rotation and translation are also important since the camera is not moving together with the guitar neck, therefore, due to the musician ancillary movements, the neck of the guitar does not appear at the exact same place at different moments of the playing process. However, invariance to translation might cause problems for chords C and G7 and rotation invariance might cause problems for chords D7 and E because to human eyes they look like a horizontal translation and a clockwise rotation of the second chord with respect to the first one (figure 4.2) .

4.3.2 Gesture Analysis

4.3.2.1 Feature Extraction

Localization

The localization phase consists on determining the gesture region, usually called the region of interest (ROI). In this prototype, the ROI has been determined by manual inspection of the video sequence. Since the guitarist and camera position are relatively stable in the video sequence, a fixed rectangle mask is applied around the region where the hand appears (figure 4.3, line 1, step 2). The dimensions of the rectangle are chosen to minimize the noise

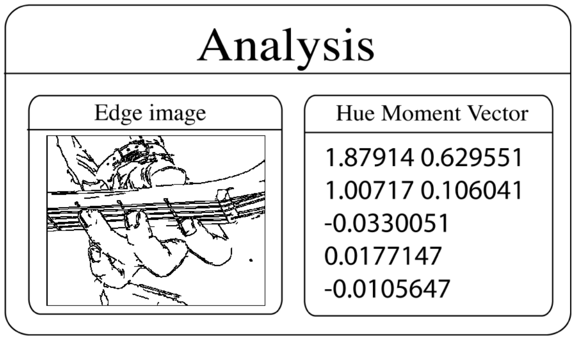


Figure 4.6 Analysis of the chord using Hu moments on the edges image. Note: For better printout results, black and white pixels are inverted in the edge image

created by surrounding objects and to maximize the preservation of the integrity of the hand shape (results of the localization step during the training phase can be observed in figure 4.2).

Segmentation

Once the ROI has been defined, the gesture must be “extracted” from the image. In this prototype, the segmentation was simplified by recording the guitarist on a constant, light background. This way, the segmentation could be done by applying a threshold on the video image and by filtering the noise using a median filter (figure 4.3, line 2).

4.3.2.2 Parameter Estimation

The last step of the analysis process consists in converting the segmented input gestures into the appropriated format for the recognition process. In this case, the chosen parameter space is a seven number vector obtained by applying the Hu moments on an edge image of the chord (figure 4.4). This method is inspired by the work of Paschalakis and Lee (1999) where Hu moments are used to recognize different objects from the silhouette image of their shape. The conversion to edge image is done using the Canny edge detection algorithm (Canny, 1986) on the segmented image. This process and the conversion to Hu moments vector can be seen on the two first steps of line 4 of figure 4.3.

4.3.3 Gesture Recognition

As it can be seen in figure 4.7, the recognition step is simple and consists of a comparison between the vector obtained during the analysis step and the vectors obtained during the system training. A difference between the vector of the analyzed image and all the vectors in the knowledge-base is computed. The system chooses the closest vector as the more probable chord. Preliminary experimental results suggest that the system should confirm the recognition and consequently output a result only if the confidence level for that vector proximity is higher than seventy percent. Furthermore, results also suggest that the system is more stable if the chords are recognized in at least twenty consecutive frames. With a frame rate of thirty frames per second this implies a minimum delay of two-thirds of a second to recognize a chord; it is important to note, however, that this delay does not include the computation time of the vectors difference that grow up linearly with the number of chords in the knowledge-base.

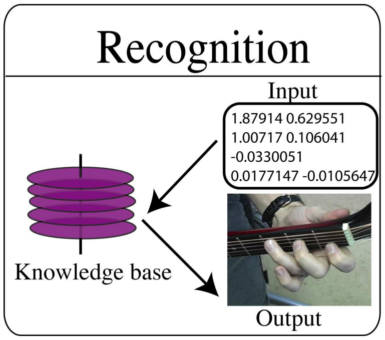


Figure 4.7 Chord recognition by comparison between the Hu moments of the image and Hu moments of previously analyzed images

4.4 Results

Figure 4.8 presents the output of the system. Played chords are indicated on the x-axis. The chords are the sixteen first chords of *La complainte du phoque en Alaska* (Rivard, 1991) played with a guitar pick using the following pattern: base, down pluck, down pluck. On figure 4.8, it can be observed that there is more “space” between the first C and the other chords. That is because the first C is played for two bars and all the subsequent chords are played for one bar as indicated in the score. The y-axis is the index of the chords in the knowledge-base. The columns are the indexes output by the recognition system. The width of the columns indicates the duration of time a chord is identified. Consequently, the output is zero during transition phases and when no chord could be identified. The correctly recognized chords are circled and the ones that were confused with another chord in their similarity set are squared. The test was performed five minutes after the training with the same musician and the same camera setting. Even though this is an optimal situation the recognition level was low. During the sixteen chords playing test, a correct recognition happened four times and of these four recognized chords, three were also recognized at places where they were not played. The number of recognized chords is augmented if the recognition confidence level and the number of consecutive recognitions are lowered but at the price of introducing multiple recognitions of a unique chord. This phenomenon can be observed twice in figure 4.8: with the first C where the chord is wrongly recognized as G7/D twice, and with the last C where the chord is recognized twice as G and once as G7. If the confidence level and number of consecutive recognitions are augmented, the number of recognized chords is reduced. The seventy percent confidence level and the recognitions in twenty consecutive frames is consequently the best setting for that test.

Another important aspect of the results is that the recognition errors are not as initially expected. Confusion with a chord of the same similarity set happened only twice, while confusion with chords of random sets happened six times. This implies that the Hu moments algorithm does not categorized the hand shapes as expected by a human analyzer. Consequently, this implies that it may not be an appropriate classification method for that kind of task.

4.5 Conclusion

These preliminary analyses presented an attempt to use the global shape of the left hand of a guitarist to recognize the chords he is playing. Different camera views were evaluated and the top view (figure 4.1(c)) was retained for its interesting characteristics with respect to the problem, namely a detailed view of the fingers, the possibility to detect the strings and frets, and the possibility to observe the finger-string proximity. The hand was segmented from the rest of the image, first by the manual selection of the region of interest, a rectangular region around the hand (figure 4.3, line 1, step 2), then by applying the Canny edge detection algorithm on the threshold image of that region (figure 4.3 and figure 4.6). Time

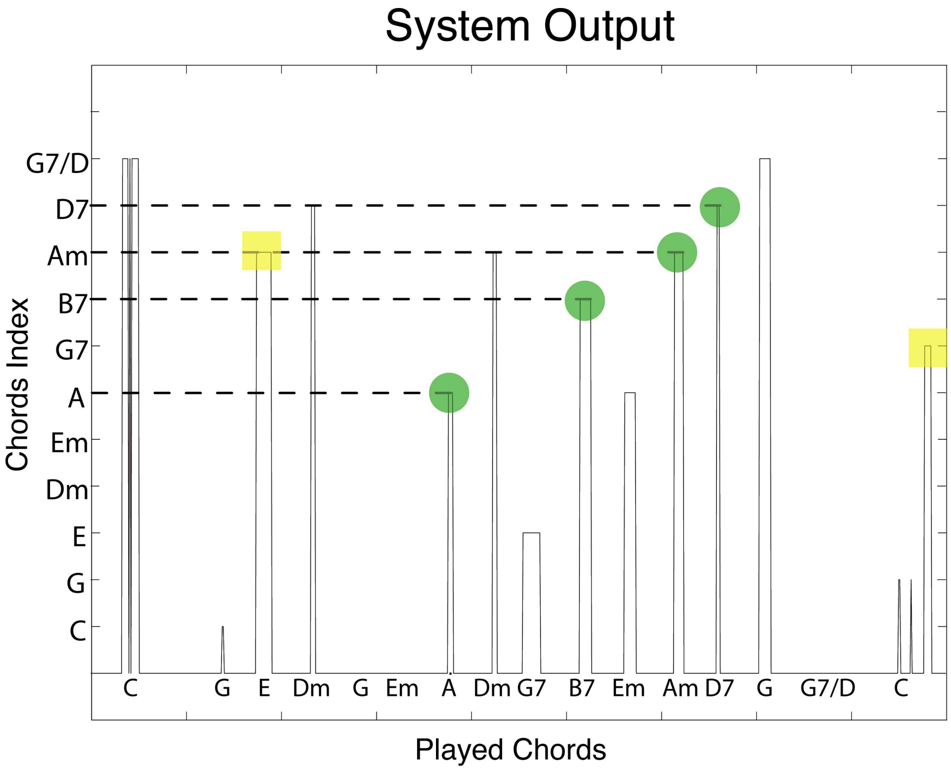


Figure 4.8 Output of the chords recognition algorithm. The x-axis is the sequence of played chords. The y-axis is the index of the chords in the knowledgebase. The columns are the output of the recognition system during the sequence. The width of the columns is the duration of the identification. Circles are corrected matches. Squares are matches with a chord of the same similarity set.

segmentation was also applied in order to evaluate only stable images of chords. In other words the transitory phases of preparation and retraction were eliminated and only the nucleus phase was evaluated (figure 4.4). The images obtained after that process were converted to vectors by applying the Hu moments algorithm (figure 4.4). They were then compared to previously learned chords in the knowledge-base (figure 4.7). A chord was recognized when a chord of the knowledge-base matched it with a sufficiently high proximity factor.

Preliminary tests were performed with a knowledge-base of eleven chords learned five minutes before the test with the same setting of guitar, camera, and musician. Although this is an optimum evaluation scheme, the system succeeded to recognize chords in only twenty-five percent of the cases. This is an extremely low recognition rate but these preliminary analyses permitted to identify some problems in the assumption of the system:

1. The Hu moments do not seem to categorize the hand shapes as expected.
2. Using an appearance-based method relying on a knowledge-base recognition mechanism limits the system to previously learned material.
3. Using the global shape of the hand limits the system to the recognition of chords.
4. Recognition time grows with the knowledge-base size.

4.5.1 Problems Related to the Choice of the Hu Moments Vector Representation

Prior to the test, the chords were divided in different similarity sets, chords from the same set, therefore with similar visual shape were expected to be confused by the system. However, test results show that it was not the case. The system confused chords with others of apparently random sets. This might be due to many reasons including:

- A wrong shape representation (Canny edge image). Canny was chosen in this experiment in order to keep information on both the guitar neck and the hand, but Paschalakis and Lee (1999) used silhouette shapes. It is possible that the Hu moments work better with silhouette shapes than with edge shapes.
- A wrong vector representation (Hu moments). As explained previously, the Hu moments are invariable to size, rotation, and translation, although it was expected to be an advantage, it is possible that this also has a negative impact on the vector representation of the chords.

- An under estimation of the impact of ancillary gestures and remaining background elements.

Consequently, potential solutions for the representation problem include:

- Use another shape representation, for example silhouette;
- Use another vector representation, for example image geometric moments or image eigenvectors;
- Store vector representation of more than one image for each chord, therefore accounting for rotation or translation of the neck position in the image;
- Ameliorate the segmentation of the gesture from other elements;
- Limit the impact of ancillary gestures by affixing the camera to the neck.

4.5.2 Problems Related to the Choice of an Appearance-based Method Relying on a Knowledge-Base Recognition Mechanism

The three last problems are all related with the choice of an appearance-based model to represent the gesture. During the recognition process, appearance-based models need to compare an actual representation of an image with previously observed representation of similar images stored in a knowledge-base. In this case, the chosen image representation was the Hu moments of the edge image of the hand on the guitar neck. This scheme implies that for a chord to be recognized, at least one Hu moments vector representing it should be in the knowledge-base. Consequently, the more chords the system can recognized the larger the knowledge-base has to be and the longer the recognition process will take since a difference of vectors has to be computed for each vector present in the knowledge-base. In addition, by considering the global shape of the hand, the system is limited to the recognition of chords since it cannot consider the individual actions of each finger of the left hand. Solution for these three problems include:

- Using another recognition mechanism, for example neural networks or Hidden Markov Models (HMM);
- Using another modeling method, for example fingertips-based models or three dimensional models;

4.5.3 Prototype Specifications

In conclusion, even if the preliminary analyses were not successful in retrieving fingering information, they provided insights about how to visually acquire guitar fingering information. Therefore, the main specifications for a fingering recognition system are:

1. Focus on effective gestures by further reducing the presence of ancillary gestures and background elements.
2. The use of a representation that considers the action of individual fingers more precisely.
3. The use of a recognition mechanism that eliminates the burden of a knowledge-base and that is therefore not limited to previously learned material.

The first specification can be achieved using the guitar mount as it will be presented in section 6.1.1.2. In order to fulfill the other specifications, three studies were conducted. A first study, presented in chapter 5, evaluated four different finger-localization algorithms. A second study examined the use of the linear Hough transform for string and fret detection (chapter 6, section 6.1.1.2) and a third one explored movement segmentation (chapter 6, section 6.1.1.1). Finally, a prototype respecting these specifications has been developed and will be presented in chapter 6.

Chapter 5

General Finger-Localization Algorithms Using Eyesweb

This chapter presents the study on finger-localization algorithms performed by the author during an internship at InfoMus laboratory, Università of Genova (Burns & Mazzarino, 2006). This study was necessary to achieve the requirements for a guitar fingering retrieval system outlined in chapter 4. Effectively, one characteristic of general finger-localization algorithms is that they detect the position of individual fingers, consequently satisfying the requirement to consider the action of each individual fingers.

In order to avoid the problem of complex and not reproducible high-cost systems, this study focuses on two-dimensional systems using a single simple video camera. Algorithms using projection signatures, the circular Hough transform, and geometric properties have been chosen and are compared to an algorithm using color markers. Color markers are used solely as a reference system to evaluate the accuracy and the precision of the other algorithms, the presence of markers being a non-desirable constraint on the user of such a system. All the algorithms have been implemented in EyesWeb using the Expressive Gesture Processing Library (Camurri, Mazzarino, & Volpe, 2004) together with newly developed blocks (available in EyesWeb 4).

The algorithms presented in this study are inspired by the research on tabletop applications (Koike, Sato, & Kobayashi, 2001; Letessier & Brard, 2004). These kinds of applications are often limited to the use of one finger instead of using the information that can be provide by detecting the position of all fingers. Furthermore, these applications often use specific and expensive hardware (infrared camera, for example). In this paper we suggest alternative methods that can work with simple hardware, such as a low-cost webcam. We use methods that were traditionally used in static pose identification (e.g., contour and signature) to do fingertips localization. The use of the Hough transform, on the other hand, was inspired by research in 3-dimensional tracking (Hemmi, 2002), but also by some of the previously mentioned tabletop applications. These applications use the specific geometric shape of the fingertip with various templates matching algorithms to locate fingers.

The first section of this chapter briefly describes and illustrates the EyesWeb implementation of the four algorithms. Next, the test procedures are explained. The third section presents the results obtained from each algorithm during the tests. Finally, the chapter concludes with a comparative discussion of the potential uses of the different algorithms.

5.1 Methods

All the algorithms were evaluated in EyesWeb using 640x480 pixels RGB two-dimensional images of a hand performing different finger movements on a flat surface. The videos were recorded by a single fixed camera with a frame rate of 25fps (frame per second), fixed gain and fixed shutter. The tests were run on a Pentium 4 3.06GHz with 1Gb of RAM under Windows XP operating system. In order to test the algorithms, the problems of finding the region of interest and of eliminating complex backgrounds were reduced by shooting only the hand region on a uniform dark background. The second line of figures 5.1, 5.2, and 5.3 illustrates the segmentation process. In this simplified case, it consists of converting the image to gray-scale, applying a threshold to segment the hand from the background (using the fact that the hand is light while the background is dark), and filtering with a median filter to reduce residual noise.

5.1.1 Projection Signatures

Projection signatures, are performed directly on the resulting threshold binary image of the hand. The core process of this algorithm is shown on line 3 of figure 5.1 and consists of adding the binary pixels along the hand angle, which must be know previously. A low-pass filter is applied on the signature (row sums) in order to reduce high-frequency variations that create many local maxima and cause the problem of multiple positives (more than one detection per fingertip). The five maxima thereby obtained should roughly correspond to the position of the five fingers.

5.1.2 Geometric Properties

The second algorithm is based on the geometric properties and, as shown on line 3 of figure 5.2, uses a contour image of the hand on which a reference point is set. This point can be determined either by finding the center of mass of the contour (barycenter or centroid) or by fixing a point on the wrist (Yörük, Dutağaci, & Sankur, 2006). Euclidean distances from that point to every contour points are then computed, with the five resulting maxima assumed to correspond to the finger ends. The minima can be used to determine the intersections between fingers (finger valleys). The geometric algorithm also requires filtering in order to reduce the problem of multiple positives.

