An Automatic Guitar Fingering Assessing System Based on Convolutional Neural Network and Spatio-temporal Support Vector Regression

July 2018

Zhao WANG
王　釗

An Automatic Guitar Fingering Assessing System Based on Convolutional Neural Network and Spatio-temporal Support Vector Regression

July 2018

Waseda University
Graduate School of Creative Science and Engineering
Department of Modern Mechanical Engineering
Research on Image Engineering

Zhao WANG
王　釗

Contents

Chapter 1 Introduction

1.1 Background

It is estimated that there are 20 million guitar players in the U.S. and 50 million worldwide [1]. A survey by CARAVAN [2] found that in US, 43% of adults could play an instrument and 13% could play the guitar. Furthermore, 51% of younger adults aged 18 to 34 were more likely to know how to play an instrument than older adults, and within all the younger adults who want to learn an instrument, 69% would like to learn to play the guitar. On the other hand, being a guitar teacher in the United States is not required to have any training or special certifications [3], and only 2–3% of musicians are classified as musically "literate" [4].

Actually, learning how to play the guitar is a very interesting but extremely complicated process, because it requires guitarists to perform many techniques at the same time: reading scores, pressing fretboard, sweeping or plucking strings and so on. Figure 1.1 shows some essential skills required for guitar playing. One of the most difficult skills in guitar learning is "Fingering" of the left hand, which is a cognitive process that maps each note on a music score to a fingered position of the left hand on the guitar fretboard [1]. More specifically, the fingering problem consists of two parts: (1) determine a two-dimensional position <*string, fret*> on the guitar fretboard for each note in the score, and (2) determine which finger of the guitarist's left hand should be used to press each note [6]. This thesis focuses on this left-hand fingering and reading scores as Fig 1.1 shows.

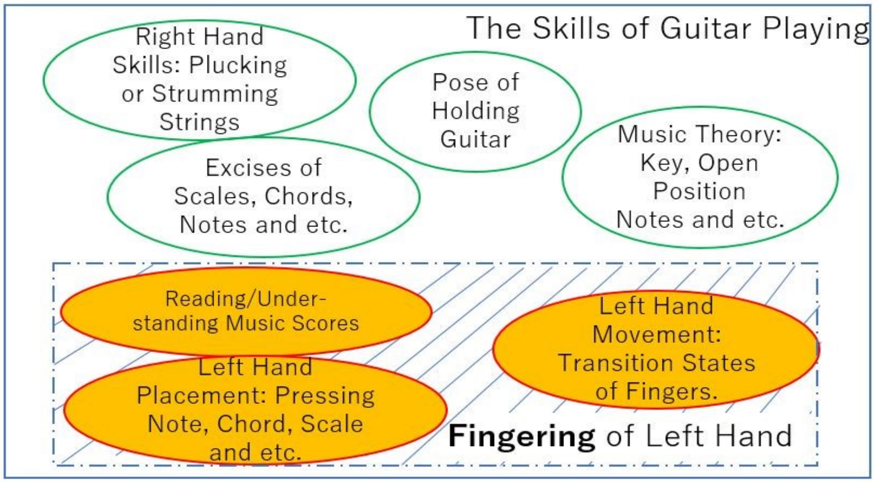

The Skills of Guitar Playing
Right Hand Skills: Plucking or Strumming Strings
Pose of Holding Guitar
Excises of Scales, Chords, Notes and etc.
Music Theory: Key, Open Position Notes and etc.
Reading/Under-standing Music Scores
Left Hand Movement: Transition States of Fingers.
Left Hand Placement: Pressing Note, Chord, Scale and etc.
Fingering of Left Hand

Fig 1.1 The Skills of Guitar Playing: the Fingering of left hand (shaded area) is the research subject of this thesis.

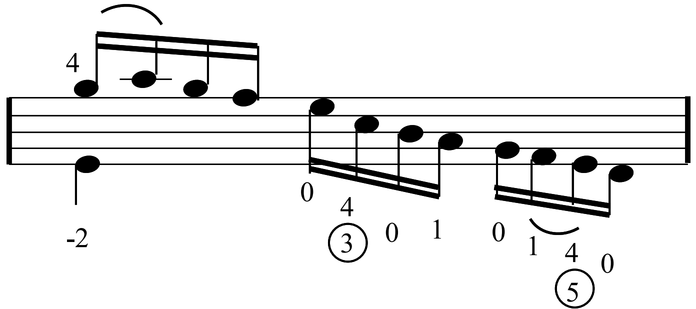


Fig 1.2 Fingered Music for Guitar: the numbers 1 to 4 indicate the stopping fingers, 0 an open note, circled numbers strings, and dashed numbers slipping. 38 possible fingering for guitarist, and within these possible fingerings, only one fingering is correct.

Reading score is one of the most important skills in guitar playing. In particular, in early stages of practicing, it is very easy for beginners of guitar to make mistakes in reading score. Unlike piano, string instruments with fixed fret positions such as guitar or violin offer alternative positions <*string, fret*> for every single note. For instance, the note *Middle C* (C in the fourth octave on a piano) is located on five alternative positions on the guitar fretboard: *<2, 1>, <3, 5>, <4, 10>, <5, 15>, <6, 20>* [6]. Furthermore, considering that each position can be accessed by any of the four left hand fingers and multiple notes are played in the score, 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, a maximum of 16^n combinations of <*string, fret, finger*> [32] can exist. Among them, the guitarist needs to read and execute the notes by selecting few correct ways (in most cases, there is only one correct fingering) to perform beautiful and elegant music. Therefore, the case "sound maybe right but fingering is wrong" exists, and because of this situation happens, it is very likely that beginners of the guitar may ignore reading the correct fingering from the score and checking whether his or her fingering is wrong as his or her playing performance sounds not bad. This situation brings a lot of harm to guitar beginners [6,7] because it must solicit them to develop a bad fingering habit of guitar playing. Figure 1.2 shows an example of guitar score. In this piece of notes, there are totally 2^{13} possible ways to play it, considering the physical constraint of the human hand, there are 38 possible fingering for guitarist, and within these possible fingerings, only one fingering is correct.

The rapid growth of computer science and Internet during these ten years has caused rapid developments in music areas. Nowadays, learning to play instruments is not limited to attending an interactive class of human teachers, online music resources (video tutorials, digital score and etc.) and computer applications (Chordify [36], Capo [37], GuitarMaster [38]) increase the possible way for beginners of instruments to find resources to help them improve skills. Compared with traditional way, the advantage of the new learning is low-cost and timesaving, but it also has its fatal limitation: no interaction

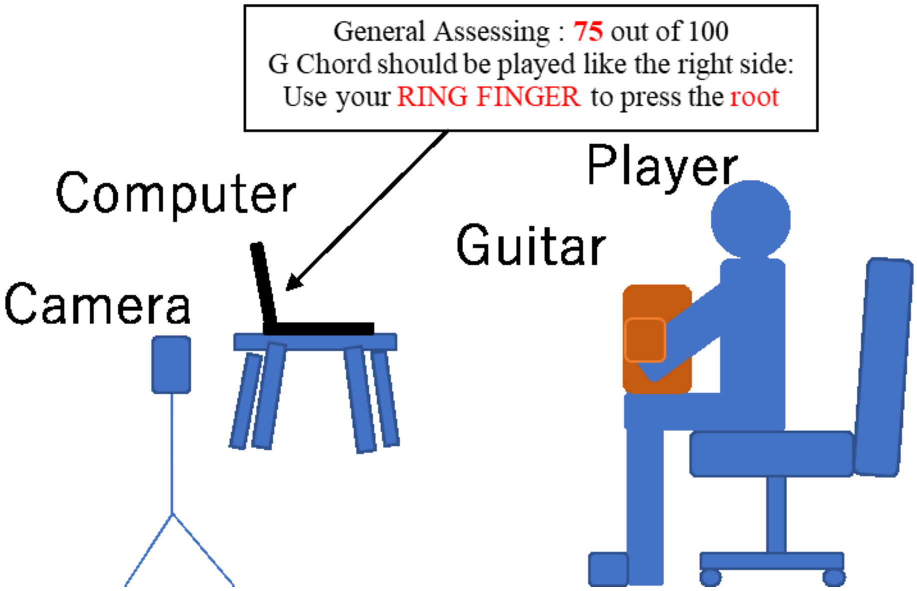

General Assessing : 75 out of 100
G Chord should be played like the right side:
Use your RING FINGER to press the root
Computer
Player
Guitar
Camera