5.1.3 Circular Hough Transform

The circular Hough transform is applied on the contour image of the hand but could as well be applied on an edge image with complex background if no elements of the image exhibit the circular shape of the fingertip radius. The circular Hough transform algorithm uses the fact that the finger ends and the finger valleys have a quasi-circular shape while the rest of the hand is more linearly shaped. In this algorithm, circles of a given radius are traced on the edge or contour image and regions with the highest match (many circles intersecting) are assumed to correspond to finger ends and valleys (this process is illustrated on line 3 of figure 5.3). Searched fingertips radius can be set manually or determined by an algorithm using the palm radius to fingertip radius proportion as an estimate (Chan, 2004; Yörük et al., 2006; Hemmi, 2002). The circular Hough transform can find both finger ends and valleys but, as opposed to the geometric algorithm, does not output them in two distinct sets. Furthermore, the circular Hough transform requires filtering to eliminate false positives (detected regions that are not finger ends or valleys) that frequently appear between fingers. As illustrated in line 4 of figure 5.3, this can be done efficiently for finger ends by eliminating points that are found outside the contour image. The inconvenience is that the set of discarded points contains a mix of finger valleys and false positive that cannot be sorted easily.

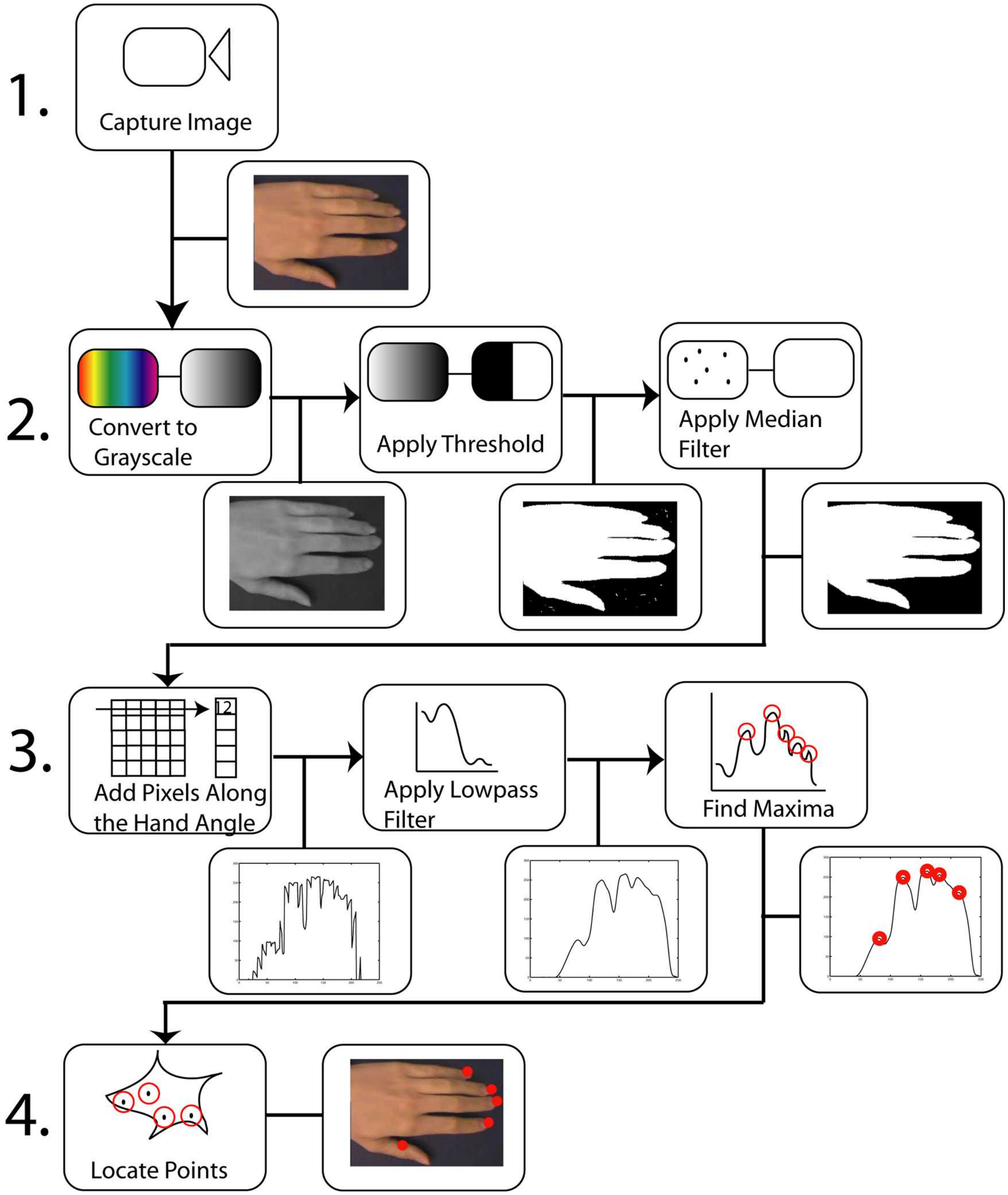


Figure 5.1 Finger localization using the projection signature algorithm

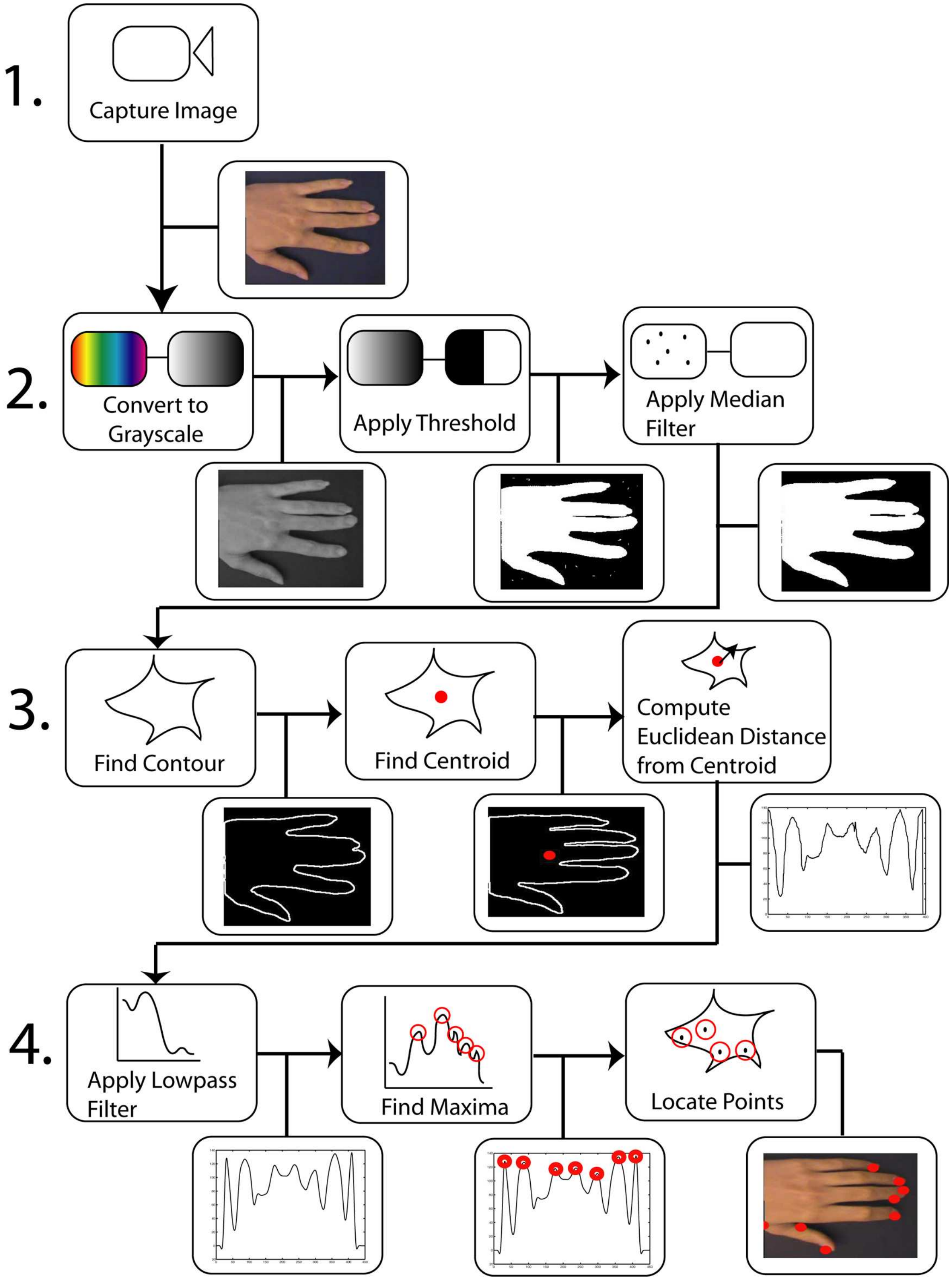
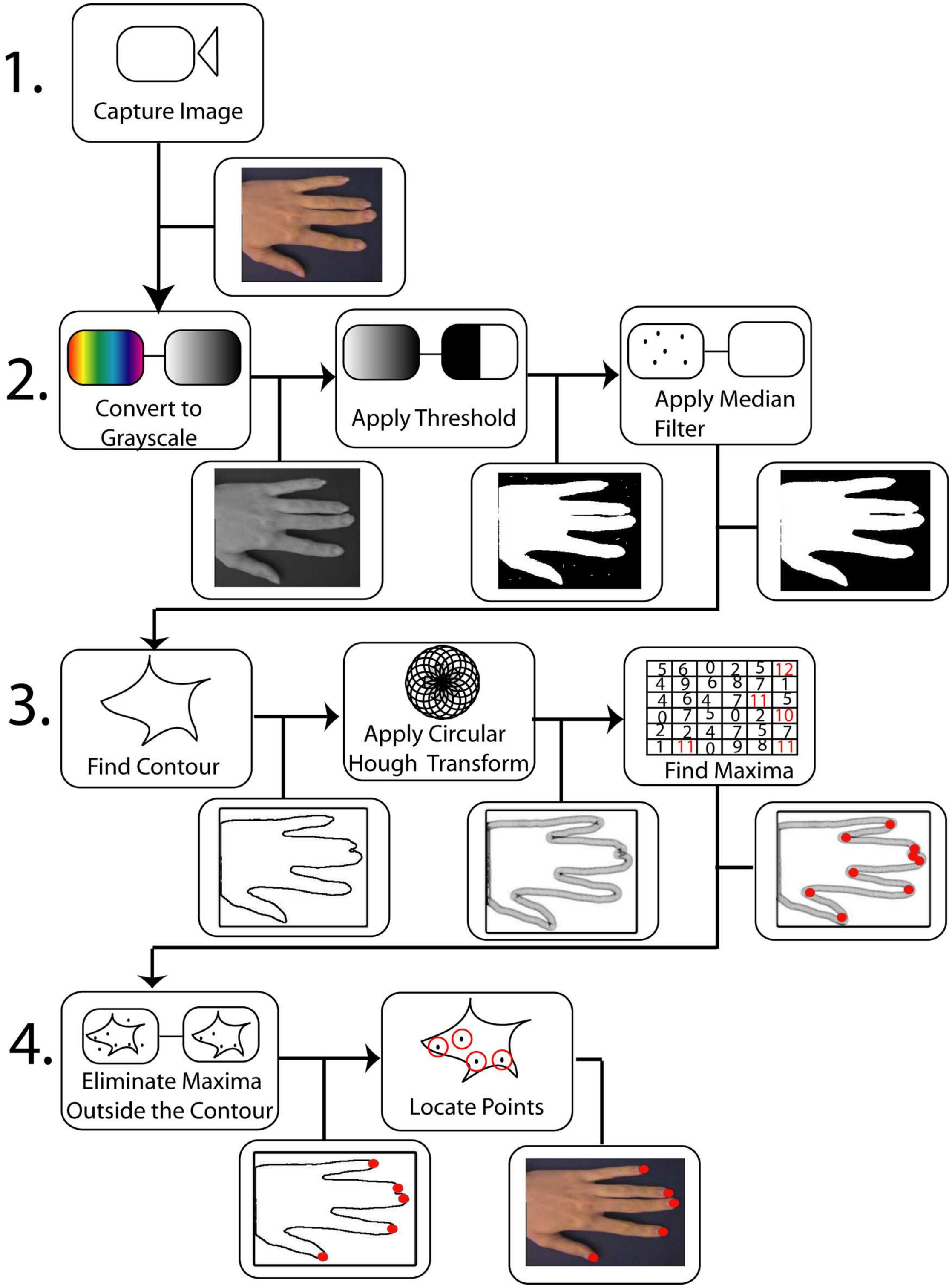


Figure 5.2 Finger localization using the euclidean distance algorithm



5	6	0	2	5	12
4	9	6	8	7	1
4	6	4	7	11	5
0	7	5	0	2	10
2	2	4	7	5	7
1	11	0	9	8	11

Figure 5.3 Finger localization using the circular Hough transform algorithm. Note: For better printout results, black and white pixels are inverted in the output images of lines 3 and 4

5.1.4 Color Markers

While the three previous algorithms rely only on the hand characteristics to find the position of the fingertips, the marker algorithm detect color markers attached to the main joints of the fingers. Each color is detected individually using color segmentation and filtering as illustrated in line 2 of figure 5.4. This permits the identification of the different hand segments. The marker colors should therefore be easy to detect and should not affect the threshold, edge or contour image of the hand. Respecting these constraints makes it possible to apply all algorithms to the same video images and compare each algorithm's degree of accuracy and precision with respect to the markers.

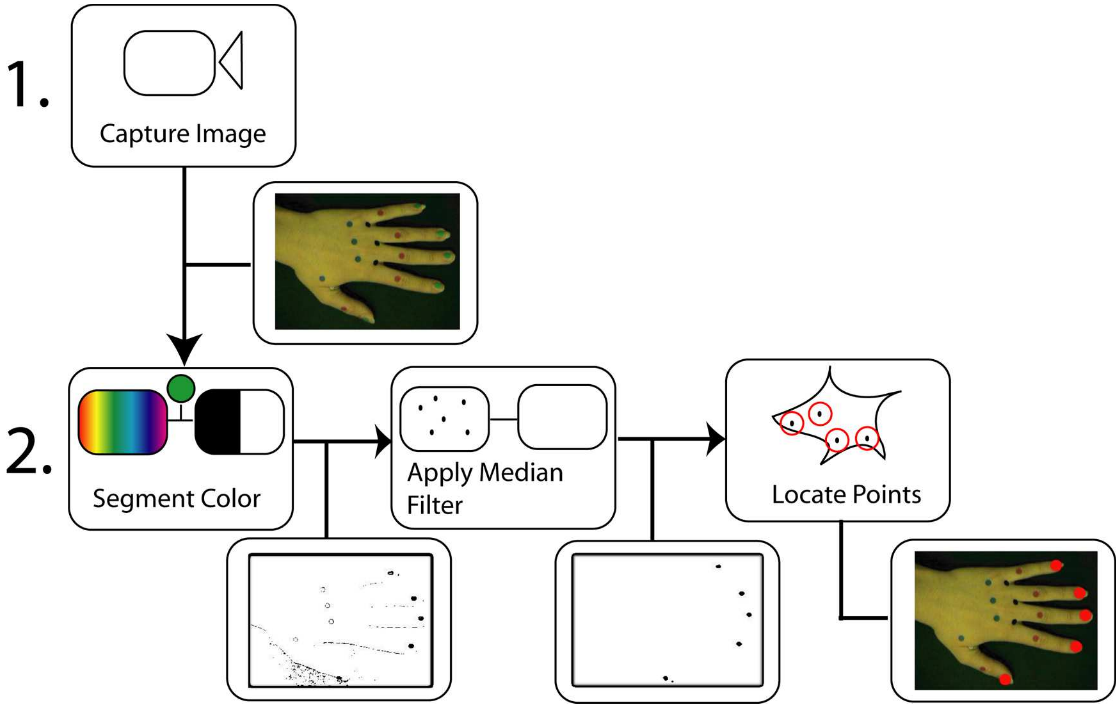


Figure 5.4 Finger localization using the corlor markers algorithm. Note: For better printout results, black and white pixels are inverted in the output images of line 2

5.2 Tests

5.2.1 Accuracy and Precision

Accuracy and precision are important factors in the choice of a finger-localization algorithm. The accuracy and precision of the different algorithms were determined with respect to the result obtained from the evaluation of the marker positions. To evaluate the accuracy and precision of the algorithms, the coordinates of 4 joints on each finger were detected by applying the color markers method (figure 5.4). Coordinates obtained with the three other algorithms were then related to the first set. The Euclidean distance between the marker and the closest point of each algorithm was computed. The accuracy of an algorithm can be determined by its distance from the marker. A curve close to zero denotes an accurate algorithm. The precision of an algorithm can be determined by observing the shape of the curve. A precise algorithm will exhibit an almost flat curve. Figure 5.5 presents the results obtained by detecting the position of the tip of the small finger using each of the three algorithms. The values are compared to a marker placed at the center of the tip of the small finger.

It can be observed that both the circular Hough transform and the geometric properties algorithm are precise algorithm since the distance between the marker and the point they return is almost constant. However, the circular Hough transform seems to be more accurate than the geometric properties. The average distance to the marker is really close to zero in the case of the circular Hough transform, but is approximately ten pixels in the case of the geometric properties. The difference is mainly due to the fact that the geometric properties algorithm detects the extremity of the finger while the circular Hough transform finds the center and that is where the markers are placed. In the case of the projection signatures, the detection of the fingers is robust but rough: the algorithm can only find the fingers and not a specific region of the finger like a tip or a valley. It can be observed in figure 5.5 that for an almost flat angle of the small finger, the accuracy is near twenty pixels (frame 0 and 160), for a small angle (between frame 50 and 160) it is approximately thirty pixels, and can go over a difference of sixty pixels for a large angle (after frame 160). This is due to the computation method, when the finger is making an angle, the end of the section that is in straight line with the palm will create a maximum and not the real finger end. This algorithm is consequently efficient only to find fingers or finger ends when the fingers are not angled. This algorithm is therefore neither accurate nor precise.

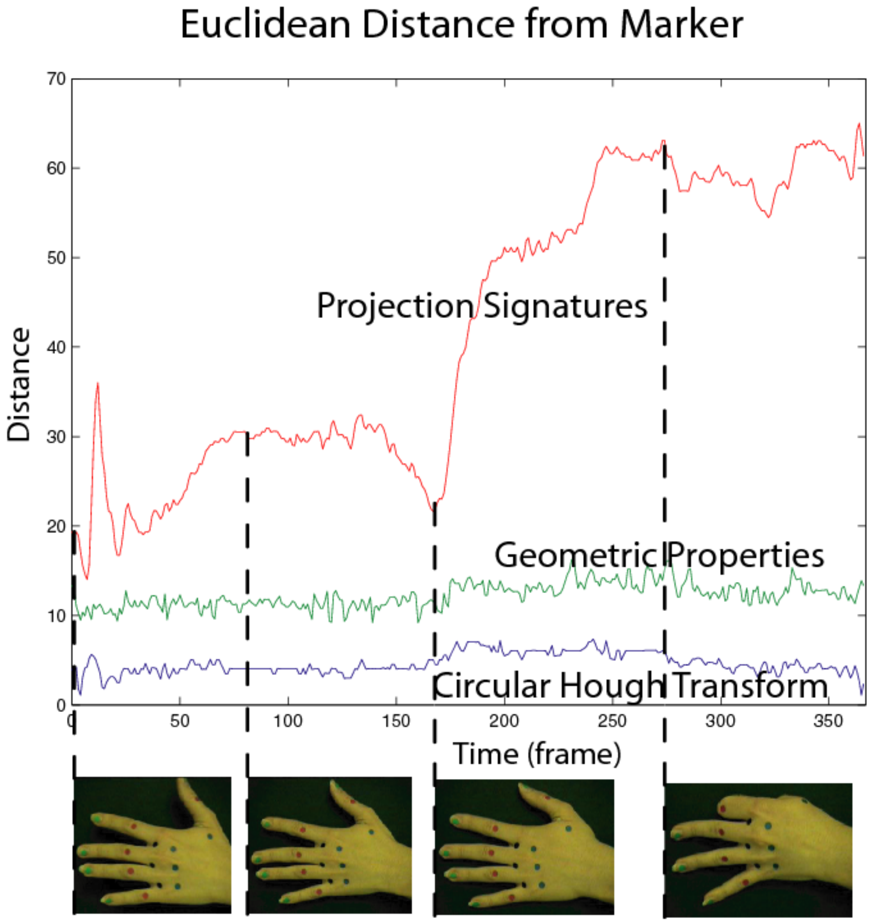


Figure 5.5 Accuracy and precision curves for each algorithm compared with the markers algorithm

5.2.2 Latency and Resources Usage

The latency of each of the algorithms is determined by computing the delay between the evaluation of a frame and the output of its results. If the output rate is the same as the input rate (expressed in terms of the amount of time lapse between two input frames), no significant delay is generated by the evaluation part of the algorithm. In order to know the processing rate and the resource usage of the evaluation algorithm, all screen or file outputs were turned off. All algorithm were tested in EyesWeb in the condition described in section 5.1. Table 5.1 displays the CPU (central processing unit) usage for each algorithm. The range is the observed minimum and maximum CPU usage percent throughout the duration of the test. The mode is the most frequently observed percentage.

Input Rate	**Algorithms**	**CPU Usage Range**	**CPU Usage Mode**	**Output Rate**
33 ms	Projection Signatures	10-18%	15%	33 ms
	Circular Hough Transform	38-77%	55%	
	Geometric Properties	16-45%	30%	

Table 5.1 CPU usage of the three methods

Table 5.1 shows that all the algorithms can be used in real time since no significant latency has been observed. Projection signature is extremely easy on computer resource with a mode of 15% of CPU usage and peaks ranging between 10 and 18%. Geometric properties is a bit more demanding with a mode of 30%. The poor performance of the circular Hough transform is probably due to the usage of the traditional algorithm (Duda & Hart, 1972; Schulze, 2003) that requires a lot of computation and storage for the accumulator cells, more modern implementations using probabilistic and heuristic approaches to optimize the algorithm performance exist (Illingworth & Kittler, 1988) and are known to detect circles with the same degree of accuracy and precision.

5.3 Results and Discussion

We tested the four algorithms with video recordings of the left and right hand of 5 users (3 females and 2 males, all adults). These preliminary tests were performed with recordings of two-dimensional finger motion on a flat surface and not on the guitar in order to highlight the general characteristic of the finger-localization algorithms. Results of these preliminary tests were coherent among all users and are qualitatively summarized in Table 5.2.

	Projection Signatures	**Geometric Properties**	**Circular Hough Transform**	**Color Markers**
Locates fingers	+	+	+	+
Locates fingertips	-	0	0	+
Locates finger ends and valleys	-	+	+	+
Distinguishes between finger ends and valleys	-	+	0	+
Works with complex background	-	-	0	0
Works in real time (low latency)	+	+	+	+
Computer resources usage	+	+	-	+
Accuracy	-	+	+	+
Precision	-	+	+	+
Works with unknown hand orientation	-	+	+	+
Works with unknown fingertips radius	+	+	-	+

Table 5.2 Finger-localization algorithms characteristics comparison table (+ → good to excellent, 0 → neutral to good, - → poor to neutral

All the presented algorithms have succeeded, in various degrees, in detecting each finger. The projection signatures algorithm can only roughly identify a finger, but the circular Hough transform and geometric properties algorithms can find both finger intersections and finger end points (it is important to note that in the case where fingers are folded, the end points do not correspond to the fingertips). The geometric properties algorithm outputs intersections and extremities in two distinct sets, but the circular Hough transform algorithm cannot make this distinction. The marker algorithm is the only one that can distinguish the various joints of the finger when different colors are used.

The projection signatures and geometric properties algorithms need a strong segmentation step prior to their application. The circular Hough transform, when combined with edge detection instead of contour, can work in complex environments, but some confusion can occur if other circular shapes of the size of the fingertip radius are present. Color markers can be used in complex backgrounds if the colors are properly chosen but are sensitive to light variation.

At 25fps all the algorithms output results without any significant delay; the input and output rate is the same. However, the circular Hough transform algorithm is much more demanding on CPU usage than the others. This characteristic might limit its use when it is combined with pose and gesture recognition algorithms. The geometric properties and the circular Hough transform algorithms have similar and acceptable accuracy and precision values. The projection signatures algorithm cannot be used if these two characteristics are important.

The projection signatures algorithm can only be used in a controlled environment where the hand orientation is known and where finger angles don't vary to much from the straight line. The circular Hough transform algorithm needs previous knowledge of the fingertip radius or the palm radius. It can work in an environment where the distance from the video camera will change only if a method to estimate these radii is attached to it (Chan, 2004). The geometric properties algorithm does not need any prior knowledge to be performed.

5.4 Conclusion

This chapter presented three algorithms to locate fingertips in two-dimensional video images. These algorithms have been compared to one another and evaluated with respect to a fourth algorithm that uses color markers to locate the fingertips. All the algorithms were implemented and tested in EyesWeb. Results relative to the precision, accuracy, latency and computer resource usage of each of the algorithms showed that geometric properties and circular Hough transform are the two algorithms with the more potential. The circular Hough transform should be preferred when a clean segmentation from the background is impossible while the geometric properties algorithm should be used when the fingertip radius is unknown and when information on both the finger ends and valley is required. Projection signature can be used as a fast algorithm to roughly obtained finger position. The choice of an algorithm should, therefore, depend on the application and on the setup environment. Future users should refer to the algorithms' characteristics and constraints in table 5.2 to chose the appropriate one. It is also important to note that in this study, the algorithms were tested alone, in a controlled environment, and on a general two-dimensional task. Consequently, the choice of an algorithm can also be influenced by the system in which it is supposed to work. As an example, the segmentation algorithm used in the pre-processing step and the pose or gesture algorithm used in the post-processing step can create constraints that will dictate the use of a specific finger-localization algorithm. The choice of the appropriate finger-localization algorithm for the guitar fingering problem will be explained and related to the general algorithms characteristics outline in this chapter in chapter 6.

Chapter 6

Development of a Vision Based System: A Prototype

This chapter presents the prototype designed to fulfill the requirements for a fingering recognition system highlighted by the preliminary analysis. The focus on effective gestures is partially realized at the hardware level by affixing the camera to the guitar neck, thereby eliminating the motion of the neck caused by ancillary gestures. Elimination of background elements is done by selecting a strict region of interest (ROI) around the neck and by applying a background subtraction algorithm on the image. Movement segmentation is performed by finding minima in the motion curve, obtained by computing the pixel difference between each frame. The action of each individual finger is considered by using one of the finger-localization algorithms described in chapter 5. The details of the algorithm are shown in figure 6.11.

6.1 Algorithm Design

6.1.1 Gesture Modeling

6.1.1.1 Temporal Modeling

Movement segmentation is essential in order to detect fingering positions during the playing sequence. Furthermore, in order to save computer resources, this segmentation is done early in the algorithm so that the subsequent analysis steps are performed only on significant finger positions (see figure 6.11 line 3). Movement segmentation is used to separate

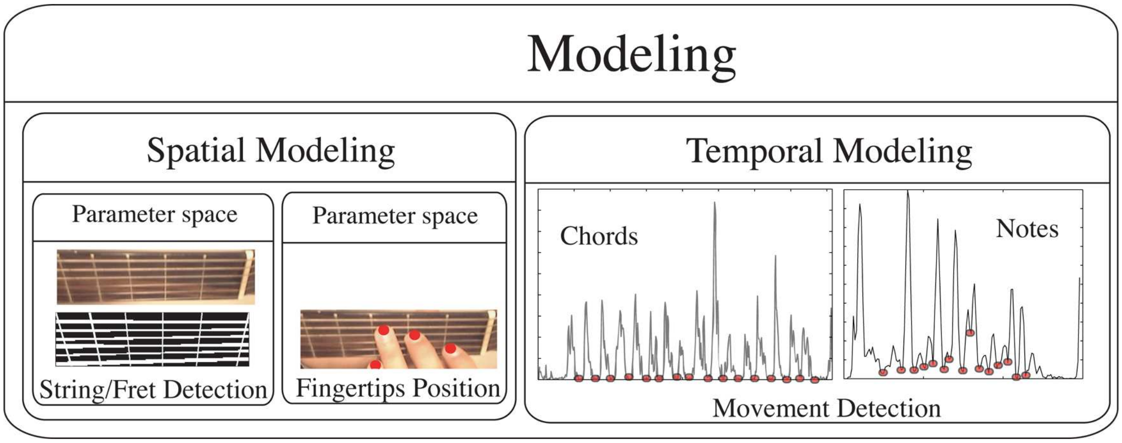


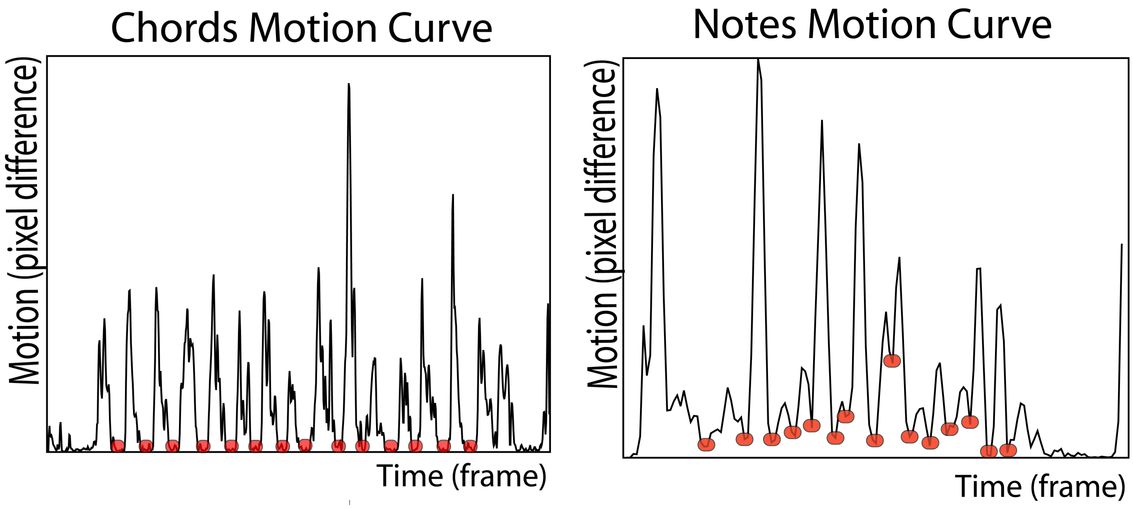
Figure 6.1 Modeling of the gesture: the parameter space is the fingering position related to the string and fret coordinates

the nucleus phase of the gesture from the preparation and retraction phases. Assuming that the temporal division of empty-handed gestures in three phases (preparation, nucleus, retraction) is correct and consistent (Pavlovic et al., 1997), a similar structure can be used to analyze instrumental gestures.

In the preliminary analyses, movement segmentation was done by applying a threshold on the motion curve (figure 6.2(a)) generated by the computation of the pixel difference between each frame. The characteristic lower velocity phase of the nucleus was easily detected between each chord. However, in other playing situations, such as when playing a series of notes, the separation between the movement transitory phases and the nucleus is not that clear (figure 6.2(b)). This is due to a phenomenon called "anticipatory placements of action-fingers" that has been studied in violin (Baader, Kazennikov, & Wiesendanger, 2005) and piano (Engel, Flanders, & Soechting, 1997). In these cases, the preparation phase of other fingers occur during the nucleus of the action-finger. Thus the motion is not serial, and consequently the global motion curve does not exhibit clear global minima as in the case of simple chord fingerings. However, local minima can still be observed and detected, and are assumed to correspond to the moment the note is trigged by the right hand. Local minima are found by computing the second derivative of the motion curve.

6.1.1.2 Spatial Modeling

In this case, two elements of the image are important:



(a) Motion curve of a guitarist playing chords (b) Motion curve of a guitarist playing notes

Figure 6.2 Motion curve of the left hand of a musician playing musical excerpts: (a) Motion curve of a guitarist playing chords (b) Motion curve of a guitarist playing notes

1. Fingertip positions,
2. String and fret coordinates.

The Choice of a Finger-Localization Algorithm

From the four finger-localization algorithms presented in chapter 5 only two could be applied to the guitarist fingering problem. The algorithm using color markers has been rejected simply because it implies that the user wear markers, the presence of which being a non-desirable constraint on the user of such a system. The algorithm using geometric properties could not be applied in this situation since the camera angle, the curved shape of the hand of a guitarist playing, and the segmentation process do not create a contour image of the complete hand, therefore, the barycenter is not at the center of the palm and consequently, the fingertips are not necessarily the furthest points from it. On the other hand, the version of the circular Hough transform algorithm using edge images can be adapted to work on this problem as well as the projection signature algorithm. The circular Hough transform algorithm was finally retained for its superiority in terms of accuracy and precision. The retained algorithm is in the fingertips-based subgroup of appearance-based models and used a transformed version of the original colored image (an edge image).