Fig 1.3 Conceptual Image of a Simplest Guitar Fingering Assessing

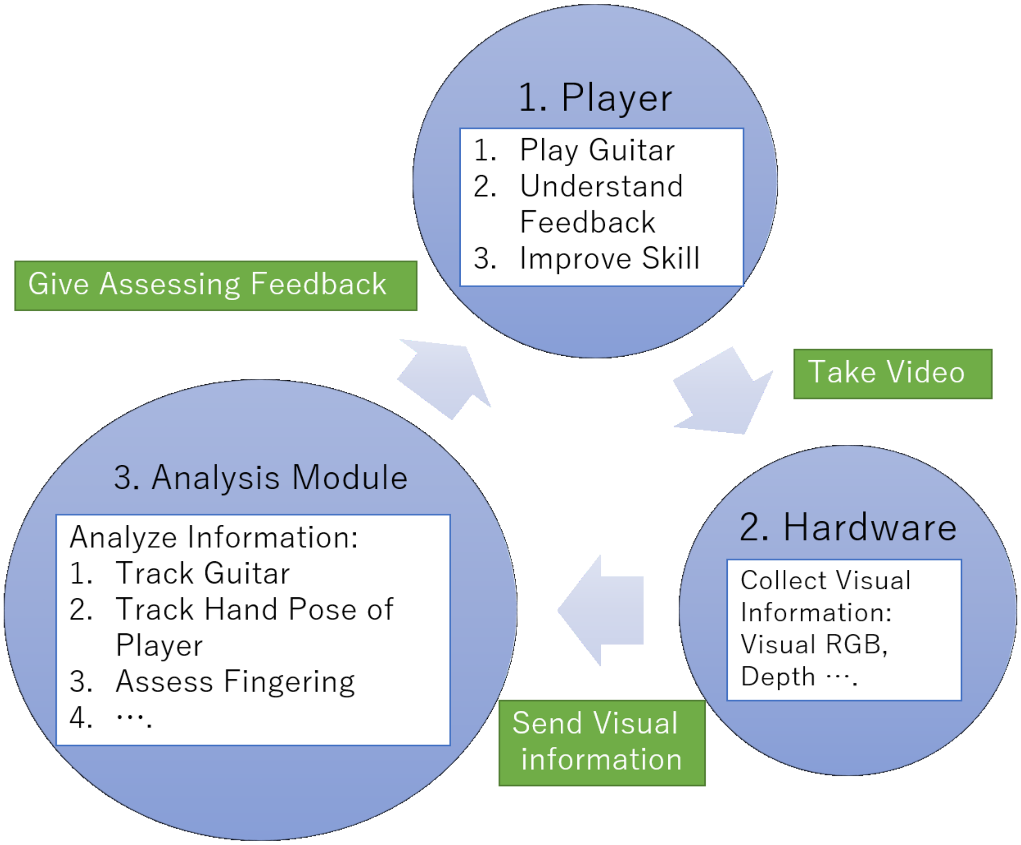


Figure 1.4 Conceptual Structure of an Guitar Fingering Assessing System.

between “teacher” and “student” [5]. Therefore, the demand for autonomous instrumental teaching system becomes greater than ever. A conceptual configuration of the simplest automatic guitar teaching system is shown in Fig.1.3: by setting up a camera and a computer in front of the guitar playing scene, the system could assess the fingering of the player, and give feedback to the player to help to improve his or her fingering.

A conceptual structure of this thesis of the guitar teaching systems is shown in Fig.1.4. The system consists of three parts: the player, the hardware and the analysis module. The hardware of the guitar teaching system is only a computer and an image sensor, which are placed in front of the user. The analysis module is the core of the system. Similar to human teachers, the analysis module needs to collect information on the analysis first. Since playing guitar is a complicated process, plenty of visual information needs to be obtained. For example, (1) guitar neck location, (2) the guitarist’s hand region, (3) hand pose (joints’ position of the hand) of the guitarist. Then, the analysis module analyzes and assesses the fingering of the guitarist based on the collected information mentioned above [8,9,10,11,12], and gives feedback to the guitar player.

This thesis deals with the analysis module of the system. However, as mentioned earlier, analyzing guitar fingering is extremely complex and complicated, from the perspective of computer vision, it involves at least three important and difficult cognitive steps in the analysis module:

(1) How to Accurately Track Guitar Neck

Guitar players tend to move their hands fast while playing guitars, which could result in occluding the guitar necks partially or entirely at almost every frame. Also, the determination of the features on the guitar neck is an issue to be solved, because the above-mentioned occlusion could happen at any place on the neck due to fast movements of the guitar player’s hand. Besides, the pixel distance between the adjacent strings of guitar is very small (within 100 pixels if the resolution of the frame is around 2000*1000 pixels). Therefore, if the guitar neck cannot be accurately tracked, it is very likely that the system wrongly assesses fingering of the guitarist.

(2) How to Accurately Track or Estimate Hand Pose

Compared with other multiple target tracking problems such as pedestrian tracking or vehicle tracking, the hand pose tracking and estimation include the following technological difficulties: (a) the hand areas of guitarists are difficult to be segmented due to different color of human skin, complex backgrounds, different illuminations etc.; (b) each finger has so similar features such as semi-circular shape of fingertips and similar skin color that it is very difficult to discriminate each fingertip during tracking; (c) during guitar plays, the fingertips move fast and do not follow any regular movement patterns such as linear rectilinear motion; while pedestrians' or vehicles' movements almost follow some rules such as moving at a nearly constant speed along a straight line or at a nearly constant acceleration, which results in facilitating tracking tasks; (4) self-occlusions and joint finger happen frequently during the guitar play, which could make it difficult to track or estimate each finger individually.

(3) How to Precisely Assess Finger Movement

Action assessing, which could automatically evaluate or score the quality of action is quite different than action recognition, and only few promising results of action-assessing algorithm of computer vision is explored and still a long way to rivaling the performance of expert judges, even the mid-level judges.

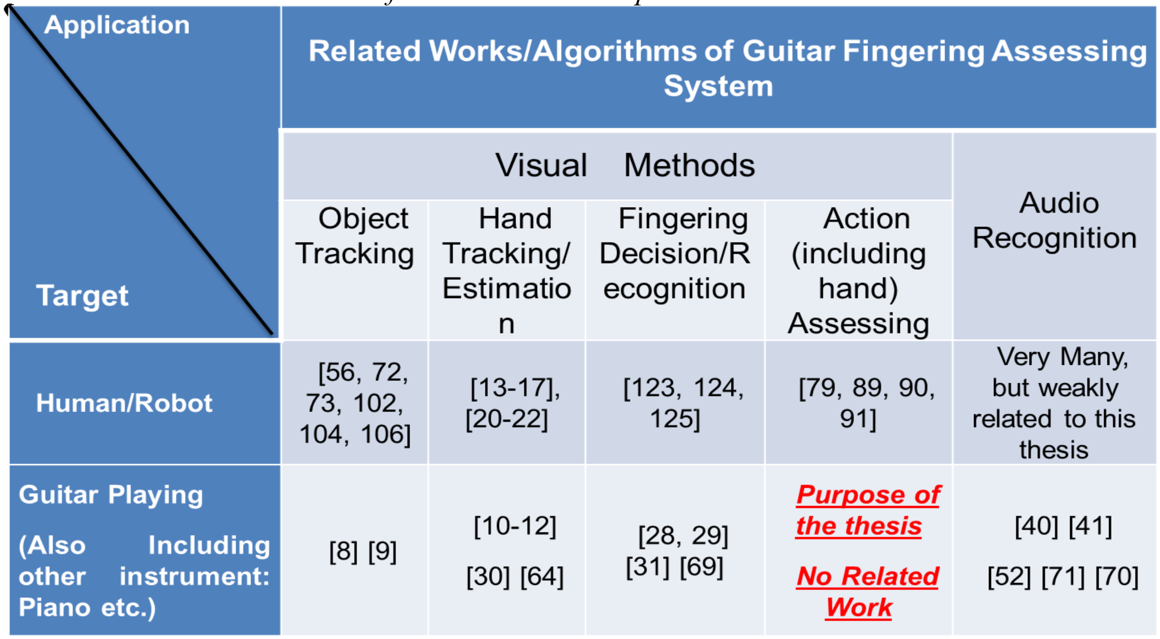
In conclusion of Section 1.1, as fingering of the guitarist is an extremely hard process to be analyzed, this thesis deals with the analysis module of the guitar fingering assessing system shown in Fig.1.4. Overview of the music related computer algorithms and applications is discussed in Section 1.2. The most advanced algorithms of computer vision that have strongly related to the system including hand segmentation, hand pose estimation, multiple target tracking, human action assessing are detailed in Chapter 2.

1.2 Related Work

Nowadays, online music resources such as Ultimate-Guitar Website [43] and computer application such as Chordify [36], Capo [37], GuitarMaster [38] are very popular to young beginners of musicians. Although these online materials provide practical functions that help beginners a lot, such as augment score [44, 45] However, the problem is that these self-teaching cannot evaluate how users performed, and users cannot actually learn from the mistakes made by themselves during the playing. With the rapid growth of machine learning technology, music application combined with signal processing (both audio and visual) attracts academic researcher's attention significantly.

Recently, with the development of image processing and machine learning, the related works of the music related computer applications have been drawing researchers' significant attentions. Such applications and algorithms about music identification, music transcription and etc. represent solutions to the problems at hand for musicians such as music identification, music transcription and so on.

Table 1.1 Related Work List of Music Related Computer Science



Application Target	Related Works/Algorithms of Guitar Fingering Assessing System				
	Visual Methods				Audio Recognition
	Object Tracking	Hand Tracking/ Estimation	Fingering Decision/Recognition	Action (including hand) Assessing	
Human/Robot	[56, 72, 73, 102, 104, 106]	[13-17], [20-22]	[123, 124, 125]	[79, 89, 90, 91]	Very Many, but weakly related to this thesis
Guitar Playing **(Also Including other instrument: Piano etc.)**	[8] [9]	[10-12] [30] [64]	[28, 29] [31] [69]	***Purpose of the thesis*** ***No Related Work***	[40] [41] [52] [71] [70]

Table 1.1 lists the works and algorithms related to this thesis. As shown in Table 1.1, in terms of functions and modules used in the guitar fingering assessing system, related works are categorized into object tracking, hand tracking or estimation, fingering decision or recognition and action assessing in the horizontal rows of Table 1.1. On the other hand, based on the different targets of the research works, the vertical column of Table 1.1 lists two general objects: for human or robot general usage and for instruments playing purpose. Note that the first row "Human/Robot" of Table 1.1 lists the related works of object tracking, hand pose tracking or estimation, fingering decision and etc. for general usage in all computer vision applications, they are detailed in Chapter 2. The detailed related works of the first row "Guitar Playing" are detailed in the remaining part of this section.