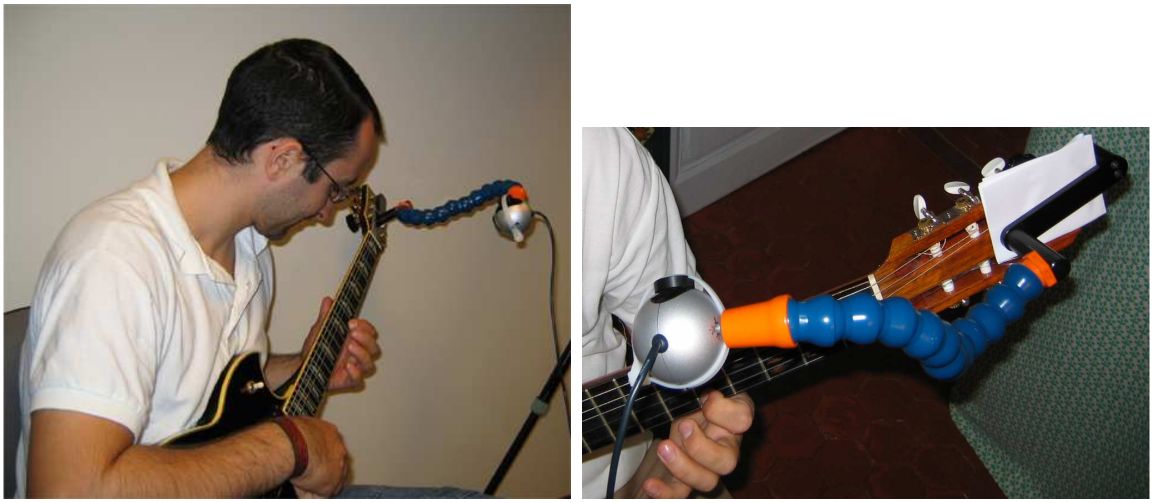
String and Fret Detection

The string and fret detection algorithm was implemented in EyesWeb and is based on the linear Hough transform method. The linear Hough transform EyesWeb block has been developed by the author and makes use of the Intel OpenCV library of functions (Intel, 2001). The finger-localization algorithm returns the fingertip positions in the form of (x, y) pixel spatial coordinates. In the case of fingering on the guitar, the left-hand gestural space can be defined in terms of $(string, fret)$ coordinates. Consequently, both string and fret must be detected. String and fret detection is not directly related to gesture but is useful to quantize the (x, y) pixel coordinates returned by the finger-localization algorithm into valid $(string, fret, finger)$ fingering coordinates.

Implementation of the Algorithm

Since the camera is attached to the guitar neck, as shown in figure 6.3, the string and fret positions are stable during the playing session. They therefore need to be localized only once at the beginning of the process. This is efficient on computer resources and also eliminates problems like occlusion by the hand of the guitarist and noise generated by the vibration of the strings. The algorithm only requires one image taken after the camera has been fastened to the neck, before playing. This also has the advantage that this image can be taken in favorable lightening conditions, for instance, when the strings and frets are apparent before going on stage in a live performance. Since the image captured by the camera is wider then the neck, the first step is to concentrate on the neck region by cropping this section from the image. In this prototype, this is done manually, but this step could be automated by using a neck model to find the neck region. As illustrated on line 1 of figure 6.6, the final preparation step is to convert the image to grayscale.

The algorithm then divides in two parallel processes that are the detection of the strings and the frets. Detection of the frets (figure 6.6 lines 2 and 4) and detection of the strings (figure 6.6 lines 3 and 5) use the same process with different parameters. First, a threshold operation is performed. Threshold segments the strings and frets from the neck. This is possible with most guitars since the strings and frets are of different colors than the neck. Threshold is advantageous compare to other segmentation methods since it works on a wide range of lighting conditions and does not require a model. This prototype uses a single-level threshold value set manually. A different threshold value is used for the frets and for the



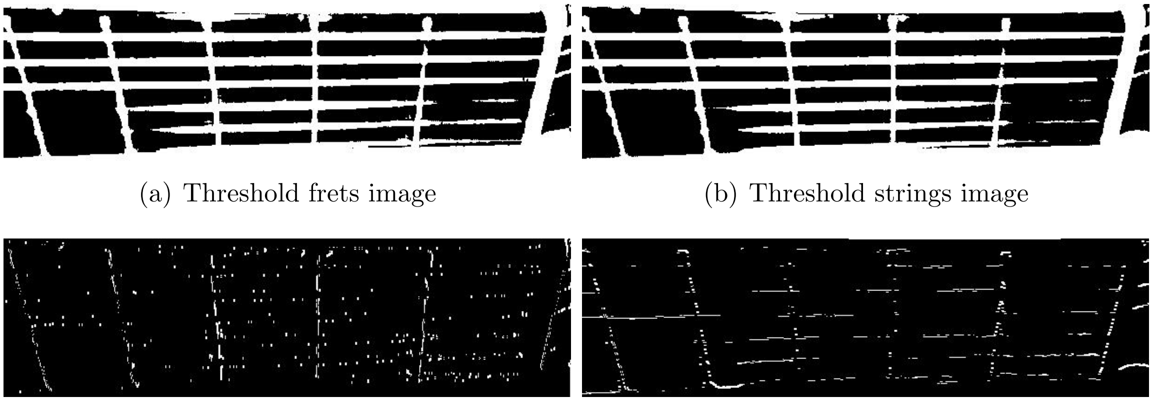
(a) Camera mount on an electric guitar

(b) Camera mount on a classical guitar

Figure 6.3 Camera attached to the neck of a guitar: (a) Electric guitar (b) Classical guitar

strings in order to enhance the searched component as much as possible. The next step is to apply a Sobel filter to the threshold image. A Sobel filter is a set of two convolution masks that can be applied separately to find the vertical and horizontal gradient responses of an image (Gonzalez & Woods, 1992). As figure 6.4 demonstrates, this allows to segment strings from frets.

The final step is to apply the linear Hough Transform. This prototype uses the probabilistic linear Hough transform as found in the Intel OpenCV library (Intel, 2001), but completes the line segments so that they fit the image full width or height. As explained in the Appendix A, the linear Hough transform finds all the possible lines passing throughout a set of points, therefore, each fret and string will be represented by many lines. These lines then need to be grouped in geometrical regions. The last operation is consequently to group the lines fitting each fret or string. This process is illustrated in figure 6.5. Fret and string regions can be displayed on the same picture but this is useful for visualization only.



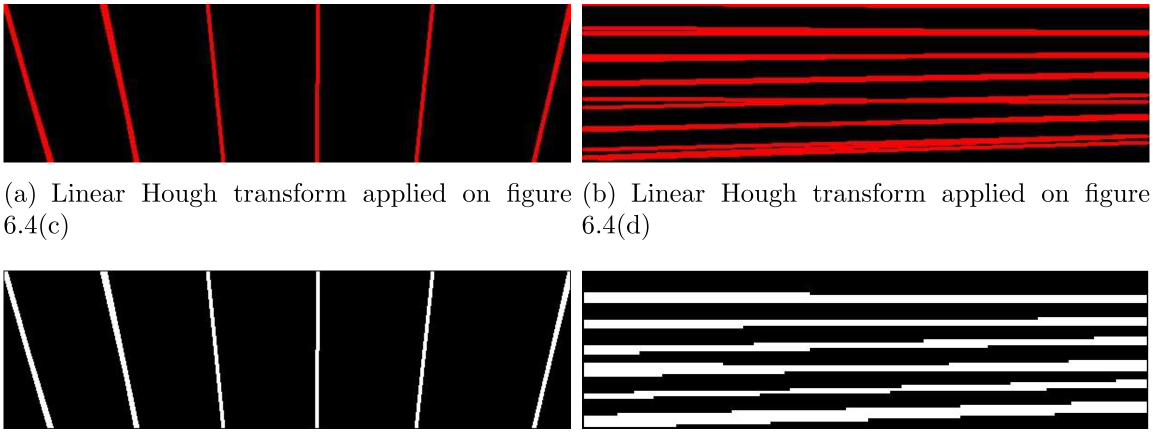
(a) Threshold frets image

(b) Threshold strings image

(c) Vertical Sobel filter applied on (a)

(d) Horizontal Sobel filter applied on (b)

Figure 6.4 Image preparation steps for the string and fret detection algorithm: (a) Threshold applied to accentuate frets (b) Threshold applied to accentuate strings (c) Vertical Sobel filter applied to the threshold frets image (d) Horizontal Sobel filter applied to the threshold strings image



(a) Linear Hough transform applied on figure 6.4(c)

(b) Linear Hough transform applied on figure 6.4(d)

(c) Regions grouping applied on (a)

(d) Regions grouping applied on (b)

Figure 6.5 Recognition steps for the string and fret detection algorithm: (a) Vertical line detection with the linear Hough transform (b) Horizontal line detection with the linear Hough transform (c) Vertical lines grouped in fret regions (d) Horizontal lines grouped in string regions

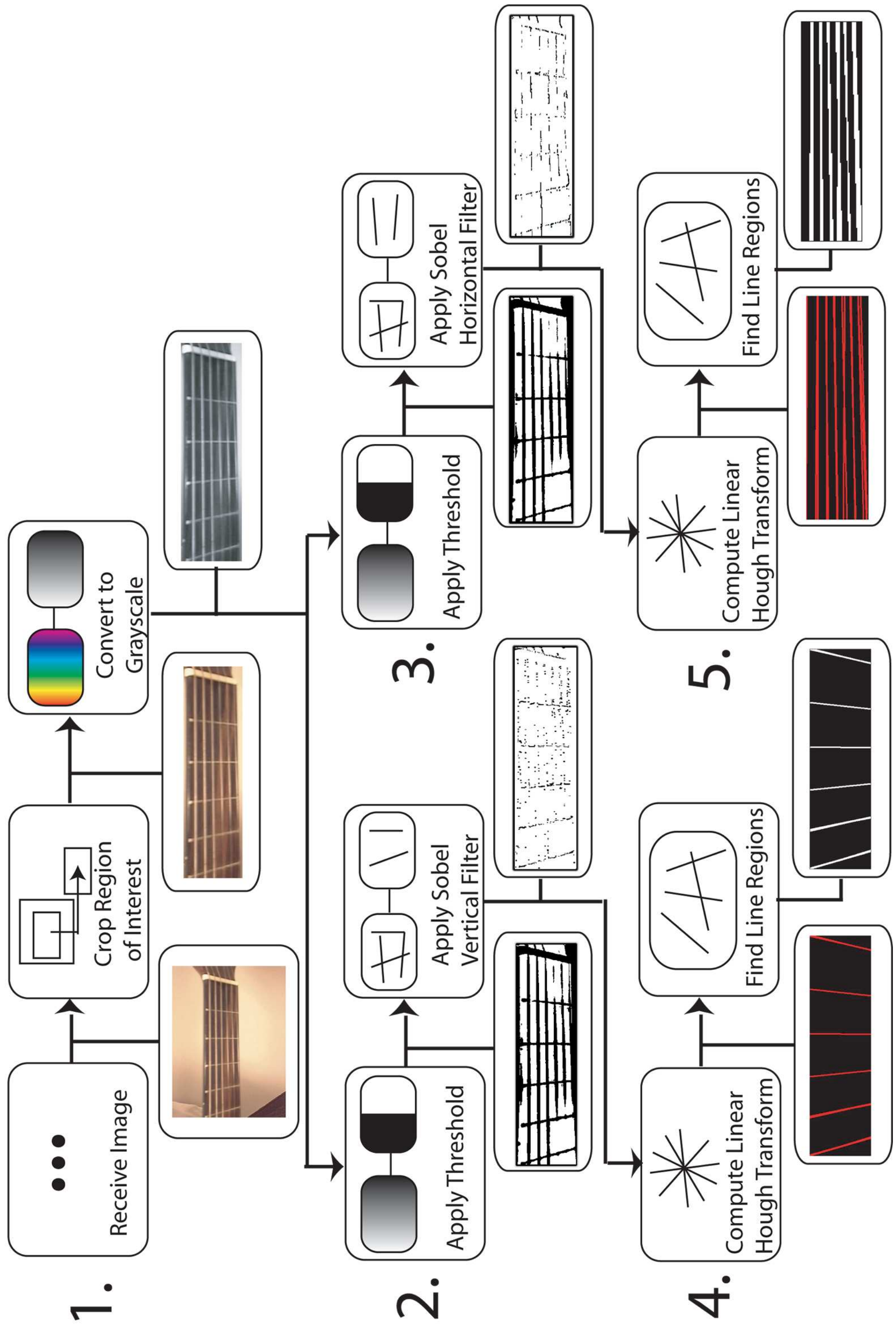


Figure 6.6 String and fret detection algorithm. Note: For better printout results, black and white pixels are inverted in the output images of lines 2 and 3

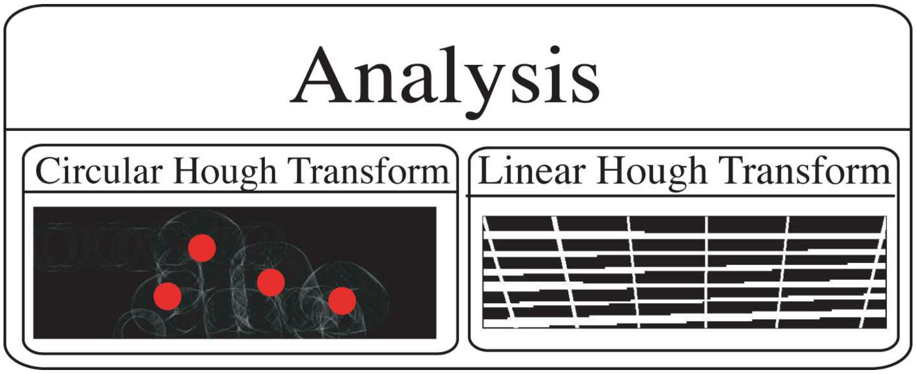


Figure 6.7 Analysis of the fingering image applying the circular and linear Hough transform of the neck region in order to obtain fingertips' positions and strings and frets regions.

6.1.2 Gesture Analysis

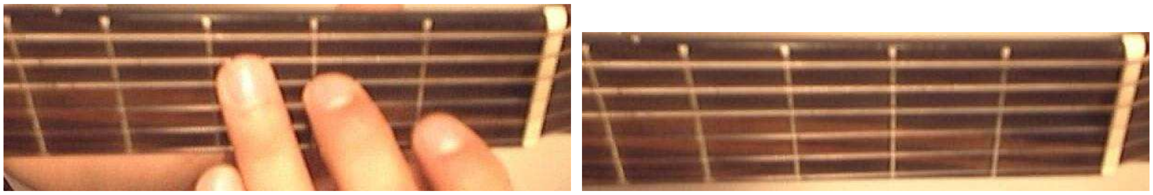
6.1.2.1 Feature Extraction

Localization

The first task of the analysis process is to define the ROI. In this system, this is done manually by drawing a rectangle around the neck region. This method implies that the camera is fastened to the neck so that the neck is perpendicular to the horizontal in the image, as seen in figures 6.3 and 6.8. The ROI for the finger-localization algorithms is defined to be a little bit wider than the neck at the bottom but tight to the neck at the top; this configuration allows a better view of the fingers on the first string. The ROI for the string and fret detection algorithm is strictly limited to the neck region. The ROI regions for the finger-localization algorithms and for the string and fret detection algorithm can be compared in figure 6.8.