1.2.1 Guitar Neck Tracking

Y. Motokawa et al. [8] present a study that shows a learner how to correctly hold the strings by overlaying a virtual hand model and lines onto a real guitar. The player learning to play the guitar can easily understand the required position by overlapping their hand on a visual guide. Therefore, in the system the core algorithms are tracking the position of the guitar from video sequences. By using Augment Reality Tags [65] as visual markers, the system could continually track the guitar. Accordingly, it can constantly display visual guides at the required position to enable a player to learn to play the guitar in a natural manner [8]. The drawbacks of the system are obvious, (1) they use supportive tool to track the guitar neck by fixing the tool onto the guitar neck, it brings serious inconveniences to its users; (2) although the system can give the visual guide to the guitarist, it cannot judge whether the guitarist hold the right string, and obviously it cannot assess the fingering of the guitarist either.

J. Scarr et al [9] propose an algorithm that uses a markerless approach to successfully locate a guitar fretboard in a webcam image, normalize it and detect the individual locations of the guitarist's fretting fingers. Specifically, by using Hough Transform for line detection to detect the horizontal string and RGB color detection to detect the fingertips, it can recognize the chord, which is being pressed by the guitarist. The drawback of this system is (1) instead of tracking, it detects strings by using Hough Transform at every frame, which causes "jumpiness" [9] (the guitar neck cannot continually detect frame by frame) problem; (2) they also use an RGB threshold to detect fingertip positions, which causes a very low recognition rate of the chord (only 52% recognition rate for chord detection).

1.2.2 Hand Pose Tracking and Estimation for Guitarists

The earliest system that combines computer vision with the guitar is proposed by O. Cakmakci et al. [64]. Their work presents a system that assists beginner level musicians in learning the electric bass guitar. The general idea is to synthesize the bass part of a score, only one note at a time, using the sound hardware on a computer and visually put a red mark, for that note in time, on the fingerboard through a head-worn display. Upon marking a note, the system waits until the user finds that note and puts their finger on that note. Using computer vision techniques, it evaluates if the user is pressing the correct fret at that point in time [64]. The drawback of the system is (1) it only can deal with one note at a time, which could be considered unpractical because the guitar is a polyphonic instrument; (2) it requires the user to wear a head-worn display, fix a red marker of the fingertip; (3) since red color appears different RGB value in different illumination conditions, the system cannot achieve high accuracy.

A. Burns et al. [10] present a method to visually detect and recognize fingering gestures of the left hand of a guitarist. it first analyses some important findings about the design methodology of a vision system for guitarist fingering, namely the focus on the effective gesture, the consideration of the action of each individual finger. Motivated by these results, studies of three aspects of a complete fingering system are conducted: first finger tracking; second strings and frets detection; and third movement segmentation. Finally, these concepts are integrated into a prototype, and a system for left hand fingering detection is developed [10]. The drawback of the system is that (1) it requires the guitarist to fix a camera to the guitar head, so the guitar and camera are relatively static; (2) it segments the hand area of the guitarist by setting the RGB value, which cannot be practically applied under different illumination situations.

C.Kerdvibulvech et al [11, 12] propose a stereo-camera based algorithm that first estimates the projection matrix of each camera by utilizing Augmented Reality Tag (ARTag) [65]. ARTag's marker is fixed to the guitar neck. Therefore, the world coordinate system is defined on the guitar neck as the guitar coordinate system so that the system allows the players to move the guitar while playing. They also utilize a particle filter [66] to track the finger color markers in 3D space, where propagate sample particles are in 3D space and projected onto the 2D image planes of both cameras to get the probability of each particle to be on the finger markers based on RGB color in both images. In addition, they refine the work by tracking the fingertips without using color marker. They develop another particle filter-based tracking algorithm [67] by detecting the semi-circle shape of the fingertips as features. However, since (1) they use RGB value as a threshold for hand region segmentation, (2) they do not develop a mechanism to deal with the problems of the finger movement in guitar playing, such as occlusion, joint finger and etc. mentioned as Section 1,2, it achieves a low tracking accuracy (experimental results are shown in Section 4.5).

S. Manitsaris et al [30] proposes a marker–less computer vision methodology for the simultaneous recognition of complex finger musical gestures performed in space without any tangible musical instrument. Image analysis techniques are applied in order to detect and identify the fingertips in the video. The finger gesture recognition and prediction are based on the stochastic modelling of the extracted high–level features: Hidden Markov models and Gaussian mixture models [30]. However, their system only detects fingertips at each frame of the input video to recognize the gesture of the musician, but it cannot assess the movement of the fingers and give general evaluation of feedback to its users.

1.2.3 Fingering Decision and Recognition

The problem of fingering decision and recognition for polyphonic transcription can be formally described as the transformation of a time ordered sequence of audio or video samples into a set of tuples describing start, end, fundamental frequency or pitch and optionally amplitude of the notes that are played [40]. The fingering decision and recognition is a fundamental and very popular problem in academic research [52] because it involves the cognitive process of fingering: it has to determine that the <*string, fret, finger*> combinations for each note in the audio or video samples in time sequence.

Several recent polyphonic music transcription systems utilize deep neural networks [40, 41, 52] to achieve state of the art results. R. Kelz et al [40] provide a detailed analysis of the particular kinds of errors called "glass ceiling" that state of the art deep neural transcription systems makes. S. Sigtia [52] also present a supervised neural network model for polyphonic piano music transcription. The architecture of the model is analogous to speech recognition systems and comprises an acoustic model and a music language model. The acoustic model is a neural network used for estimating the probabilities of pitches in a frame of audio. The language model is a recurrent neural network that models the correlations between pitch combinations over time [52].

P. Suteparuk [28] proposes an automated approach for visually detecting and tracking the piano keys played by a pianist using image differences between the background image and video frame and transcripting playing scene to music score. M. Akbari et al. [29] propose an innovative computer vision-based automatic music transcription system named claVision to perform piano music transcription. By developing a four-stage process: keyboard detection, background update, keys detection, and hand detection, it automatically transcripts piano playing scene to music score in real time. Recently, CNN based method [31] present a new real-time learning-based system for visually transcribing piano music using the CNN-SVM classification of the pressed black and white keys. The whole process in this technique is based on visual analysis of the piano keyboard and the pianist's hands and fingers,

A.Aggarwal [69] introduces a novel monophonic audio-visual approach of automatic music transcription, with the purpose of aiding monophonic music compositions [69]. By using an audio-based C-Support Vector Classifier [76] and computer vision based partial guitar neck detection [9, 10], it tests on the dataset of annotated 10,800 samples of audio and images, respectively. The multi-modal analysis carries out with high accuracy of transcription rate for watching note. Furthermore, the data has been released online and made available for download free of cost [69]. However, since it is a monophonic labelled dataset, it is not practical for guitar users as guitar is a polyphonic instrument.

1.2.4 Summary

This sub-section summarizes the related work as follows.

Quite many methods that combined music and computer science are studied in academic research. Based on functions of all the algorithms listed in Table 1.1, they are categorized into music identification, music transcription, fingering recognition and fingering assessing. Within these four categories, fingering recognition and fingering assessing are the most related to this thesis.

Computer vision based guitar fingering recognition and fingering assessing methods show many drawbacks of research approaches: (1) some methods require the guitarist to use supportive tools, such as head-worn camera, color marker, AR Tag and etc. These tools are inconvenient to guitar playing; (2) some works cannot work under complex circumstance, such as different illumination situations, because they use a fixed RGB threshold to segment hand region of players concerning the diversity of human skin appearance, complex background and etc.

Among the related works of guitar fingering recognition and fingering assessing, based on Table 1.1, there is no previous works on guitar fingering assessing system, i.e. no work can assess the fingering of guitar, and give feedback such as general evaluation of the performance to the guitarist.

1.3 Purpose of the Thesis

Towards the actualization of a guitar fingering teaching system that can autonomously assess the fingering of the guitarist, as discussed in Section 1.1 and 1.2, this thesis aims at developing a computer vision and machine learning based system that can assess the left-hand fingering of guitarists by recognizing the pressed chord at some specific frames and training a regression model to give scores (evaluations) based on the transition of the finger pose of the guitarists to help them improving skills. The specific purpose (1) to (3) of this thesis is described as follows:

(1) Accurately Tracking Guitar Neck

As mentioned in Section 1.1, the guitar neck needs to be accurately tracked for the subsegment assessing of the guitarist's fingering. More specifically, it **is required that the mean track error of the guitar neck should be less than half of the distance between adjacent strings (8mm) in the input video.**

(2) Accurately Tracking Left Hand Pose of Guitarist.

Considering the diversity of performing manners and individual differences of guitar playing, hand pose of the guitarist should be tracked or estimated via two ways: (a) **deep learning-based hand segmentation and hand pose estimation** (b) **data association based multiple fingertip tracking** to achieve high assessing accuracy in (3).

(3) Training-based Left Hand Fingering Assessing with A Regression Model

Instead of manually designing an evaluation function for guitar fingering, the off-line training-based scoring result without human intervention is preferable as (1) data-driven, offline training method can be generalized to any human action assessing problem, while evaluation function-based method requires researcher to define each single action rules for the action assessing (2) data-driven, offline training method is more possible to achieve accurate, fair assessing result. Therefore, **off-line training and data-driven based assessing without human intervention is required.**

The comparison between the system and human judge should be done. The general assessing result of the system should be **more accurate than mid-level player of guitar** in order to show the effectiveness of the system

(4) No Supportive Tools and No Constraint of Guitar Playing Circumstance

Unlike related works [8, 10, 11, 12], the system does **not use any supportive tools such as wearable sensors, AR tag, color markers and etc**. to track or estimate the guitar neck or hand pose. As mentioned in Section 1.1, wearing sensors or fixing supportive tools is inconvenient to guitar playing. Besides, the system should work effectively **under any complex background and different illumination situations.**

1.4 Proposed Approach

The proposed approaches that tackle the four issues described in Section 1.3 are shown in Fig.1.5. The guitar fingering assessing system consists of the three main modules: Guitar Neck Tracking (Module 1), Finger Pose Estimation (Module 2) and Fingering Assessing (Module 3).