Segmentation

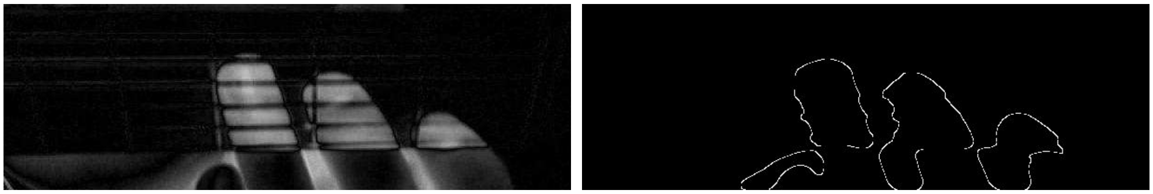
Once the ROI has been defined, the gesture must be "extracted" from the image. The segmentation process is illustrated on line 2 of figure 6.11. The system uses difference of pixel background segmentation on grayscale image to segment the hand of the guitarist from the neck. This is possible since the neck is stable in the image due to the camera setup. As it can be seen in figure 6.9(a), a little bit of noise is introduced by the vibration of the played strings but this is not significant enough to affect the silhouette and edges



(a) ROI for the finger-localization algorithms (b) ROI for the string and fret detection algorithm

Figure 6.8 [Region of interest in the image captured by the camera mount on the guitar neck: (a) ROI for the finger-localization algorithms (b) ROI for the string and fret detection algorithm

images of the guitarist hand. Most of this noise will be filtered by the finger-localization algorithms using a median filter on the threshold image of the hand (see line 2 of figure 5.3 and 5.1 in chapter 5).



(a) Background subtraction (b) Canny edge detection

Figure 6.9 Segmentation steps of the fingering algorithm: (a) Result of background subtraction of the grayscale image of figure 6.8(b) from the grayscale image of figure 6.8(a) (b) Canny edges detection applied on (a)

6.1.2.2 Parameter Estimation

Parameter estimation is an internal process of the finger-localization algorithm and the string and fret detection algorithm and is illustrated on line 3 and 4 of figure 6.11 respectively. The details of these algorithms are explained in chapter 5 for the finger-localization algorithm and in section 6.1.1.2 for the string and fret detection algorithm. What it is important to know is that the final output of the parameter estimation step will be fingertips (x, y) pixel spatial coordinates in the case of the finger-localization algorithm and $(string, frets)$ regions coordinates in the case of the string and fret detection algorithm (image 6.7). The fingering algorithm receives the pixel spatial fingertip positions from the finger-localization algorithm and needs to quantize these position into $(string, fret)$ coordinates. Different quantization methods can be used and were evaluated during the test sessions. A spatial position vertical component can be quantized to the nearest fret, to the left-most fret or to the right-most fret. In the same way, horizontal components of fingertip positions can be quantized to the nearest string, to the down-most or to the upper-most. Right-most quantization was chosen for frets since musicians rarely play directly on the frets but rather slightly to their left, consequently, fingertip's centers are most likely to be found on the left of the frets. Upper-most quantization was chosen for strings since they are pressed by the musician's fingertips, the center of which should therefore be detected under the strings. It is important to note that throughout this thesis left correspond the guitar nut direction and right to the guitar tonehole direction, as up refer to low-pitch strings and down to high-pitch strings. Exhaustive automated comparisons of the outputs of each combination of methods should be performed in the future to draw a final conclusion on the best quantization method for the largest group of playing situations and styles possible.

6.1.3 Gesture Recognition

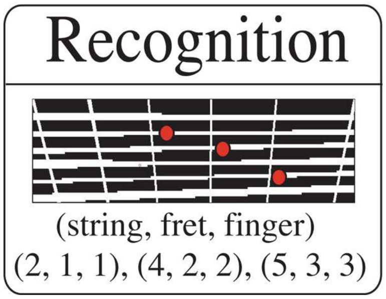


Figure 6.10 Fingering gesture recognition

The coordinates of the fingertips and the string and fret regions output at the previous step are now related together. At the previous step, fingertips coordinates were in pixels, after this step, they will be quantized to $(string, fret, finger)$ coordinates. The recognized gesture alphabet is therefore composed of six horizontal levels (the six strings), five vertical levels (the frets) and four fingers (index, middle, ring, and little). The vertical levels are limited to five in this setup due to hardware limitation (i.e. camera view angle), at the software level, any number of frets present in the image could be detected. This prototype output the fingering positions in the format $(string, fret, finger)$. The finger parameter is a number between 1 and 4 corresponding to the appearance order reading from left to right and, therefore, numbers do not necessarily always correspond to the same finger. For example, in the case of occlusion of the index by the middle finger, the middle finger will be labeled 1 by this prototype since occlusion is not supported in this version. The strings are labeled from 1 to 6 starting from the bottom and the frets are labeled from 1 to 5 (or more) reading from the right of the captured image (left of the guitar). If the camera is placed so that the first fret visible to the right is not the left-most fret of the guitar neck, the relative labeled number must be transposed to correspond to the absolute fret.

6.2 Test Methods

The prototype was tested on three different types of excerpts: the C major scale, the C major chords progression, and a short excerpt of the melody of Beethoven's *Ode An die Freude*. These excerpts cover the six strings, the three first frets, and are played with the index, middle, and ring fingers, further tests will be performed in the future to cover the whole camera view fret range and the four fingers. During the test session, the camera was fastened to the neck of a classical guitar. The ROI around the neck for the finger-localization algorithm and for the string and fret detection algorithm was manually selected. The threshold for the finger-localization algorithm and for the string and fret detection components of the string and fret detection algorithm were also manually selected. Finally, the circular Hough transform radius was selected to match the guitarist fingertip radius. The musician was then asked to play the three chosen excerpts using the fingering displayed on the scores (refer to figures 6.12, 6.13, and 6.14). The video images of the playing session were recorded by the camera attached to the neck and by a camera on a tripod in front of the musician. The videos taken with the camera on the guitar mount were then processed in realtime (i.e., without altering the playback speed) in the Eyesweb patch. The processing step was defered to allow testing of different settings, for example, to test the different quantization methods. Preliminary tests were also performed using an electric guitar and produced similar results although an exhaustive automated comparison needs to be done to confirm that the type of guitar does not influence the recognition rate.

The videos were also processed manually with the assistance of the musician in order to identify transition phases, note beginnings, and note ends. It is important to note that this manual processing was done in the same conditions as the automated processing, namely

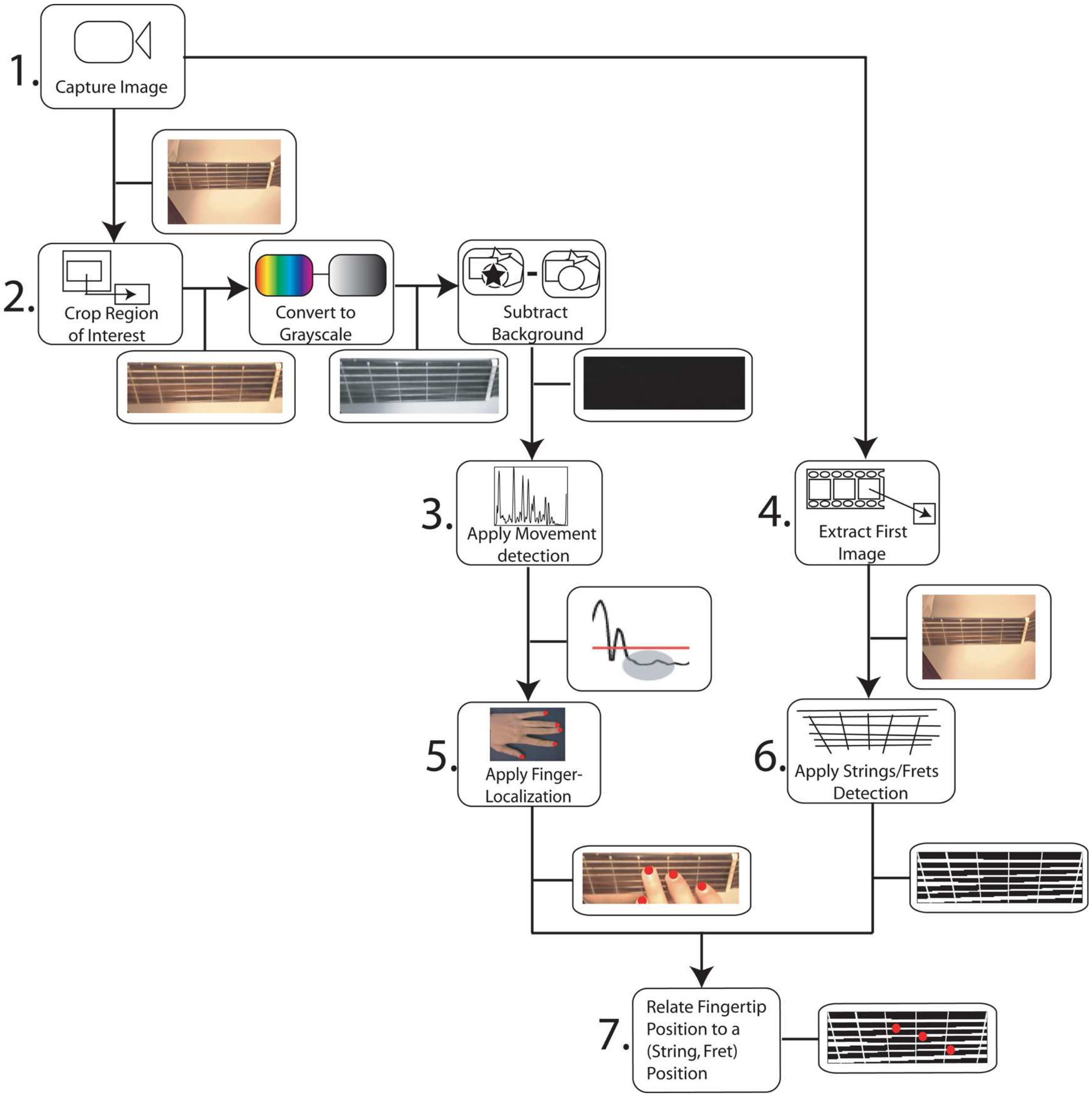


Figure 6.11 Prototype algorithm

by inspection of the images only, therefore, without the support of sound. In some cases, it was hard for the musician to precisely identify the exact frame at which a note begins or ends. Other comparison methods should be developed in the future to better segment transitions from note beginnings and ends. These new methods could involve multimodal integration of image and sound, for example, as it will be discussed in the future work section.

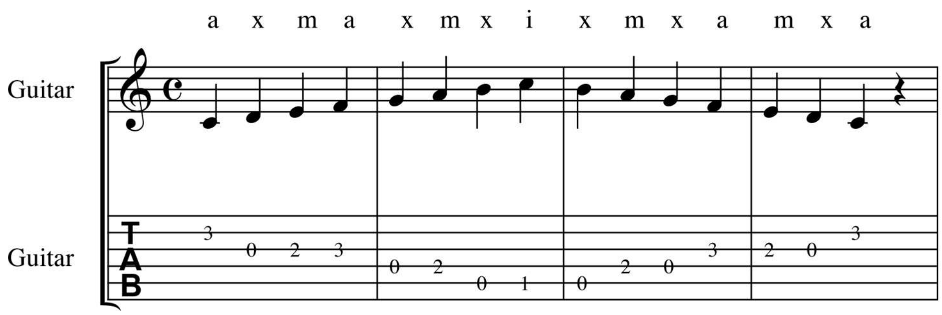


Figure 6.12 Test excerpt: the C major scale. Letters on the top line represent the finger used for the note (i = index, m = middle, a = ring, o = little)

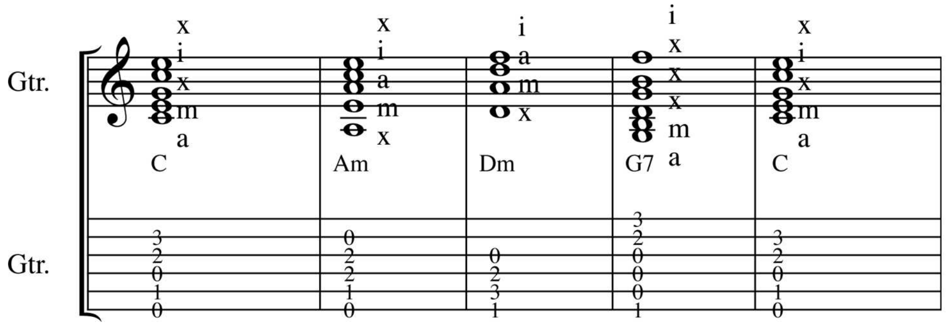


Figure 6.13 Test excerpt: the C major chords progression. Letters represent the finger used for the note (i = index, m = middle, a = ring, o = little)

6.3 Results

The system and musician's output were compiled in a table (available as a supplement of this thesis on the project website: `http://www.music.mcgill.ca/~amburns/masterproject/`). Analysis of the results for the three excerpts was automated in order to compare the musician and the system output. Results were compiled in the following way:

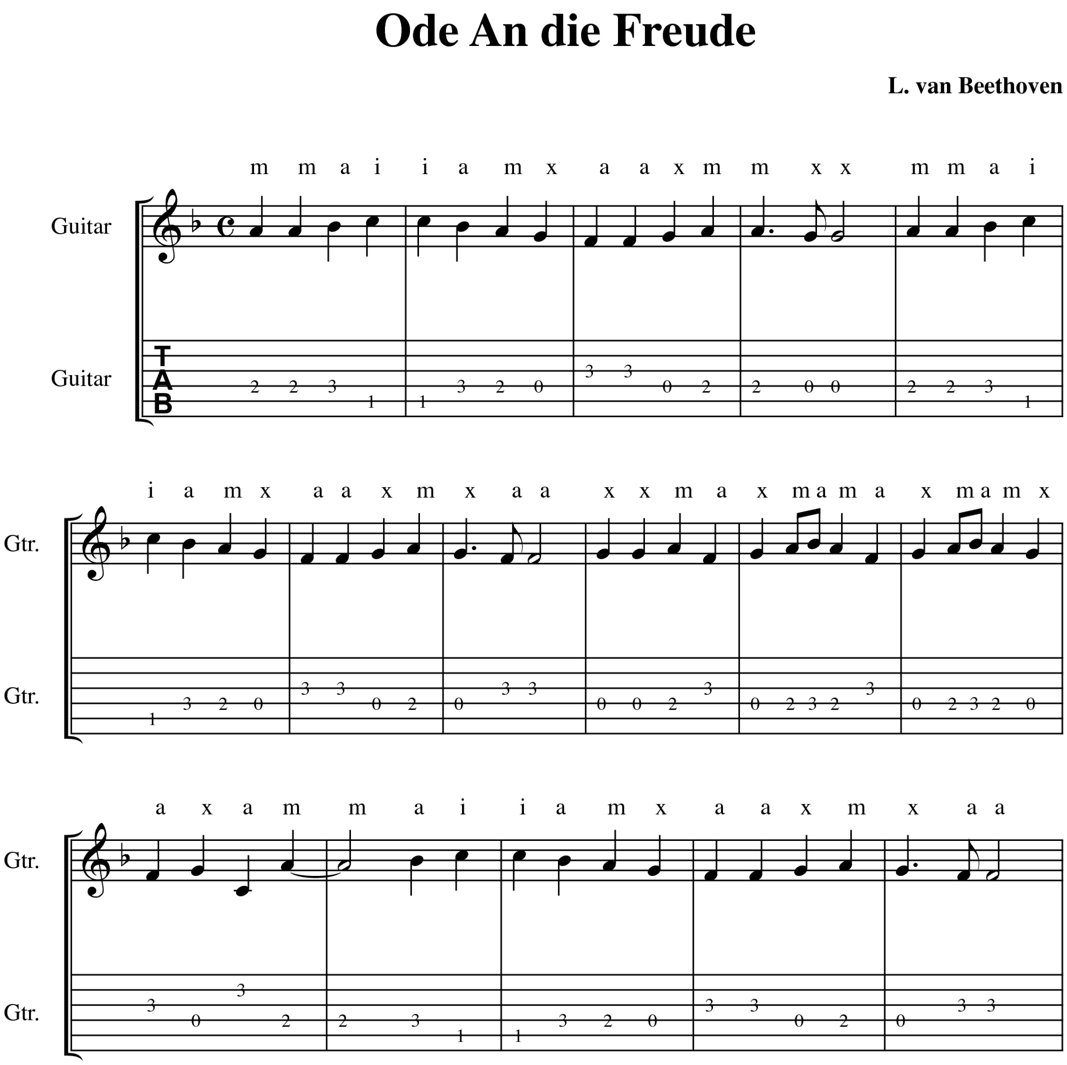


Figure 6.14 Test excerpt: Beethoven's *Ode An die Freude.* Letters on the top line represent the finger used for the note (i = index, m = middle, a = ring, o = little)

- Fingering positions are defined by the musician for the duration of notes and chords. System output during transition phases is consequently not considered (see table 6.1 for an example). It is the movement segmentation algorithm's task to eliminate system output during these phases. The results are compiled using the assumption that this task would have been accomplished successfully. The aim of the compiled results is to evaluate the recognition algorithm only, the movement segmentation algorithm is evaluated separately and will be discussed in section 6.4.

- Fingering positions triplets *(s#: string number x, f#: fret number x, d#: finger number x)* for open strings notes and unplayed strings are left empty by the musician, in this case, the positions of the fingertips are considered to be undefined. In a real playing situation fingers will probably be placed somewhere over a $(string, fret)$ position in preparation for the next note but this position will not be pressed. The actual prototype can evaluate a fingertip position with respect to the $(string, fret)$ grid but cannot determine if the position is pressed or not. Consequently, the system output is not considered during open string positions (see table 6.2 and table 6.3 for example).

In short, all the fingering positions left empty by the musician were not considered. All the other positions are considered. A match (displayed in bold in tables 6.1 to 6.4) can be partial, for example the system correctly identify only the string or fret, or complete, the $(string, fret, finger)$ triplets of the musician and system output are identical (see table 6.4).

			Output			
Frame	Phase	Output type	(s1, f1, d1)	(s2, f2, d2)	(s3, f3, d3)	(s4, f4, d4)
69	Transition	Musician				
		System	(3, 1, 1)	(1, 0, 2)	(4, 2, 3)	(4, 3, 4)
70	E	Musician		(**4**, **2**, 2)		
		System	(4, 1, 1)	(**4**, **2**, 2)	(4, 3, 3)	(1, 3, 4)

Table 6.1 Example of the output during a transition to E. On frame 70, E fingering is correctly recognized by the prototype.

Table 6.5 presents the recognition rate per excerpt. Each line reports the results for one musical excerpt and the last line reports the total recognition rate for all excerpts.

			Output			
Frame	Phase	Output type	(s1, f1, d1)	(s2, f2, d2)	(s3, f3, d3)	(s4, f4, d4)
160	G7	Musician	(**1**, **1**, 1)	(**5**, **2**, 2)	(**6**, **3**, 3)	
		System	(**1**, **1**, 1)	(**5**, **2**, 2)	(**6**, **3**, 3)	
161	G7	Musician	(**1**, **1**, 1)	(**5**, **2**, 2)	(**6**, **3**, 3)	
		System	(**1**, **1**, 1)	(**5**, **2**, 2)	(**6**, **3**, 3)	(4, 3, 4)

Table 6.2 Example of an undefined fingering position. In the G7 chord only the three first finger positions are defined, the little finger does not participate to the chord, consequently its position is undefined and is not considered for a match. Both frames 160 and 161 are perfect matches.

			Output			
Frame	Phase	Output type	(s1, f1, d1)	(s2, f2, d2)	(s3, f3, d3)	(s4, f4, d4)
64	D	Musician				
		System	(2, 1, 1)	(2, 2, 2)	(4, 3, 3)	(1, 4, 4)
65	D	Musician				
		System	(2, 1, 1)	(2, 2, 2)	(3, 3, 3)	

Table 6.3 Example of an open string “fingering”. This D is played on the open 4th string consequently all finger positions are undefined and cannot be identified by this prototype.

			Output			
Frame	Phase	Output type	(s1, f1, d1)	(s2, f2, d2)	(s3, f3, d3)	(s4, f4, d4)
50	C	Musician			(**5**, **3**, 3)	
		System	(2, 1, 1)	(4, 2, 2)	(**5**, **3**, 3)	
51	C	Musician			(5, **3**, 3)	
		System	(2, 1, 1)	(4, 2, 2)	(1, **3**, 3)	(5, 3, 4)

Table 6.4 Example of a complete and partial match. On frame 50, the C triplet $(5, 3, 3)$ is completely recognized while on frame 51 only the fret is correctly identified. Also note that only the triplet (s3, f3, d3) is considered for a match because the other positions are undefined.

	Index		Middle		Ring		Total		
	String (%)	Fret (%)	String (%)	Fret (%)	String (%)	Fret (%)	String (%)	Fret (%)	Complete (%)
Chords progression	86.9	100.0	38.3	70.1	17.8	52.3	47.7	74.1	**14.0**
Scale	100.0	100.0	79.0	79.0	22.4	70.7	38.0	73.4	**34.2**
Ode An die Freude	79.2	100.0	84.3	85.4	27.3	74.8	52.0	80.9	**51.6**
Total	85.7	100.0	60.9	77.2	23.1	65.3	48.2	76.2	**40.2**

Table 6.5 Recognition rate per musical excerpt

Recognition rates are divided per finger. The string and fret division means recognition rate of the strings played by the index finger, recognition rate of the frets played by the index finger, recognition rate of the strings played by the middle finger, and so on. The total column is the recognition rate of the played strings and frets independently of the finger. Finally, the complete column presents the recognition rate of a complete fingering (all the $(string, fret, finger)$ triplets composing a chord or a note).

	Fret			String					
	1 (%)	2 (%)	3 (%)	1 (%)	2 (%)	3 (%)	4 (%)	5 (%)	6 (%)
Chords progression	100.0	76.9	56.0	92.7	62.5	24.4	27.3	24.2	84.2
Scale	100.0	79.0	63.8	NA	100.0	87.5	64.7	19.2	NA
Ode An die Freude	100.0	85.4	75.9	NA	79.2	70.3	24.3	33.3	NA
Total	**100.0**	**80.3**	**67.4**	**92.7**	**66.7**	**59.2**	**28.9**	**22.5**	**84.2**
	76.2			48.2					

Table 6.6 Recognition rate per string and fret

Table 6.6 presents the results for each excerpt classified by fret and string. This presentation allows to observe if there is a variation in the degree of recognition between the different regions of the guitar neck. A recognition rate *NA* means that this string or fret was not played in this musical excerpt. The last line outlines the total recognition rate for each fret and string and the total recognition rate for all frets and all strings.

6.4 Discussion

6.4.1 Missed or Extra Fingertips Detection

More than eighty percent of the errors are caused by the detection of false fingertips or by the non-detection of a fingertip. The first type of errors is caused by phenomena such as string vibration, shadow or reflection due to lighting variation, and variation in the guitar neck color, for instance, due to aging of the instrument. These phenomena create noise that can be misinterpret as a circular shape by the Hough transform. These errors are difficult to solve but in some cases – like the one illustrated in table 6.7– they could be solved by applying rules like forbidding the detection of two fingers on the same $(string, fret)$ position. Problems due to lighting variations could potentially be solved using an infrared camera together with a ring of infrared LEDs providing a constant lighting. The second type of errors is mostly due to partial or total occlusion of a finger by another finger or by deformation of the quasi-circular shape of the fingertips due to the camera angle. These errors also cause the $(string, fret)$ position to be attributed to the wrong finger and in the worst case – for instance, when two fingers play the same fret on two different strings– a fingering position will completely be omitted. These errors are almost impossible to solve with hardware solutions unless more than one camera (with different views) are used. They could potentially be solved algorithmically by estimating fingertip trajectories from the previous non-occluded images or by locating the non-occluded part of finger and estimating the tip position from it.

6.4.2 Strings Spacing

Table 6.6 presentation is interesting because it clearly shows some of the problems linked with the hardware setup. Due to the placement of the camera, the space between the strings is smaller for the upper strings (D, A, E) than for the lower strings (G, B, E), affecting the accuracy of the recognition system. The angle of the camera also affects the quasi-circular shapes of the fingertips making these appear more flat and consequently more subject to be missed by the Hough transform as explained previously. In fact, it is possible to observe a decrease in the recognition rate from string 1 (high pitch) to string 5 (low pitch). The sixth string seems to be recognized better. This might be due to the fact that it is the last string, consequently fingertips that are found above it will also be quantized to the sixth string. As illustrated by figure 5.5 of chapter 5 and in (Burns & Mazzarino, 2006), the circular Hough transform has an accuracy of 5 +/- 2 pixels with respect to the color marker references placed at the center of the fingertip. The resolution of the camera used in this prototype is 640x480 pixels, giving a 610x170 pixels neck region. The distance between the A and E (low pitch) strings is of 12 pixels at the first fret and 17 at the fifth fret. Between the E (high pitch) and B strings, the distance is 16 and 20 pixels for the first and fifth fret, respectively. In the worst case the finger-localization algorithm error exceeds half the space between the upper strings and the fingertip center is detected above the string, this results in the fingertip being quantized one string upper then its real position. However, since this problem happens less frequently with high-pitched strings, where the distance between two strings is larger, the problem could have been solved using an higher-resolution camera. The higher recognition rate for the fret positions where the space between two frets is much larger also tends to confirm this hypothesis.

6.4.3 Guitar Neck Image Deformation

From table 6.6 it can also be observed that there is a small decrease in the fret recognition rate from left to right. This problem might be due to the camera angle that creates a deformation of the neck image (see figure 6.8(b)) and of the fingertips' shapes or to the angle at which the musician attacks the different frets. The neck image deformation or some attack angles can cause the fingertip center to appear slightly to the right of the fret. The chosen quantization method will therefore quantize the fingertip to the neck fret position. This problem could potentially be solved by applying a perspective correction algorithm to straighten the image. Perspective correction might also help to reduce the ''missing fingertips" type of error.

			Output			
Frame	Phase	Output type	(s1, f1, d1)	(s2, f2, d2)	(s3, f3, d3)	(s4, f4, d4)
162	G7	Musician	(**1**, **1**, 1)	(5, 2, 2)	(6, 3, 3)	
		System	(**1**, **1**, 1)	(1, 1, 2)	(5, 2, 3)	(6, 3, 4)

Table 6.7 Example of the detection of a false fingertip. The system is detecting two fingertips on the first string and fret, this causes the detection of the fifth string, second fret and sixth string, third fret to be shifted to the third and fourth fingers.

6.4.4 Movement Segmentation Error

Results of the segmentation algorithm are not presented in this chapter because they seem to be unrelated to the phases observed by the musician. The method of thresholding the motion curve presented in chapter 4 works for chords and the assumption was that it would have been possible to detect minima in the motion curve of sequences of notes, but this assumption failed. It is either because the assumption is wrong, and consequently it might not be possible to rely on left-hand image only for movement segmentation, or because the motion curve would need further high-pass filtering to remove small variations that cause minima unrelated to note nucleus and generate false segmentation. The second hypothesis is the preferred one since it was possible to located minima at note nucleus by visual inspection of the motion curve as it can be seen in figure 6.2(b). Further tests are required to draw a definitive conclusion on this matter.

6.5 Conclusion

This prototype meets most of the objectives set in chapter 1:

- The system outputs the musician solution and consequently accounts for all aspects of the fingering choice.
- The system does not require any preliminary information or analyses of the musical excerpt, it reads the fingering solution directly from the musician execution of the excerpt.
- The system is non-obtrusive, the musician does not need to adapt his playing style or to wear special devices. Only the weight of the guitar mount can be disturbing but this could be solved by using a lighter camera-mount setup.
- The system is composed of a regular webcam on a mount and is easy to affix to the guitar. The software requires only few manual settings that it will be possible to automate in the future version. The system is therefore accessible in terms of cost and ease of use. However, further testing are still required to conclude on the reproducibility of the results on a variety of guitars.

Although the recognition rate for chords is lower than the one in the preliminary analyses (chapter 4) this algorithm demonstrated the potential of the use of computer vision to solve the fingering problem. In fact, by detecting individual fingers, it is possible to obtain partial fingering information, for instance two notes of a three notes chord are solved or $(string, fret)$ coordinates are correctly recognized but are attributed to the wrong finger. In some cases, it is possible that this partial information could be used to make an educated guess on the complete fingering. Also, as the discussion section highlighted, many of the problems could be solved by small modifications of the hardware and software settings. These improvements will be discussed further in chapter 7. This prototype therefore satisfies this thesis objectives and opens the possibility of several future work developments.

Chapter 7

Future Work

Results of the prototype are encouraging and open possibilities for studies on many aspects of the guitarist instrumental gesture, namely gesture segmentation, anticipatory movements, bimanual synchronization, movement optimization, and choice of fingering. Future work can be divided into three categories: a) hardware and software developments, b) test methods and comparisons automatization, and c) data analyses and usage.

7.1 Hardware and Software Developments

The actual prototype has hardware limitations:

1. Problems related to the choice of camera and environment:
   - (a) Only 5 frets can be observed at the time.
   - (b) The resolution of the camera currently used is 640 x 480 pixels, 30 frames per second.
   - (c) A controlled lighting environment is required.
   - (d) The guitar neck and the strings need to be of contrasting colors. For example, regular nylon high pitch strings are hardly detectable on a light brown maple wood guitar neck.
2. Problems linked to the unimodal data acquisition system:
   - (a) Only the images of the left-hand are observed. The system does not have any information about the right-hand gesture, the guitarist global position and gesture, and the sound.

3. Problems linked to the guitar mount:
   (a) The guitar mount is flexible which makes the test angle difficult to reproduce.
   (b) The flexible guitar mount is sensitive to large, fast movements of the guitarist, decalibrating the string and fret detection algorithm.
   (c) The guitar mount weights approximately 650 grams, adding a disturbing weight to the guitar neck.

And software limitations:

1. Fingertip positions are determined in each individual frame, individual fingers are not tracked from one frame to another. Consequently finger number one is not always the index, two the major, and so on.
2. Fingertip positions are not evaluated during occlusion.
3. Gesture segmentation based on left-hand finger movement works for chords but does not provide satisfying results with sequences of notes.

Some paths to solve these hardware limitations are the following:

1. Experiment with different types of camera:
   (a) Wide angle.
   (b) Higher resolution.
   (c) Infrared camera and specific illumination.
2. Experiment with multimodal data acquisition environment:
   (a) Use multiple cameras.
   (b) Add right-hand view, tracking, and synchronization.
   (c) Add sound and image analysis and synchronization.
3. Create a new camera mount:
   (a) Fixed view angle.
   (b) More stable to guitarist motion.
   (c) Lighter materials.

Software problems could be solved by exploring algorithmic solutions and by performing more experiments on these currently really active research topics:

1. Individual fingertips tracking and identification.
2. Fingertip positions estimation during occlusion.
3. Preferred and non-preferred hand gesture, bimanual synchronization, and anticipatory placement movement.

Another potential software improvement would be to automate the choice of the currently manual parameters. These parameters include: the Hough transform threshold and radius, the string and fret segmentation thresholds and the ROI region for both algorithm. Threshold automatic selection methods exist and could be applied, a basic example of these being Otsu (1979) method. As mention in chapter 5 methods for estimating the fingertip radius from the detection of the palm size already exist. The ROI region could also be automatically found using pattern matching of a model of the guitar neck on the captured image or by using the two extremity lines found by the Hough transform as the neck limits.

7.2 Test Methods and Comparisons Automatization

In order to improve the development of the prototype, more extensive tests and automated comparison methods are required. Future tests should:

- Cover the full range of frets viewed by the camera
- Involve the use of all four fingers
- Cover a wider variety of excerpt styles
- Involve a larger set of musicians and guitar types
- Involve multiple captures of the same excerpt play by the same musician
- Cover all the combinations of quantization methods

Future comparison methods should:

- Allow an automated comparison of the output of the different quantization methods
- Allow to determine the percentage of error relative to each error types

The method used to obtain the musician output regarding the segmentation of note beginning, ending, and transition should also be reviewed since the method relying on the image only proved to be difficult at boundaries between transition and note. A method relying on the detection of note beginning from the sound file could maybe be useful.

7.3 Data Analysis and Usage

The prototype has demonstrated to have many interesting possibilities that could be used in many musical and non-musical fields. For example, motion curves are relevant for the study of anticipatory placement movements and bimanual synchronization, topics of interest in cognitive psychology and music education. Fingering information could be used in live performance to control sound effects or synthesis variables, or recorded to generate a score or to be analyzed for theory or educational purposes.

The actual prototype allows for the retrieval of a motion curve representing the global amount of movement of the left hand. Future work on individual tracking of fingertips will allow the acquisition of a similar curve representing the amount of motion for each finger. This kind of motion curve allows the detection and study of anticipatory placement gesture of fingers. Combined with information about the string excitation acquired throughout sound analysis or right-hand motion analysis, this could also be used for the study of bimanual gestures and synchronization in music.

The prototype demonstrated that a more robust version (not sensitive to lighting changes and guitarist motion) could be useful for live performances. A system like the one used by Doati (2006) presented in chapter 2 already expresses musicians and composers' will to use the guitar as a controller. Also as explained by Cuzzucoli and Lombardo (1999) and Laurson et al. (2001), fingering information is important in guitar physical modeling; it would therefore be natural to use the guitar itself to acquired this information. The guitar could, of course, be used to control any kind of sound effect or synthesis. The guitar, as a controller providing information about the triplet $(string, fret, finger)$ could also be used for score following and automatic accompaniment generation, and automatic score and tablature generation during improvisation or composition.