[Module 1] Guitar Neck Tracking

The guitar neck tracking proposes an algorithm for accurately and robustly tracking the 3D position of the fretboard of guitar from the video of guitar plays. First, the guitar neck area is extracted from the first frame of the input video by extracting the rectangles formed by guitar strings and frets. Second, three types of feature points (SIFT, SURF, Shi-Tomasi) are detected respectively within that fretboard area. Then, from the second frame, SIFT/SURF features are detected on every frame and matched with the SIFT/SURF detected at the first frame by using a KD-Tree based searching method to accelerate matching efficiency. On the other hand, Shi-Tomasi features are tracked frame by frame by using optical flow. Furthermore, the method filters out the mismatched SIFT/SURF feature point pairs and the mistracked Shi-Tomasi due to the occlusion problem between the first frame and the current frame by implementing a modified RANSAC mechanism. In addition, the method obtains the perspective transform matrix based on correctly matched SIFT/SURF/Shi-Tomasi pairs so that the guitar neck is tracked correctly based on the perspective transformation matrix.

[Module 2] Finger Pose Estimation

Module 2 includes the following two processes: (1) 2D Fingertip Tracking and (2) 3D Hand Pose Estimation.

(1) 2D Fingertip Tracking

After inputting the tracking result of the guitar neck by Module 1, a CNN-based hand segmentation net is used to discriminate the hand area from the background. Then, the template matching and reversed Hough Transform are performed to the hand areas so that the count map for fingertip candidates is generated using the segmentation result, where the results of the template matching and reversed Hough Transform are used as weighted features to extract the fingertip candidates. Furthermore, a temporal grouping is applied to remove noise and group the same four fingertips (index finger, middle finger, ring finger, little finger) on the successive count maps. Then, an ROI association algorithm is utilized to associate the four

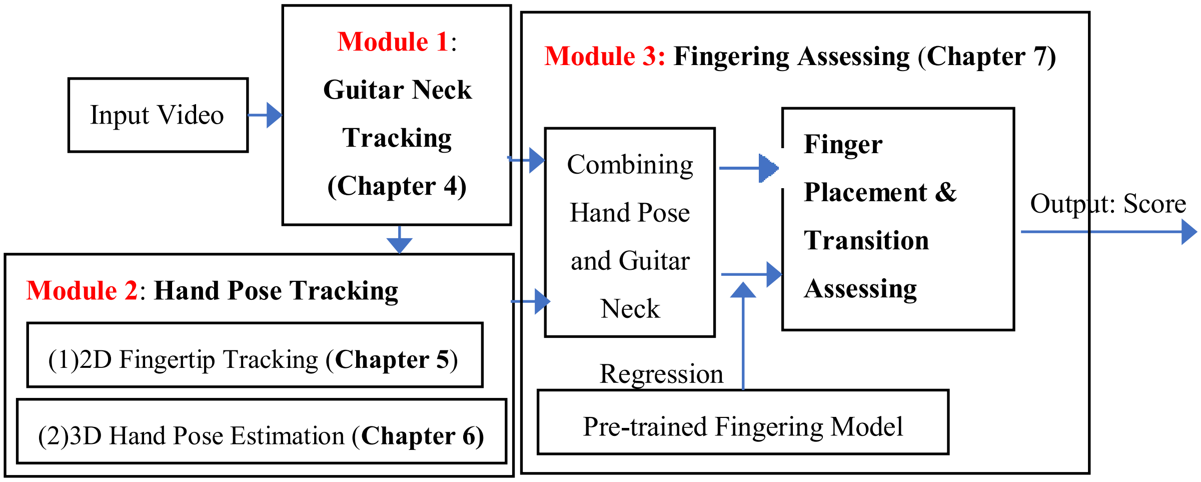

Input Video
Module 1: Guitar Neck Tracking (Chapter 4)
Module 2: Hand Pose Tracking
(1)2D Fingertip Tracking (Chapter 5)
(2)3D Hand Pose Estimation (Chapter 6)
Module 3: Fingering Assessing (Chapter 7)
Combining Hand Pose and Guitar Neck
Finger Placement & Transition Assessing
Output: Score
Regression
Pre-trained Fingering Model

Figure 1.5 Conceptual Diagram of Guitar Fingering Assessing

fingertips with their individual trajectories in the frame-by-frame count maps. Here, for this ROI association algorithm, three patterns for tracking fingertips movement during the whole process are defined: the active pattern, adding pattern, vanishing pattern. All the tracked trajectories of fingertip candidates are fitted into one of these three patterns in order to solve the problem such as self-occlusion etc. Finally, ROI associated particle filter is utilized to track the fingertips by distributing particles within the associated ROIs of the fingertips at every two adjacent frames of the video

(2) 3D Hand Pose Estimation

Parallel to the 2D Fingertip tracking, the 3D hand pose estimation based on CNN is performed. After inputting the tracking result of the guitar neck, the same CNN-based hand segmentation net mentioned in (1) is used to discriminate the hand area from the background. Then, the hand region of a 128*128-pixel region is cropped by extracting from the depth input and the segmentation mask. After that, the longest contour on the segmentation result is traced to get the hand region area. Furthermore, the depth values of the cropped image are normalized to [-1,1]: the deepest pixels in cropped image, i.e. background is set to 1, and the nearest pixels in hand region are set to -1. Finally, a convolutional neural network with three convolutional layers, two max pooling layers, and four fully connected layers is trained to output the 3D positions of 16 joints of the guitarist's hand.

[Module 3] Fingering Assessing

After acquiring the 2D fingertip position in (1) and 3D hand pose with 16 joints of guitarist in (2) of Module 2, both of them are inputted to a regression model of SVR (Support Vector Regression) using the feature of 3D spatio-temporal DCT (discrete cosine transformation) to get the comprehensive assessing result of the fingering of the guitarist.

1.5 Organization of the Thesis

In this thesis, each chapter is detailed as shown in Fig 1.6.

Chapter 1 is the introduction of this thesis. The background, related work about music related computer technology, purpose and approach of this thesis are described.

Chapter 2 details the related works of the most advanced algorithms of computer vison related to the thesis and discusses their advantages and disadvantages in case that the related works are applied to the guitar fingering assessing system.

Chapter 3 discusses the fundamental knowledge of guitar left-hand fingering, defines the coordination systems used in this thesis and presents the code table of this thesis.

Chapter 4 explains the proposed method for tracking the guitar neck in 3D. Experimental results show that the validity of this proposed method. This proposed method is presented in two reviewed international conferences [5,6]

Chapter 5 and Chapter 6 explain the proposed method for finger pose estimation module. Specifically, Chapter 5 explains 2D fingertip tracking algorithm, and Chapter 6 explains 3D finger pose estimation algorithm. Experimental results confirm the validity of the proposed method. The finger pose estimation module is presented in two reviewed international conferences [2,3] and under reviewing process as the author's journal [1].

Chapter 7 explains the proposed method for fingering assessing module. Experimental results confirm the validity of the proposed method. This method is presented in a reviewed international conference [4].

Chapter 8 concludes this thesis and states future work.