Finally, the most obvious use of the prototype is for the study of fingering. Fingering is evidently an important topic in musicology, but it has also been studied by psychologists trying to understand the choice of biomechanical optimum gesture in human actions and more specifically in trained actions like music playing. The current prototype allows the detection of the fingertip position related to the grid formed by the strings and frets on the fingerboard at any moment. The study of this information retrieved from many musicians with equivalent or different training levels may determine if there exist common fingering strategies and at what level of their training musicians acquire these strategies.

Chapter 8

Conclusion

The aim of this thesis was to develop a prototype for the realtime retrieval of a guitarist left-hand fingering. The main objective of the thesis was to investigate if computer vision can be used for this kind of task. Intermediate objectives were to develop a prototype that:

- Accounts for all factors involved in the choice of fingering
- Does not require prior information or analyses of the musical excerpt
- Does not impose constraints on the musician, i.e. the musician should not have to wear external devices or to adapt his playing style
- Is accessible in term of cost, ease of use and allow for the reproducibility of the results.

Chapter 2 presented existing methods to solve the fingering problem. These methods are applied at three different strategic moments of the music production process:

- Before the performance (pre-processing)
- During the performance (realtime)
- After the performance (post-processing)

All pre-processing methods presented rely on the analysis of the score. These methods are not satisfying the requirements of this thesis in that they do not consider all factors involved in the choice of a fingering, they mostly favor biomechanical optimal solution, and they require prior analyses of the score to establish a set of rules and in some case to prepare score fragments. The realtime methods presented in this thesis fall in two categories: MIDI guitar and guitar-like controllers. Both categories of methods only partially answer the fingering problem because they only solve the $(string, fret)$ component. Furthermore, these methods do not satisfy the requirements of this thesis since they require an adaptation of the playing style and consequently do not respect the musician naturalness, they are also sometime expensive and difficult to use. The post-processing method presented in this thesis is based on sound analysis of a guitar performance recording. Like the realtime methods it only solves the $(string, fret)$ component of the fingering problem. At the moment of writing this thesis it works only with one note at the time, making it unusable with chords. Furthermore, its accuracy on fretted strings varies between 3.8 and 8.3 centimeters consequently covering more than fret spacing. A method using computer vision was also presented. It demonstrated the will of musicians and composers for alternative guitarist gesture retrieval methods. Its constraints are that the musician is required to paint his finger and need to restrain his movements.

Since none of the existing methods completely satisfy the thesis requirements a new method was developed. Computer vision was chosen based on the thesis requirements, since, as opposed to the other sensing technology normally used in HCI and music gesture retrieval, it has the potential of being non-intrusive, and can be developed using a wide-public low-cost camera that is accessible in term of cost and usability. To facilitate the development and evaluation of the method, a methodology based on three steps (gesture modeling, gesture analysis, gesture recognition) was proposed in chapter 3.

Chapter 4 presented preliminary analyses that were conducted to use already available blocks of the EyesWeb software to perform the fingering recognition task. The analyses showed that the existing blocks are not sufficient to meet this thesis requirements, but allowed to further define the specifications for a prototype. The additional requirements were:

1. To reduce ancillary gestures to the minimum in order to concentrate on effective gestures only
2. To use a representation that considers the action of individual fingers
3. To use of a recognition mechanism that is not limited to previously learned material.

To fulfill these requirements, a study on finger-localization algorithms was performed (chapter 5). Four methods were implemented in Eyesweb and their characteristics were highlighted. The characteristics of the finger-localization algorithms oriented the choice to the circular Hough transform to solve the guitar left-hand fingering problem.

Chapter 6 presented a prototype in which the circular Hough transform was combined to a string and fret detection algorithm to output fingertip positions in $(string, fret, finger)$ coordinates that solve the fingering problem. Based on the hypothesis, developed in chapter 3, that the fingering gesture can be divided in three phases (preparation, nucleus, retraction), a movement segmentation algorithm was also proposed.

Chapter 6 concluded that the prototype meets the thesis requirements because it demonstrates that computer vision can be used to solve the guitarist fingering problem without imposing constraints on the guitarist nor requiring prior information on the performances. A system based on computer vision captures the musician fingering and consequently accounts for all factors influencing the choice of a fingering. Furthermore, a system based on finger-localization recognizes fingering according to a grammar of $(string, fret, finger)$ and not to a knowledge-base. However, the actual system needs to be improved to be robust and reliable in live performances. Improvements were suggested in chapter 7.

Appendix A

Computer Vision Tools

A.1 The Hough Transform

The Hough transform is an important concept in pattern matching. It uses the mathematical description of a geometric shape to find regions of an image that best fits that shape. Its use in computer vision is born from the observation that industrial and natural images contain shapes that can be approximated by geometric shapes. In this thesis, two kind of Hough transform are used:

1. The linear Hough transform is used to detect the guitar strings and frets;
2. The circular Hough transform is used to detect fingertips, whose ends can be approximated with a semi-circular shape.

A.1.1 Linear Hough Transform

The Hough transform first form is the linear one. It has appeared in 1962 in a patent awarded to Hough. The primary aim of the patent was to automate the analysis of complex patterns of particle tracks in pictures obtained from a bubble chamber (Hough, 1962) but as Hough himself mentions: "Persons skilled in the art will, of course, readily adapt the general teachings of the invention to embodiments other than the specific embodiments illustrated." It did not take many years for this affirmation to become true. Levine (1985) and Illingworth and Kittler (1988) reviews mention the first reference to the Hough transform in the picture-processing literature in (1969) by Rosenfeld in a book called *Picture Processing by Computer*. However, it was found in Deans (1981) that the Hough transform is a special case of the Radon transform known since 1917. The first improvements to the Hough transform computation were suggested by Duda and Hart (1972). Levine mentions that it was then followed by extensive literature coverage and applications in various fields ranging from biomedical, and scene analysis to vanishing points in three-space, binary image compression, and tracking moving targets. This is still true today where further improvements and variations of the Hough transform exist including implementations using fuzzy-logic, neural network, and heuristic and probabilistic approaches.

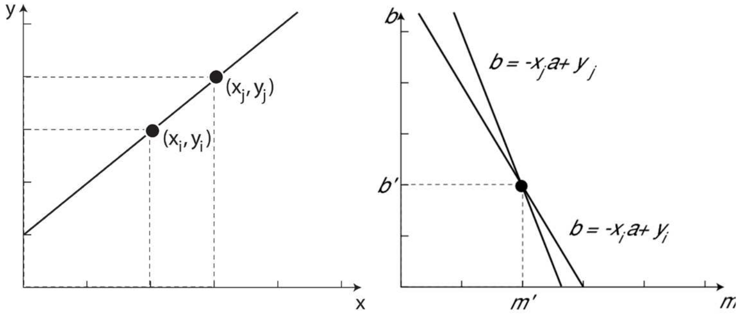
As exemplify in Gonzalez and Woods (1992) the original method proposed by Hough is a simple but efficient one. Hough first considered the line equation

$$y_i = mx_i + b \tag{A.1}$$

and observed that even if infinitely many lines pass through (x_i, y_i) they all satisfy equation A.1. He therefore modified the equation to work in the mb plane:

$$b = -x_i m + y_i \tag{A.2}$$

Using this parameters space all points contained on a line with slope m' and intercept b' will intersect at (m', b'). This fact is illustrated in figure A.1.

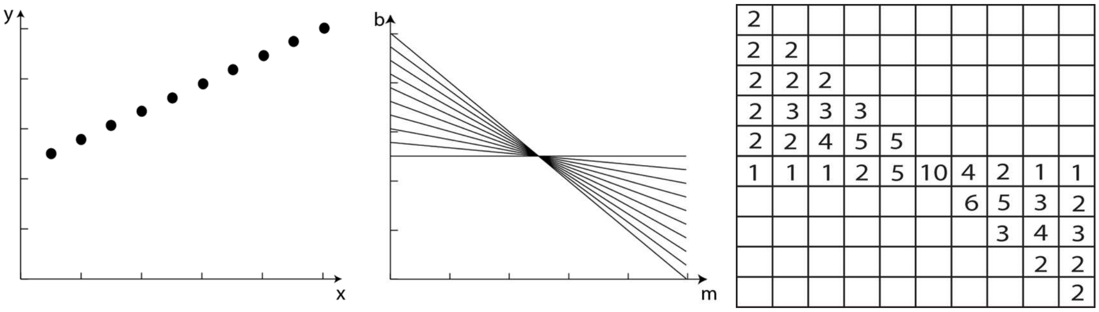


(a) Collinear points

(b) Concurrent lines

Figure A.1 Parametrization using the line equation: (a) Two collinear points in the xy plane; (b) Intersection in the mb plane of the concurrent lines representing the points i and j.

In the real domain there exist infinitely many lines that pass through a point, the mb parameter space therefore needs to be discretized to work in the digital domain. This is done by dividing the parameter space in accumulator cells. Each point is then tested for all possible m values in the discrete mb space. The cells are called accumulators because they are incremented each time a point is tested on their (m, b) coordinates. Maxima in the mb space correspond to detected lines in the xy space. Figure A.2 illustrates an example of ten points that can be linked into a line. Figure A.2(b) displays the solutions for the line in A.2(a). It can be observed that collinear points in the xy space correspond to concurrent lines in the mb space. These lines intersect at the (m, b) coordinates corresponding to the line that best fit the collinear points. The cell corresponding to the intersection contains a maximum, as can be observed in figure A.2(c).



2									
2	2								
2	2	2							
2	3	3	3						
2	2	4	5	5					
1	1	1	2	5	10	4	2	1	1
						6	5	3	2
							3	4	3
								2	2
									2

(a) Collinear points (b) Concurrent lines (c) Accumulator cell

Figure A.2 Example of 10 collinear points: (a) Collinear points; (b) Concurrent lines; (c) Accumulator cell.

One problem with this approach is that the intercept and the slope approach infinity as the line approaches the vertical. This problem can be solved by using the normal representation of a line, as explained by (Duda & Hart, 1972).

As illustrated in figure A.3, a line can be described by the angle θ of its normal and by its distance ρ from the origin using the following equation:

$$x \cos\theta + y \sin\theta = \rho \tag{A.3}$$

In this method, each point will be tested on a discrete interval of θ comprised of numbers between $[0, \pi]$, as it was tested for all possible values of m. Figure A.4 shows that points on a same line will have concurrent curves in the (θ, ρ) space. As was the case in the (m, b) space, the problem is therefore to find points of intersection that are represented by maxima in the two-dimensional array of accumulator cell.

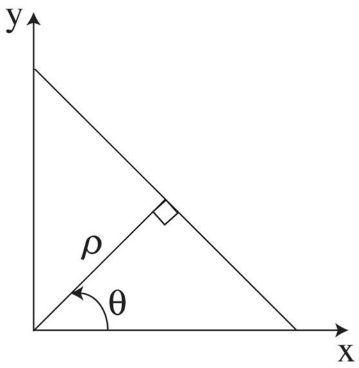
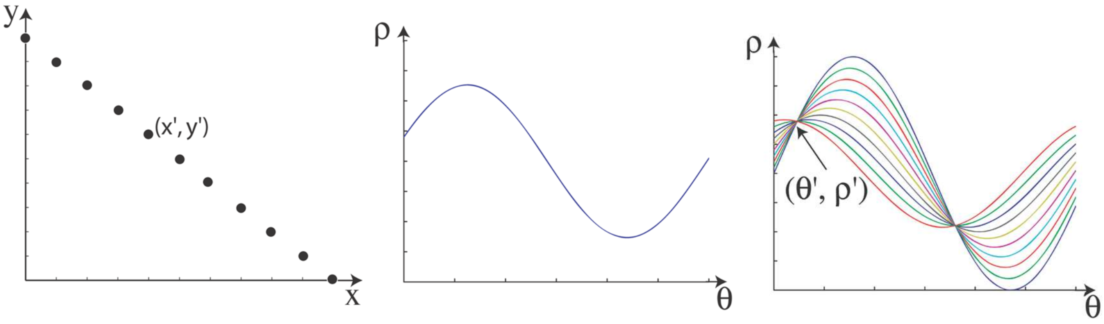


Figure A.3 Normal parametrization of a line given by equation A.3 where θ and ρ are fixed parameters



(a) Collinear points

(b) One point in the (θ, ρ) space

(c) The 10 points

Figure A.4 The normal parametrization of the line: (a) Ten collinear points; (b) ρ as a function of θ, applying equation A.3 for fixed parameter x' and y'; (c) Intersection of the concurrent curves representing the ten points in A.4(a).

A.1.2 The circular Hough Transform

The concept underlying the circular Hough transform introduced by Duda and Hart (1972) is quite similar to the linear one. As described in that article, the Hough transform can be applied to any shape that can be represented by a parametric equation. The number of dimensions, and therefore, the complexity of the accumulator cell, depends on the number of parameters.

In the case of the circle, equation A.4 can be used and generates a three-dimensional array with parameter space (a, b, r) where (a, b) are the coordinates of the center of the circle and r is the radius.

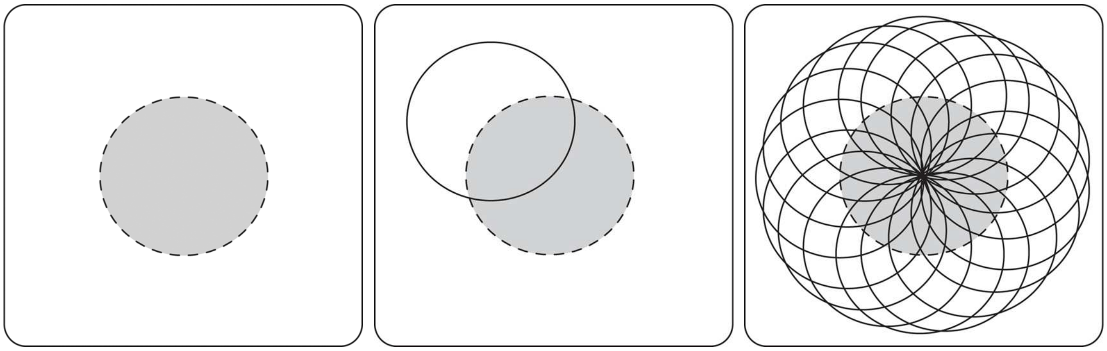
$$(x - a)^2 + (y - b)^2 = r^2 \tag{A.4}$$

When possible, it is advantageous to reduce the parameter space to only (a, b) and to test the image over a fixed radius or a reduced set of r. In this case, equations A.5 and A.6 can be used.

$$x = a + r \sin\theta \tag{A.5}$$

$$y = b + r \cos\theta \tag{A.6}$$

Figure A.5 illustrates how the Hough transform is applied to the contour image of a circular shape. Figure A.5(a) represents the circular shape to detect. Figure A.5(b) demonstrates how circles of a given radius are drawn around the contour image. Discrete points of the circular shape contour are used as center (the circular contour is shaded for clarity). Finally, figure A.5(c) shows that in the case of a match (circular shape of the search radius) all drawn circles will intersect at the center of the detected circle. This will translate in a maximum in the accumulator cells array.