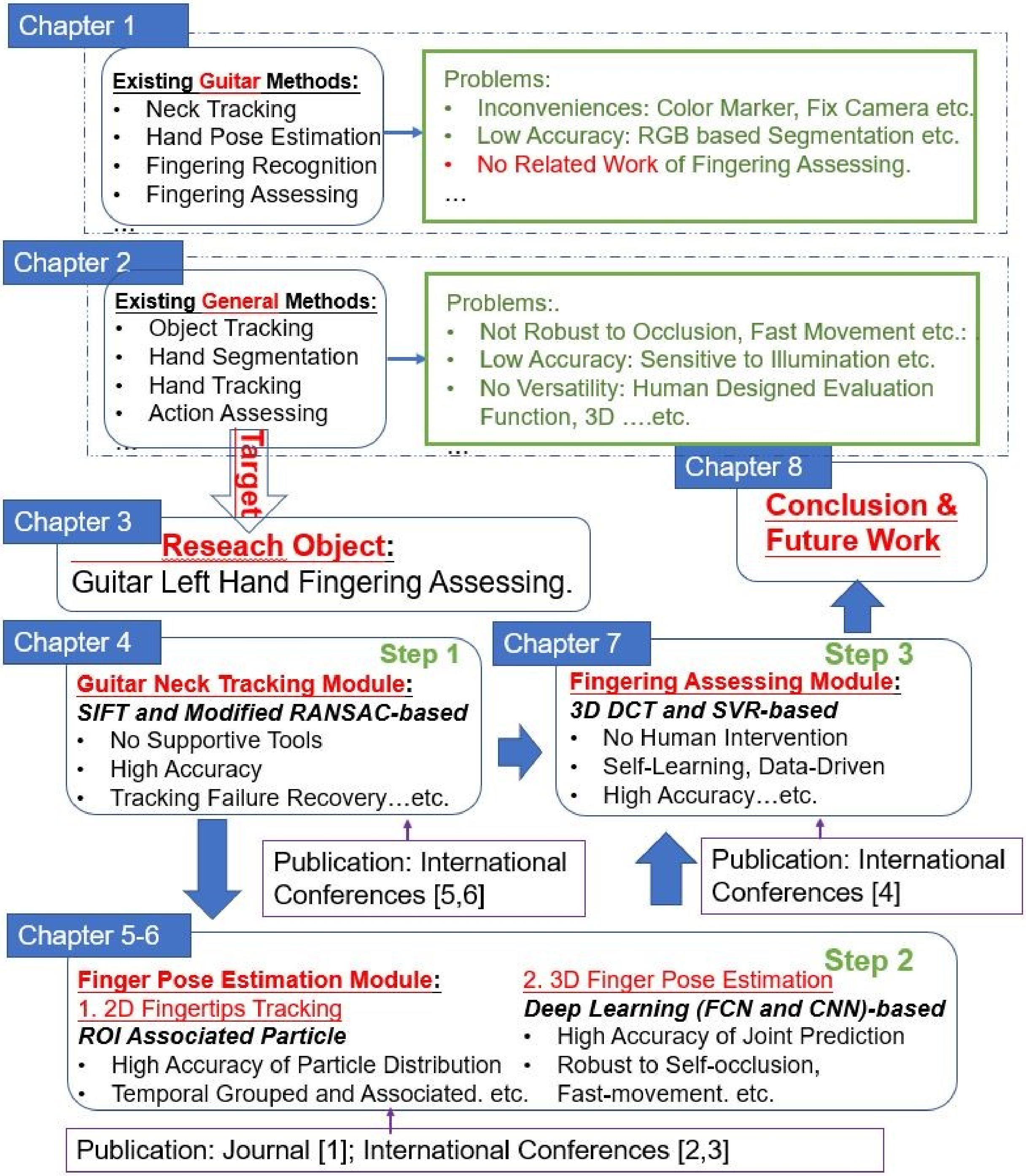

Chapter 1
Existing Guitar Methods:
• Neck Tracking
• Hand Pose Estimation
• Fingering Recognition
• Fingering Assessing
…
Problems:
• Inconveniences: Color Marker, Fix Camera etc.
• Low Accuracy: RGB based Segmentation etc.
• No Related Work of Fingering Assessing.
…
Chapter 2
Existing General Methods:
• Object Tracking
• Hand Segmentation
• Hand Tracking
• Action Assessing
…
Problems:.
• Not Robust to Occlusion, Fast Movement etc.:
• Low Accuracy: Sensitive to Illumination etc.
• No Versatility: Human Designed Evaluation Function, 3D ….etc.
…
Target
Chapter 3
Reseach Object:
Guitar Left Hand Fingering Assessing.
Chapter 8
Conclusion & Future Work
Chapter 4
Step 1
Guitar Neck Tracking Module:
SIFT and Modified RANSAC-based
• No Supportive Tools
• High Accuracy
• Tracking Failure Recovery…etc.
Publication: International Conferences [5,6]
Chapter 7
Step 3
Fingering Assessing Module:
3D DCT and SVR-based
• No Human Intervention
• Self-Learning, Data-Driven
• High Accuracy…etc.
Publication: International Conferences [4]
Chapter 5-6
Step 2
Finger Pose Estimation Module:
1. 2D Fingertips Tracking
ROI Associated Particle
• High Accuracy of Particle Distribution
• Temporal Grouped and Associated. etc.
2. 3D Finger Pose Estimation
Deep Learning (FCN and CNN)-based
• High Accuracy of Joint Prediction
• Robust to Self-occlusion, Fast-movement. etc.
Publication: Journal [1]; International Conferences [2,3]

Fig 1.6 Organization of This Thesis

Chapter 2 Related Work

Table 1.1 details the related works that are highly relevant to this thesis: fingering recognition, fingering assessing and discuss their advantages and disadvantages. In this Chapter, the current and the most advanced computer vision-based algorithms related to this thesis are discussed. More specifically: related works of (1) object tracking algorithm which could be applied to guitar neck tracking, (2) hand region segmentation which could be applied to guitarist's hand region, (3) multiple targets tracking algorithm and hand pose estimation which could be applied for tracking or estimating finger pose of guitarist, (4) human action assessing algorithm which could be applied for fingering assessing are detailed in this chapter.

2.1 Related Works of Object Tracking

Normally, computer vision based single object tracking system can be classified into three types: point tracking, kernel tracking and silhouette tracking [72, 73]. In point tracking, objects are detected in consecutive frames and represented by feature points. By tracking the state or the appearance of the feature points, objects can be tracked. In kernel-based tracking, a kernel of the object also refers to the object shape and appearance. For example, the kernel can be a rectangular template or an elliptical shape with an associated histogram. Objects are tracked by computing the motion of the kernel in consecutive frames. In silhouette tracking, normally it is performed by estimating the object region at each frame. Silhouette tracking methods use information encoded inside the object region. This information can be in the form of appearance density and shape models which are usually in the form of edge maps. Given the object models, silhouettes are tracked by either shape matching or contour evolution [72, 73].

Table 2.1 lists the subcategories of the three tracking types, and their advantages and drawback. The subcategories are explained in the following.

2.1.1 Point based Tracking Algorithm.

Point based tracking is one of the most accurate and classical methods for tracking. In consecutive frames, a moving object is tracked by tracking the points that can represent the object. The two subcategories of point tracking method are: a. feature point-based method and b. statistical method.

a. Feature point-based method

Table 2.1 Related Works of Tracking

Categories	Representative Work	Adavatanges & Drawbacks
Point Tracking		
Feature Point Methods	SIFT, SURF, Shi-Tomashi	Most accurate, but time -consuming, unrobust when occlusion
Statistical methods	Kalman Filter Particle Filter	Accurate, but suitable for motion model perdiction
Kernel Tracking		
Template and density based appearance models	Mean-shift KLT Template Matching	Not accurate, slow computation Fast computation but not accurate Extremly slow Computatuin
Multi-view appearance models	Eigentracking SVM tracker	Accurate but require multiple cameras.
Silhouette Tracking		
• Contour evolution	State space models, Variational methods, Heuristic methods	Fast, but suitable for rigid object and non-occlusion.

Feature point [72, 73, 56] is identified as an expressive texture in their respective localities in image [72]. Commonly used interest point detectors include Shi-Tomasi [99], SIFT detector [56] and SURF detector [101]. To find the feature point in image, Shi-Tomasi [99] computes the first order image derivatives, $(I_x \, , I_y)$, in the x and y directions to highlight the directional intensity variations, then a second moment matrix, M, which encodes this variation, is evaluated for each pixel in a small neighborhood of the image [72]:

$$M = \begin{pmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_{xy} I_y \end{pmatrix} \tag{2.1}$$

Then the Shi-Tomasi feature points are identified using the determinant and the trace of M in Eq. (2.1) which measures the variation in a local neighborhood by thresholding $R,$ which is the minimum eigenvalue of M in Eq. (2.1) [72, 99]. However, it is not invariant to affine or projective transformations.

In order to introduce robust detection of interest points under different transformations [72], SIFT algorithm [56] extracts discrete features (or keypoints) corresponding to locations on images that can be reliably identified upon varying viewpoint and scale [98]. Compared with other feature points widely used in computer vision method such as Shi-Tomasi [99], SIFT is widely used in applications particularly in medical image processing [98], object tracking [100]. SIFT is known as an accurate and robust algorithm that generates local features robust to changes in image scale, noise, illumination and local geometric distortion [100]. In computer vision, speeded up robust features (SURF) [101] is another local feature detector and descriptor used for tasks such as object recognition, image registration, classification or 3D reconstruction. The standard version of SURF is inspired by SIFT and works several times as fast as SIFT [101].

SIFT and SURF are considered as one of the most accurate feature points [104] because it computes the feature across several scale of image. However, in tracking application, it is not robust to the occlusion situation: once the occlusion happens, the feature points cannot be captured anymore. In the guitar playing, especially in guitar neck tracking module, it is very hard to track the guitar neck only by applying feature point-based tracking if the neck is partially occluded by the hand of the guitarist at every frame of guitar playing.

b. Statistic Methods

Kalman filter [102] is one of the earliest methods applied in tracking of computer vision and it performs the restrictive probability density propagation. One limitation of the Kalman filter is the assumption that the state variables are normally and linearly distributed; in other words, the state of the movement of the object has to be non-Gaussian and non-linearity. This limitation can be overcome by using another statistic method: particle filter [104]. The particle filtering generates all the models for one variable before moving to the next variable. Algorithm has an advantage when variables are generated dynamically and there can be unboundedly numerous variables. It also allows for new operation of resampling. The particle filter [104] is a Bayesian sequential importance Sample technique, which recursively approaches the later distribution using a finite set of weighted trials. It also consists of fundamentally two phases: prediction and update as same as Kalman Filtering [103].

Both the Kalman Filter and Particle Filter are accurate because they both model the movement of the object during the tracking, and their prediction of the movement. However, in guitar playing situation, the movement of fingers of the guitarist cannot be categorized into any physical model, such as linear movement with constant speed or etc. Besides, particle filter is suitable for a relative small object in images for tracking e.g. pedestrian because it distributes particles all over the whole image. In guitar playing, the guitar neck is a relatively large area within the image, it costs much time for distributing plenty of particles to track the guitar neck.

2.1.2 Kernel based Tracking Algorithm.

Kernel tracking [105] is usually performed by computing the moving object, which is represented by an embryonic object region, from one frame to the next. The object motion is usually in the form of parametric motion such as translation, conformal, affine, etc. [103].

Kernel tracking is categorized into some sub-classes. In this thesis, Template Matching, Mean-Shift and SVM tracker are discussed.

1. Template Matching

Template matching [105, 106] is one of the most classic image processing methods to compare the reference image and the source image, in order to locate the reference image in the source image to locate the object. The matching processing is conducted by scanning the reference image from left to right, from top to bottom of the source image. Traditional template matching has some demerits: (1) it cannot solve the occlusion problem. For example, once the object is occluded in source image, the matching result would become inaccurate; (2) it is not robust to the scale change and rotation of the target object. The two above problems are obviously very frequently happed in guitar playing scenes, for example, the guitar neck is always occluded by the hand of the guitarist, and the hand of the guitarist may have self-occlusion problems when playing guitar. Another one of the most serious problem is that, in guitar neck tracking, the reference image of the guitar neck needs to be as large as the guitar neck in the video, which causes the huge computation and memory.