(a) A circular shape

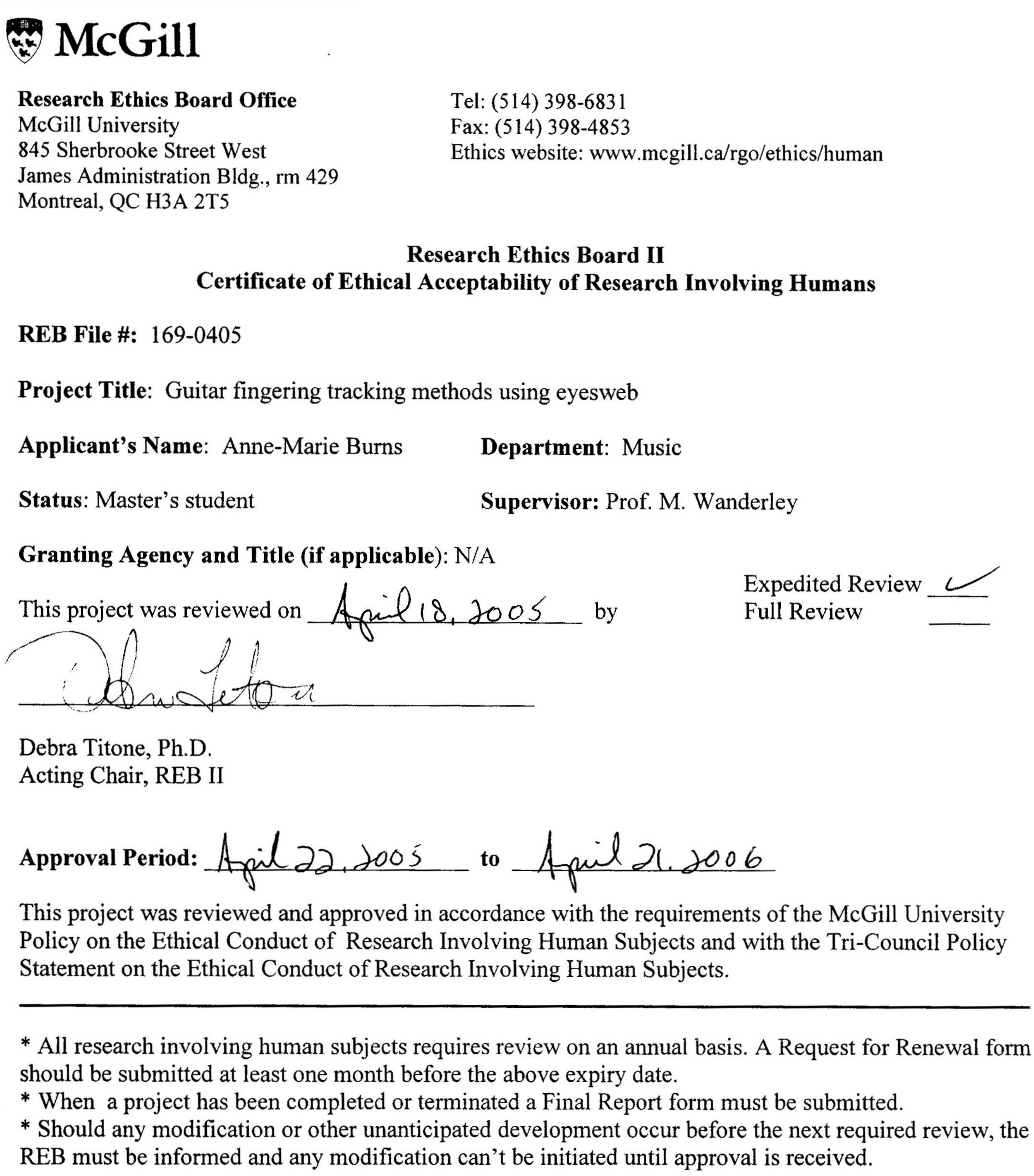
(b) Hough transform apply on (a)

(c) Maxima at the center of (a)

Figure A.5 The circular Hough transform: (a) A circular shape; (b) Circles of a given radius are drawn along the contour of the circular shape; (c) The intersection of all circles indicate the center of the detected circular shape.

Appendix B

Certificate of Ethical Acceptability of Research Involving Humans



McGill

Research Ethics Board Office
McGill University
845 Sherbrooke Street West
James Administration Bldg., rm 429
Montreal, QC H3A 2T5

Tel: (514) 398-6831
Fax: (514) 398-4853
Ethics website: www.mcgill.ca/rgo/ethics/human

Research Ethics Board II
Certificate of Ethical Acceptability of Research Involving Humans

REB File #: 169-0405

Project Title: Guitar fingering tracking methods using eyesweb

Applicant's Name: Anne-Marie Burns **Department**: Music

Status: Master's student **Supervisor:** Prof. M. Wanderley

Granting Agency and Title (if applicable): N/A

Expedited Review ✓
Full Review ____

This project was reviewed on April 18, 2005 by

Debra Titone, Ph.D.
Acting Chair, REB II

Approval Period: April 22, 2005 to April 21, 2006

This project was reviewed and approved in accordance with the requirements of the McGill University Policy on the Ethical Conduct of Research Involving Human Subjects and with the Tri-Council Policy Statement on the Ethical Conduct of Research Involving Human Subjects.

* All research involving human subjects requires review on an annual basis. A Request for Renewal form should be submitted at least one month before the above expiry date.
* When a project has been completed or terminated a Final Report form must be submitted.
* Should any modification or other unanticipated development occur before the next required review, the REB must be informed and any modification can't be initiated until approval is received.

References

Baader, A. P., Kazennikov, O., & Wiesendanger, M. (2005). Coordination of bowing and fingering in violin playing. *Cognitive Brain Research*, *23*, 436-443.

Blue Chip Music Technology. (n.d.). *Axon Ax-100 manual.* Retrieved June 23, 2006 from `http://www.musicindustries.com/manuals/axon/AX-100-manual.pdf`

Bod, R. (2001). Using natural language processing techniques for musical parsing. In *Proceedings joint international conference of the association for computers and the humanities and the association for literary and linguistic computing.* New-York, United States of America.

Bod, R. (2002). Memory-based models of melodic analysis: Challenging the gestalt principles. *Journal of New Music Research*, *31*(1), 27–37.

Burns, A.-M., & Mazzarino, B. (2006). Finger tracking methods using EyesWeb. In S. Gibet, N. Courty, & J.-F. Kamp (Eds.), *Gesture workshop 2005 proceedings* (Vol. LNAI 3881, pp. 156–167).

Cabral, G., Zanforlin, I., Lima, R., Santana, H., & Ramalho, G. (2001). Playing along with d'Accord guitar. In *Proceedings of the 8th brazilian symposium on computer music.*

Cadoz, C., & Wanderley, M. M. (2000). Gesture - music [electronic]. In M. M. Wanderley & M. Battier (Eds.), *Trends in gestural control of music* (pp. 29–65). IRCAM.

Camurri, A., Mazzarino, B., & Volpe, G. (2004). Analysis of expressive gesture: The Eyesweb expressive gesture processing library. In A. Camurri & G. Volpe (Eds.), *Gesture-based communication in human-computer interaction* (Vol. LNAI 2915, pp. 460–467). Springler Verlag.

Canny, J. A. (1986). Computational approach to edge detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *8*(6), 679-698.

Chan, S. C. (2004). *Hand gesture recognition.* Retrieved October 19, 2004, from `http://www.cim.mcgill.ca/~schan19/research/research.html`

Cuzzucoli, G., & Lombardo, V. (1999). A physical model of the classical guitar, including the player's touch. *Computer Music Journal*, *23*(2), 52–69.

Deans, S. R. (1981, March). Hough transform from the Radon transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI–3*(2), 185–188.

Delalande, F. (1988). La gestique de Gould, éléments pour une sémiologie du geste musical. In G. Guertin (Ed.), *Glenn Gould pluriel* (pp. 85–111). Montréal, Québec, Canada:

Louise Courteau.

Doati, R. (2006). My Eyesweb experience. In *EyesWeb week 2006.*

Duda, R. O., & Hart, P. E. (1972, January). Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, *15*(1), 11-15.

Elkoura, G., & Singh, K. (2003). Handrix: Animating the human hand. In D. Breen & M. Lin (Eds.), *Proceedings of eurographics/siggraph symposium on computer animation.*

Engel, K. C., Flanders, M., & Soechting, J. F. (1997). Anitcipatory and sequential motor control in piano playing. *Experimental Brain Research*, *113*, 189-199.

Gilardino, A. (1975a). Il problema della diteggiatura nelle musiche per chitarra. *Il "Fronimo"*, *10*, 5–12.

Gilardino, A. (1975b). Il problema della diteggiatura nelle musiche per chitarra. *Il "Fronimo"*, *13*, 11–14.

Glatt, J. (1999). *Using MIDI guitars.* Retrieved January 4, 2006, from `http://www.borg.com/~jglatt/tutr/midigtr.htm`

Gonzalez, R. C., & Woods, R. E. (1992). *Digital image processing.* Addison-Wesley.

Heijink, H., & Meulenbroek, R. G. J. (2002). On the complexity of classical guitar playing: Functional adaptations to task constraints. *Journal of Motor Behavior*, *34*(4), 339–351.

Hemmi, K. (2002). On the detecting method of fingertip positions using the circular Hough transform. In *Proceedings of the 5th asia-pacific conference on control and measurement.*

Hough, P. V. C. (1962). Method and means for recognizing complex patterns. *U.S. Patent*, *3,069,654*.

Hu, M.-K. (1962, February). Visual pattern recognition by moment invariants. *IEEE Transactions on Information Theory*, *8*(2), 179–187.

Illingworth, J., & Kittler, J. (1988). A survey of the Hough transform. *Computer Vision, Graphics, and Image Processing*, *44*, 87–116.

Intel. (2001). *Open source computer vision library.* United-States of America.

Jacobs, J. P. (2001). Refinements to the ergonomic model for keyboard fingering of Parncutt, Sloboda, Clarke, Raekallio, and Desain. *Music Perception*, *18*(4), 505–511.

Jensenius, A. R., Kvifte, T., & Godøy, R. I. (2006, June 4–8). Towards a gesture description interchange format. In *Proceedings of the 6th international conference on new interfaces for musical expression* (pp. 176–179). Paris, France.

Kendon, A. (1986). Current issues in the study of gesture. In J.-L. Nespoulous, P. Peron, & A. R. Lecours (Eds.), *The biological foundations of gesture: Motor and semiotic aspect* (pp. 23–47). Lawrence Erlbaum Association.

Kessous, L., Castet, J., & Arfib, D. (2006, June 4–8). 'Gxtar', an interface using guitar techniques. In *Proceedings of the 6th international conference on new interfaces for musical expression* (pp. 192–195). Paris, France.

Kohler, M. (n.d.). *Vision based hand gesture recognition systems.* Retrieved October 18, 2004, from `http://ls7-www.cs.uni-dortmund.de/research/gesture/vbgr-table.html`

Koike, H., Sato, Y., & Kobayashi, Y. (2001). Integrating paper and digital information on EnhancedDesk: A method for realtime finger tracking on an augmented desk system. *ACM Transaction on Computer-Human Interaction*, *8*(4), 307-322.

Kvifte, T., & Jensenius, A. R. (2006, June 4–8). Towards a coherent terminology and model of instrument description and design. In *Proceedings of the 6th international conference on new interfaces for musical expression* (pp. 220–225). Paris, France.

Laurson, M., Erkut, C., Välimäki, V., & Kuushankare, M. (2001). Methods for modeling realistic playing in acoustic guitar synthesis. *Computer Music journal*, *25*(3), 38–49.

Letessier, J., & Brard, F. (2004). Visual tracking of bare fingers for interactive surfaces. *Seventeenth Annual ACM Symposium on User Interface Software and Technology*, *6*(2), 119-122.

Levine, M. (1985). *Vision in man and machine.* New-York, United-States of America: McGraw-Hill.

Miura, M., Hirota, I., Hama, N., & Yanagida, M. (2004). Constructing a system for finger-position determination and tablature generation for playing melodies on guitars. *Systems and Computers in Japan*, *35*(6), 10–19.

Otsu, N. (1979, January). A threshold selection method from gray level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-9*(1), 62–66.

Pardo, B., & Birmingham, W. P. (2000). Automated partitioning of tonal music. In *Proceedings of the thirteenth international Florida artificial intelligence research symposium conference.* Orlando, Florida, United States of America.

Parncutt, R., Sloboda, J. A., Clarke, E. F., Raekallio, M., & Desain, P. (1997). An ergonomic model of keyboard fingering for melodic fragments. *Music Perception*, *14*(4), 341–382.

Paschalakis, S., & Lee, P. (1999). Pattern recognition in grey level images using moment based invariant features image processing and its applications. *IEE Conference Publication*, *465*, 245-249.

Pavlovic, V. I., Sharma, R., & Huang, T. S. (1997). Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 677-695.

Picard, R. W. (1997). *Affective computing.* Massachusetts, United States of America: The MIT Press.

Pollock, J. (1999). *Using MIDI guitar.* Retrieved January 4, 2006, from `http://jpsongs.com/troubadortech/usmgtr.htm`

Pollock, J. (2000). *MIDI guitars: A nearly complete roster.* Retrieved January 4, 2006, from `http://jpsongs.com/troubadortech/mgtr.htm`

Radicioni, D. (2005). *Computational modeling of fingering in music performance.* Unpublished doctoral dissertation, Centro di Scienza Cognitiva, Università degli Studi di Torino, Torino, Italy.

Radicioni, D., Anselma, L., & Lombardo, V. (2004a). An algorithm to compute fingering for string instruments. In *Proceedings of the 2nd national congress of the associazione italiana di scienze cognitive.* Ivrea, Italy.

Radicioni, D., Anselma, L., & Lombardo, V. (2004b). A segmentation-based prototype to compute string instruments fingering. In *Proceedings of the conference on interdisciplinary musicology.* Graz.

Radicioni, D., & Lombardo, V. (2005a). Computational model of chord fingering for string instruments. In *Proceedings of the 27th annual conference of the cognitive science society.* Stresa, Italy.

Radicioni, D., & Lombardo, V. (2005b, September 21-23). A CSP approach for modeling the hand gestures of a virtual guitarist. In *Proceedings of the 9th congress of the italian association for artificial intelligence (AI*IA 2005)* (Vol. LNAI: 3673, pp. 470–473). Milano, Italy: Springer-Verlag.

Rivard, M. (1991). La complainte du phoque en Alaska. *Publication Chant de mon pays inc.*, 34–37.

Rojas, P. (2005, January). *Music thing: The SynthAxe.* Retrieved July 22, 2006, from `http://www.engadget.com/2005/01/14/music-thing-the-synthaxe/` Engadget.

Rosenfeld, A. (1969). *Picture processing by computer.* New-York, United-States of America: Academic Press.

Rossing, T. D., Moore, F. R., & Wheeler, P. A. (2002). *The science of sound, third edition.* San Francisco: Addison Wesley.

Sayegh, S. I. (1989). Fingering for string instruments with the optimum path paradigm. *Computer Music Journal*, *13*(3), 76–84.

Schulze, M. A. (2003). *Circular Hough transform a Java applet demonstration.* Retrieved October 19, 2004, from `http://www.markschulze.net/java/hough/`

Sipser, M. (1997). *Introduction to the theory of computation.* Boston, United-States of America: PWS Publishing Company.

StarrLabs. (2006). *Custom and semi-custom midi performance controllers.* Retrieved July 22, 2006, from `http://www.starrlabs.com/`

Traube, C. (2004). *An interdisciplinary study of the timbre of the classical guitar.* Unpublished doctoral dissertation, McGill University.

Traube, C., & Depalle, P. (2003). Extraction of the excitation point location on a string using weighted least-square estimation of a comb filter delay. In *Proceedings of the 6th international conference on digital audio effects.* London, United-Kingdom.

Traube, C., Depalle, P., & Wanderley, M. M. (2003). Indirect acquisition of instrumental gesture based on signal, physical and perceptual information. In *Proceedings of the 2003 conference on new interfaces for musical expression.* Montréal, Canada.

Traube, C., & Smith III, J. O. (2000). Estimating the plucking point on a guitar string.

In *Proceedings of the cost g-6 conference on digital audio effects.* Verona, Italy.

Traube, C., & Smith III, J. O. (2001). Extracting the fingering and the plucking points on a guitar string from a recording. In *IEEE workshop on applications of signal processing to audio and acoustics.* New-York, United-States of America.

Truchet, C. (2004). *Contraintes, recherche locale et composition assistée par ordinateur.* Unpublished doctoral dissertation, Université Paris 7 - Denis Diderot, UFR Informatique.

Verner, J. A. (1995). MIDI guitar synthesis yesterday, today and tomorrow. an overview of the whole fingerpicking thing. *Recording Magazine*, *8*(9), 52–57.

Vintage Synth Explorer. (2005). *ARP Avatar.* Retrieved January 4, 2006, from `http://www.vintagesynth.com/`

Wanderley, M. M., Vines, B., Middleton, N., McKay, C., & Hatch, W. (2005). The musical significance of clarinetists' ancillary gestures: An exploration of the field. *Journal of New Music Research*, *34*(1), 97–113.

Wang, J.-F., & Li, T.-Y. (1997). Generating guitar scores from a MIDI source. In *Proceedings of 1997 international symposium on multimedia information processing.*

Wikipedia. (2006). *Guitar.* Retrieved June 23, 2006, from `http://en.wikipedia.org/wiki/Guitar`

Yörük, E., Dutağaci, H., & Sankur, B. (2006). Hand biometrics. *Image And Vision Computing*, *24*(5), 483–497.