2. Mean-Shift Method

Mean-shift [103] is widely used in image processing areas, such as image clustering, segmentation and tracking. The most basic idea of mean-shift is the mean values of the translation movement of the object. However, the mean-shift algorithm used in recent related works is a process of an iteration: calculating the mean value of the shift of the current point, and the calculating the new point’s shift based on the calculated value and so on. In tracking algorithm, the partial image region is tracked by calculating the histogram iteratively. A gradient ascent procedure is used to move the tracker to the location that maximizes a similarity score between the model and the current image region. In object tracking algorithms target representation is mainly rectangular or elliptical region. It contains target model and target candidate. The target model is generally represented by its probability density function (pdf) and regularized by spatial masking with an asymmetric kernel [103]. However, in the guitar fingering assessing system, the guitar neck cannot be calculated by mean-shift because the calculation process cannot give a very accurate tracking result. As mentioned in Chapter 1, the guitar neck needs to be tracked accurately since the guitar neck tracking part is the first process of the whole system, and if the result is not accurate, it is very hard to further the research.

3. Support Vector Machine (SVM)

SVM [13] is a broad classification method which gives a set of positive and negative training values. For SVM, the positive samples contain the tracked image object, and the negative samples consist of all things that are not tracked. It can handle a single image, and partial occlusion of an object, but it requires a physical initialization and training [103]. Recently, SSVM (Struct SVM) is widely used for tracking applications of image processing. However, the optimization limits of the SVM makes the tracking very hard to be trained. In particular, in guitar fingering assessing system, the heavy training process makes the guitar neck tracking and the hand tracking hard to converge, because the guitar is played indoor, and the illumination changes make the tracking problems needing huge amount of data, which is impossible to further the research.

2.1.3 Silhouette Tracking

Some objects have complex shape such as hand, fingers, shoulders that cannot be well defined by simple geometric shapes. Silhouette based methods [105] afford an accurate shape description for the objects. The aim of a silhouette-based object tracking is to find the object region at every frame by means of an object model generated by the previous frames. Contour tracking methods [105] iteratively progress a primary contour in the previous frame to its new position in the current frame. However, in the guitar playing scene, the hand contour is very hard to be tracked by using silhouette tracking method, because the hand of the guitarist moves too fast during the guitar playing, and the physical change in the hand shape are drastic, which makes it impossible to track the whole movement of the hand of the guitarist.

2.2 Related Works of Hand Segmentation

Hand segmentation is a difficult problem in computer vision, because hands can have different appearances in different images depending on various conditions. Traditional image processing methods [94,95] suggest using YUV color space or texture of hands can generate better result than RGB color space.

However, color and texture vary according to the lighting conditions, and presence of shadows, and the human hand may assume many shapes depending on the angle of the camera view and the posture of the hand [74]. All these properties can vary from people-to-people depending on gender, ethnicity, age, skin type etc.

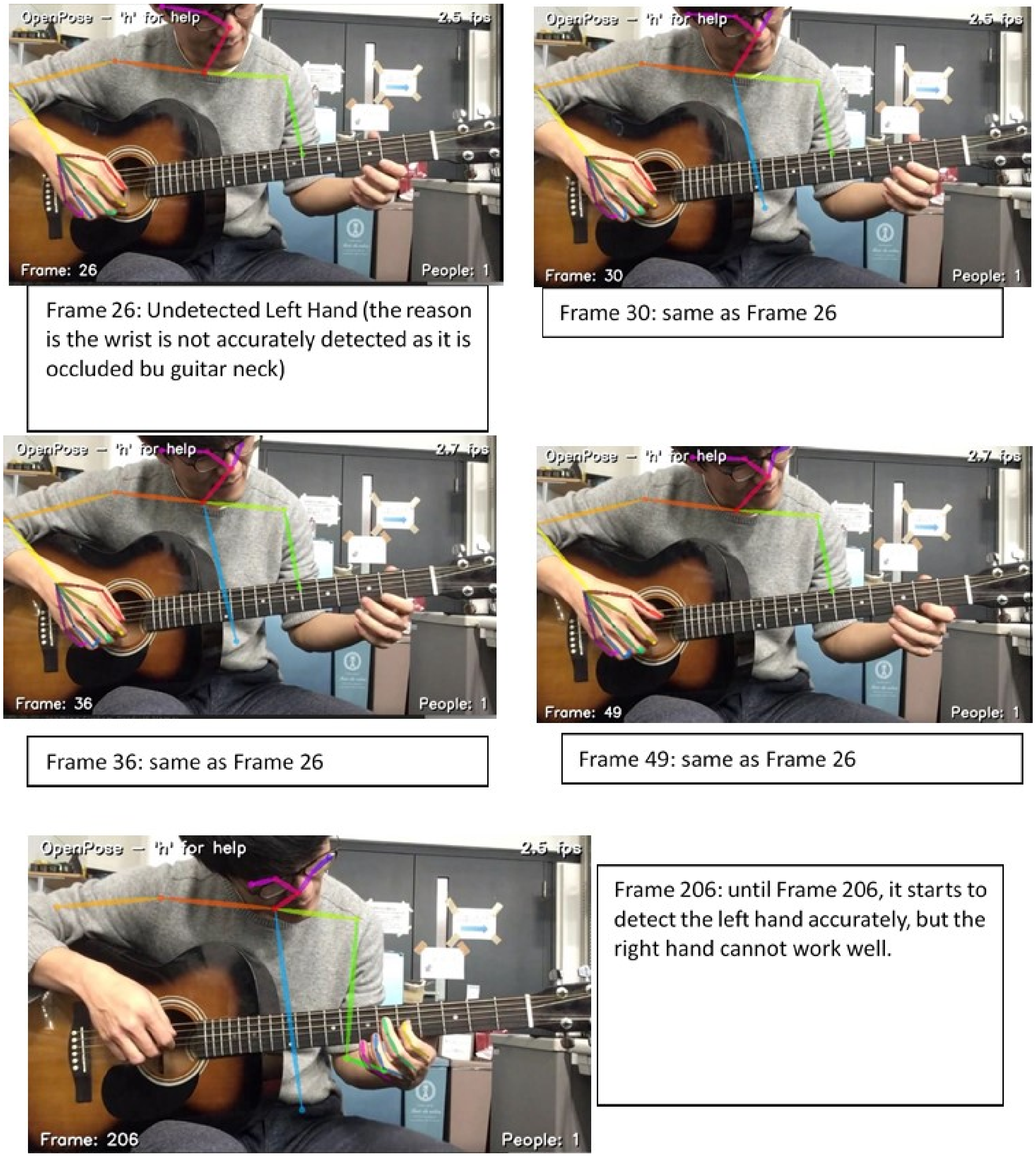
Depth cameras bring now an easy solution to handling occlusions, however, they provide a poorly accurate 3D reconstruction of the boundaries of the hand, when users hold something in their hand (in the guitar playing case, guitar players hold guitar at every frame during their playing), as the depth value of the hand and the object are same. [75]

Recently, neural networks have been successfully applied to the problem of semantic segmentation of a broad range of real world objects and scenes. Popular methods include convolutional neural network [75], which extracts features with convolutional layers without using pooling layers and map them directly to the output segmentation with a fully connected layer achieves 94% accuracy of pixel-wised accuracy. However, compared with other deep learning-based segmentations [96,97], 94 % of the accuracy is a fairly low accuracy, because the Segnet [96] or FCN [97] achieves 98% of the accuracy for semantic segmentation. Besides, as it [75] outputs the possibility of every pixel to be hand region pixel or not, both training and testing efficiency are not good enough.

2.3 Related Works of Hand Pose Tracking and Estimation

In the guitar playing assessing system, accurate hand pose estimation is another task in guitar fingering system as the fingering assessing module needs to combine the spatiotemporal information of both the guitar neck and finger pose.

Compared with other tracking problems such as pedestrian tracking or vehicle tracking , the hand pose is very complex and complicated to analyze due to: (1) in some of other multiple target tracking or pose estimation such as pedestrian tracking, individual cars or people have different appearances (clothes, color, car shape, etc.); while in finger pose tracking or estimation, each finger has so similar features such as semi-circular shape and similar skin color that it is very difficult to discriminate each finger during tracking; (2) during guitar plays, the fingertips move fast and do not follow any regular movement patterns such as linear rectilinear motion; while pedestrians' or vehicles' movements almost follow some rules such as moving at a nearly constant speed along a straight line or at a nearly constant acceleration, which results in facilitating tracking tasks; (3) self-occlusions and frame-out of fingertips tracking tasks; (3) self-occlusions and frame-



Frame 26: Undetected Left Hand (the reason is the wrist is not accurately detected as it is occluded bu guitar neck)

Frame 30: same as Frame 26

Frame 36: same as Frame 26

Frame 49: same as Frame 26

Frame 206: until Frame 206, it starts to detect the left hand accurately, but the right hand cannot work well.

Fig 2.1 Related Works of 3D Hand Pose Estimation Based on CNN

out of fingertips happen frequently during the guitar play because the camera is placed only in front of the guitarist, which makes it very difficult to track or estimate each finger individually.

Recently, 2D or 3D hand pose tracking and estimation from an RGB camera and depth sensor has attracted image processing and machine learning researchers [13-20], because it plays a very important role in the research of human-computer interface (HCI) and augmented reality applications (AR) [86]. In this section, the related works of hand pose estimation and the related works of hand tracking are described in Section 2.3.1 and Section 2.3.2 respectively, because they have two different approaches: the tracking method aims at associating each detected finger between frames while the hand pose estimation method concentrate on estimating the hand pose by each individual method. However, since both of them can output the hand pose information at every frame of the input video, they both need to be included in the related works of the guitar fingering system.

2.3.1 Hand Pose estimation

For the hand pose estimation, the most recent and representative works [13-17, 19] solve the problem by utilizing either on multi-layered Random Forests [14, 17] or CNN [13, 15, 18] or hybrid approach [16]

D.Tang et al. [14] directly regresses the 3D locations of the joints, using a hierarchical model of the hand. Specifically, it formulates the 3D hand pose estimation problem as a divide-and-conquer search for skeletal joints: it starts by taking the whole hand as input, recursively dividing the input region into two parts that defined the topological model, until all skeletal parts are located. As the samples propagate down the LRF (latent regression forest), the patch size shrinks from the whole hand to smaller local regions [14]. However, LRF based methods that shrinks the image from the whole resolution to small local region to regress the 3D coordination of joints cannot directly infer the locations of hidden finger, because the local feature of fingers cannot be found once occluded, and the self-occlusion situation (one finger is occluded by another finger) frequently happens in guitar play situations.

C.Keskin et al. [17] introduce a novel randomized decision forest (RDF) based hand shape classifier and use it in a novel multi–layered RDF framework for articulated hand pose estimation. This classifier assigns the input depth pixels to hand shape classes and directs them to the corresponding hand pose estimators trained specifically for that hand shape. The method introduces two novel types of multi–layered RDFs: Global Expert Network (GEN) and Local Expert Network (LEN), which achieve significantly better hand pose estimates than a single–layered skeleton estimator and generalize better to previously unseen hand poses [17]. However, their multi-layered approach accumulates errors, causing huge finger estimating errors for training loss. Furthermore, it cannot deal with the self-occlusion effectively enough, and once two finger joints together, it either cannot estimate hand pose accurately.

The latter, Deep Learning-based works [13, 15] assume the hand area is the object closest to the camera, and extract the hand by only cropping the pixels of the smallest region with the smallest depth value (they use depth sensor), However, in guitar playing, the guitarist holds the guitar neck during playing, which cause the depth value of the hand and guitar neck are nearly same, and it is impossible to extract the hand area in the first step of the approach. Besides, they [13, 15] require a huge training database (over

50,000 images) due to viewpoint quantization and cannot be universally used as it is overfitted to their own training dataset.

Recently, T. Simon et al. [19] estimate the hand joint based on the body joint detection result. More specifically, based on the result of body joints' detection, it first extracts a bounding box around the detected wrist position, where it requires a very accurate detection result of the wrist joint. Once the wrist is not detected accurately, the hand pose cannot be detected at all. In the guitar playing case, the wrist is occluded by the guitar neck during guitar player. Based on the test by the author of this thesis (fully implemented by using original code), there are only less than 10% of total frames that can accurately detect the hand joints in the guitar playing scene. The result is shown in Fig 2.1. Additionally, it requires 32 cameras fixed on the wall to train the CNN network, which is also very hard for users to fine-tune the system considering the setting-up or the camera calibration.

2.3.2 Hand Pose Tracking

For the hand tracking, Qian et al. [20] uses a 3D hand model and Particle Swarm Optimization to estimate the parameters of 16 DOF of the hand. Although it achieves a fairly high accuracy, it can detect and track the hand from depth image sequence only when users hold nothing and wave hand in the air, and once the user holds something or put his hand on the desk, the tracking is not accurate anymore, because the depth value of the object held in his hand is as same as the hand; the hand cannot be segmented accurately anymore.

Sharp et al [21] propose an articulated hand tracker that combines fast learned reinitialization with model fitting based on stochastic optimization of a loss function. Their evaluation demonstrates not only highly accurate hand pose estimates, but also dramatic improvements over the state of the art in robustness, recovering quickly from failure and tracking reliably over extended sequences, and flexibility, working for arbitrary global hand poses, at extreme distances from the camera, and for both static and moving cameras [21]. However, it requires users to hold nothing and to show the bare hands in the air, because all of them locate the hand based on depth information, which means that their methods cannot be applied to the hand tracking for guitar plays, because the users hold the guitars.

For the fingertip tracking, Togootogtokh, E. et al. [22] propose a 3D framework for accurately tracking fingertips, but their work needs users wave their hand in the air without holding anything while tracking; She et al. [23] propose a feature point-based method which track fingertips and palm in real-time, but their method cannot solve the self-occlusion problem that frequently happens in guitar playing scenes. Besides, it also requires the fingertips to be visible from the camera constantly

2.4 Related Works of Fingering Decision and Recognition

For polyphonic instruments such as piano and guitar, the problem of fingering decision and recognition can be formally described as the transformation of a time ordered sequence of audio or video samples into a set of tuples describing start, end, fundamental frequency or pitch and optionally amplitude of the notes that are played [40]. As mentioned in Section 1.1, the fingering decision and recognition is a fundamental and very popular problem in academic research [52]. For example, in guitar playing, it has to determine that the <*string, fret, finger*> combinations for each note in the audio or video samples in time sequence.

S.Sugano [123] proposes a robot than can play piano automatically without human instruction. The fingering decision module in the robot proposes an evaluation function to mathematically calculate the optimum route for the movement of the hand of the robot during piano playing. More specifically, by developing an evaluation function consisting of two parts: (1) the ratio of the role sharing between finger and wrist of the piano player; (2) implementing a cooperate action between the finger and the wrist by calculating the speed of the wrist of the player, it helps the robot automatically decide the route of the finger and the wrist.

Radisavljevic et al. [124] use dynamic programming (DP) to model the decision process of a guitarist by choosing the optimal fingering sequence. To estimate the DP cost functions based on examples of guitar fingering transcriptions (tablatures) they develop an original method named "path difference learning" employing a gradient descent search on the coefficients of the cost function. Features of the fingering alternatives that capture the essence of the mechanical difficulty and musical quality are used to reject impractical fingerings and thus reduce the DP search complexity [124].

Y. Tomboyish et al. [125] proposes a Hidden Markov Model (HMM)-based algorithm for automatic decision of piano fingering. The method represents the positions and forms of hands and fingers as HMM states and model the resulted sequence of performed notes as emissions associated with HMM transitions. Optimal fingering decision is thus formulated as Viterbi search to find the most likely sequence of state transitions. The proposed algorithm models the required efforts in pressing a key with a finger followed by another key with another finger, and in two-dimensional positioning of fingers on the piano keyboard with diatonic and chromatic keys [125].

The fingering decision algorithm [123-125] provide the algorithm that can automatically calculate the movement path of the finger of the player. As it is not directly related to the thesis, the drawbacks are skipped.

2.5 Related Works of Assessing System for Human Action

Recent trend and advance in machine learning, especially the deep learning-based algorithms provide researchers accurate and robust result for human action recognition. On the other hand, the research of human action assessment is still challenging in computer vision community [79]. However, compared with human action recognition, human action assessments are also very important in Human-Computer Interaction (HCI), health care and so on. The reasons are as follows: (1) Assessing action requires human professionals for a specific domain to be trained over years to develop complex skills; therefore, he or she can assess the action with the experiences and professional skills; while almost everyone can recognize the action such as sit down or walk. (2) there are many datasets for human action recognition with fully labelled annotation: 68 public datasets: 28 for heterogeneous and 40 for specific human actions with annotation [88], while there are very few datasets for human action assessment in only few specific human action, such as HTSK (Head-Toes-Knees-Shoulders) action [19] for health care area.

For the problem of human action assessment, there are a few promising research projects [79, 89-91]. A. S. Gordon [89] proposes a computer system that evaluates a specific gymnast performing: the vault. Specifically, sections of the video are digitized into 240 x 180-pixel images at a rate of 12 frames per second. Then, the images were then analyzed using a motion-tracking algorithm [92] which effectively computes the center of a moving object in a series of continuous frames. The resulting data represents the location of the gymnast in each frame expressed in image coordinates. Based on the trajectory of the calculated coordinates, it assesses how well the player performed the vault action. Obviously, human-designed feature [89, 90, 91], such as trajectory of the player of gymnast can be applied only to one specific action, and cannot be generalized into other actions; besides, it requires an accurate mathematical evaluation function designed by human being, which is considered very hard and unpractical to be applied in computer vision-based method: as it requires to calculate each important feature only based on video or image for each important aspect in evaluation functions.

H. Pirsiavash et al. [79] propose a learning-based framework that takes steps towards assessing how well people perform actions in videos. Their approach works by training a regression model from spatiotemporal pose features to scores obtained from expert judges. Moreover, it can provide interpretable feedback on how people can improve their action. With the inaccurate pose estimation of Olympic players, it can only assess the 2D image sequences by obtaining 2D joints of athletes in an inaccurate manner. However, inspired by the feature they used [79, 93]: the author of this thesis found that some of these features, Cosine transform and Fourier Transform with some minor adjustments, may be useful in the guitar playing assessment research.

Recently, with the boom of the deep learning-based computer vision method, the research of human action assessment [19, 87] are also applied in medical care areas. S. Gattupalli et al [19] automates capturing and motion analysis of users performing the HTKS (Head-Toes-Knees-Shoulders) game and provides detailed evaluations using state-of-the-art computer vision and deep learning-based techniques for activity recognition and evaluation. The system is supported by an intuitive and specifically designed user interface that can help human experts and doctors to cross-validate and/or refine their diagnosis [19]. However, due to the requirement of deep learning network, S. Gattupalli et al [19] create a huge data-set which consists of 15 subjects performing 4 different variations of the HTKS task and contains in total more than 60,000 RGB frames with annotated labels.

Another deep learning-based hand motion assessment research [87] present an automated method for quantifying the severity of motion impairment in patients with ataxia, using only video recordings. The object of the research is also a medical game named "finger-to-nose" test, a common movement task used as part of the assessment of ataxia progression during the course of routine clinical checkups [87]. In their work, neural network-based pose estimation and optical flow techniques to track the motion of the patient's hand in a video recording. It also manually extracts features that describe qualities of the motion such as speed and variation in performance. Using labels provided by an expert clinician, it trains a supervised learning model that predicts severity according to the Brief Ataxia Rating Scale (BARS). The performance of the system is comparable to that of a group of ataxia specialists in terms of mean error and correlation, and the system's predictions were consistently within the range of inter-rater variability. This work demonstrates the feasibility of using computer vision and machine learning to produce consistent and clinically useful measures of motor impairment [87]. However, tracking the hand of a patient by using optical flow is an extremely unpractical experiment in the guitar playing. As mentioned in Section 2.1, Optical Flow is not robust enough under the lighting illumination condition, such as indoor scene, because Optical Flow calculates the intensity of each pixel while the intensity (RGB value) varies significantly under the fluorescent light. Besides, their method also needs huge amount of data with fully labeled annotation to train the network, while in guitar playing, there is no public dataset for the author to use.

2.6 Summary

Chapter 1 details the related works of the algorithms that combine computer science and music. Moreover, the related works of fingering recognition and fingering assessing listed in Table 1.1 are detailed and discussed over their advantages and disadvantages. Based on Table 1.1, it turns out that there is no work on fingering assessing method for guitarists.

Chapter 2 details the computer vision and machine learning based related works including: (1) object tracking algorithm (for guitar neck tracking), (2) hand region segmentation (for guitarist hand region segmentation), (3) multiple targets tracking algorithm and hand pose estimation (for track or estimate fingering of guitarist), (4) human action assessing algorithm (for fingering assessing).

Section 2.1 to Section 2.4 are summarized as follows:

1. For Guitar Neck Tracking Module

   The most important requirement for guitar neck tracking is the accuracy, because it is the first step of the guitar fingering assessing system, and the subsegment processes of the system (the finger pose tracking module and the fingering assessing module highly rely on the result of the guitar neck tracking). The SIFT and SURF based method may be one of the best algorithm as it is invariant to the rotation, scaling, which are most frequently happened during guitar playing (the player swings the guitar while playing). However, SIFT or SURF based method is not accurate when occlusion happens (the guitar neck is occluded by the hand of the guitarist), the kernel-based method may filter the occluded feature points, and recover from the tracking failure.

2. For Finger Pose Tracking or Estimation Module

   a. For Hand Segmentation

   Traditional methods of segmentation always segment human skins in a not robust manner, because it uses features such as color, shape and etc. are not robust considering the varying illumination and complex hand shape. Convolutional Neural Network outputs the possibility that each pixel corresponds to the hand region pixel or not; therefore, the training and testing efficiency are not good enough. Recently, Fully Convolutional Neural Network (FCN) [96] achieves 98% of the accuracy for semantic segmentation in automatic driving system. Therefore, FCN based segmentation is worth trying to generate accurate segmentation result despite the different illumination condition and skin color.

   b. For Finger Pose Tracking

   After segmenting hand area, the finger pose of the guitarist needs to be tracked. As mentioned in Section 3.2, the jointed finger and self-occlusion problems during guitar playing are the two difficulties which this thesis needs to deal with. Particle filter [104] solves the occlusion and partial incompliance by adapting a non-linear, non-gaussian model to overcome the issues. However, since four fingertips share the same appearance including skin color and semi-circle shape, traditional particle filters are hard to be applied as wrong particle transition happens. Therefore, the solution of the wrong particle transition maybe fixed by a proper ROI (Region Of Interest) generation and association, more specifically, first ROI of fingertips are generated for each frame, second the ROIs are associated between consecutive frames to link the same fingertips in the consecutive frames, third particles are distributed only within the associated ROIs to track the fingertips position. On the other hand, to overcome the problems of the CNN based methods [13, 15, 19] mentioned in Section 2.3, a simple neural network that can predict 3D coordination of the hand's joint is worth trying. More specifically, the traditional CNN based methods always require huge dataset to generate accurate result, in this thesis a CNN based network that can predict accurate result without training on a huge number of dataset is much more preferable as it would not require the guitarist to train the network for 100 hours even with the most advanced GPUs.

3. For Guitar Fingering Assessing

Instead of manually designing an evaluation function, for example in related work [89], the training (data-driven) based fingering assessing is preferable as (1) each music piece may have different criteria to assess how well the guitarist played; therefore instead of designing different evaluation functions for different music pieces, a training based guitar fingering assessing is much more efficient and proper way; (2) as there is no evaluation function, the training based method can be easily generalized to other hand movement assessing, even human action assessing, because only the training video and labelled evaluation of scoring result is needed in training based method. Related works [79, 93] suggest DCT (Discrete Cosine Transform) and SVR (Support Vector Regression) based regression model is worth trying, however how to create 3D DCT (as finger pose is in 3D) is the problem and needs to be conducted in this thesis.

Chapter 3 Guitar and Definitions

3.1 Outline

This chapter explains about what kind of musical instrument the guitar is. Then, how to assess the guitar play is illustrated. Finally, the coordinate systems and codes, which are used for the subsequent chapters that describe the proposed modules, are defined.

3.2 Guitar and Assessing Guitar Play

The guitar is a fretted musical instrument that usually has six strings [120]. The sound is projected either acoustically, using a hollow wooden or plastic and wood box (for an acoustic guitar), or through electrical amplifier and a speaker (for an electric guitar). It is typically played by strumming or plucking the strings with the fingers, thumb or fingernails of the right hand or with a pick while fretting (or pressing against the frets) the strings with the fingers of the left hand. The guitar is a type of chordophone, traditionally constructed from wood and strung with either gut, nylon or steel strings and distinguished from other chordophones by its construction and tuning. The modern guitar was preceded by the gittern, the vihuela, the four-course Renaissance guitar, and the five-course baroque guitar, all of which contributed to the development of the modern six-string instrument [121].

Chapter 1.1 gives the definition of fingering. However, mapping each note into a three-dimensional space <*string, fret, finger*> is not the only criterion to assess guitar play. In other words, correctly mapping each note into a three-dimensional space (**Finger Placement**) does not equal with generating beautiful and elegant music. As guitar playing is an extremely complex process, it is very hard to summarize a unified but specific rule to assess every kind of guitar playing.

Christopher Perez [122], who is elected as a quarterfinalist for the 2017 GRAMMY Foundation Music Teacher of the year award, addresses the proper performance assessment of guitar fingering is *"a valuable source of formative and summative grading system*" [112]. More specifically, he introduces some quantitative criteria to assess ***C Major*** **Scale** of guitar playing: the assessment criteria are a scoring-based system that ranges from 0 to 75 (full mark) [122]. The specific criteria are shown as follow:

For C Major Scale of the first fret playing:

(1) all the notes should be played in right pitch;

(2) the tempo of the scale should be played correctly;

(3) the audio appearance should be smooth without breaking, muting and buzzing;

(4) player should hold guitar in a good manner: 45 degrees and being lowered.

(5) player should use right leg to support guitar body, and sit straight and all the body joint should relax during playing;

(6) playing should use alternative finger (index finger and middle finger) of right hand to play the scale;

(7) all the fingers of the left hand should be parallel to the fret of guitar fret during playing;

(8) all the fingertips of the left hand should be perpendicular to the fret of the guitar neck during pressing note.

(9) the resting finger should be relaxed during the playing. The thumb of the left hand should be placed behind the guitar neck.

(10) the right fingering should use a stroke in a good manner during the playing with a proper strength.

From *C Scale* assessment, evaluating a C Scale performance of guitarist should not only assess the **finger placement** (Criterion 1, 9, 10) but also body pose (Criterion 4,5), guitar pose (Criterion 4). Score-based sound performance (Criterion 1, 2, 3) and etc. Among all the criteria, Criterion 6-10 indicate that both **finger placement** and **finger movement** (transitions between each two-finger placement) need to be assessed in guitar fingering assessing, for instance, in Criterion 8, all the fingertips of the left hand should be perpendicular to the fret of the guitar neck during pressing note. In another word, Criterion 8 evaluates the C Scale playing from a whole finger movement, a transition of finger movement instead of a specific finger placement at a specific timing

This thesis conducts an automatic guitar fingering assessing research, referring to the Christopher Perez's theory for the left-hand fingering. Specifically,

- **computer vision based**: by collecting the guitar playing videos, the spatio-temporal information of the guitar playing such as 3D position and the guitarist's left-hand pose are extracted.
- **scoring-based assessment** is the principle of the guitar left hand fingering assessing system: the system outputs a score based on the extracted spatio-temporal video information to help the user of the system to know how well he or she performed.

- **training-driven without human designed evaluation function:** an effective image feature that can represent the collected spatio-temporal information is utilized for a training based regression model. Therefore, the videos with only correspondent labelled score are trained to predict score based on the extracted spatio-temporal video information without using human designed evaluation function

3.3 Coordinate Systems and Code Table

In this thesis, as shown in Fig 3.1, four coordination systems are defined: (1) world coordination ($X_wY_wZ_w$ Coordination in Fig 3.1 and Table 3.1): because distinct systems cannot interact with each other, the world coordination is used as a universal coordination system in order to corporate with each different system; (2) camera coordination ($X_cY_cZ_c$ Coordination in Fig 3.1 and Table 3.1): the camera coordination is defined as a three dimensional space where the origin is the position of the camera; (3) guitar coordination ($X_gY_gZ_g$ Coordination in Fig 3.1 and Table 3.1): the guitar coordination is defined as a three dimensional space where the origin is the head position of the guitar, and it is used for (a) estimating the tracking error of the guitar neck; (b) the limitation of the rotation $(Yaw, Pitch, Row)$ in Fig 3.1 and Table 3.1 and translation test in Guitar Neck Tracking Module in Chapter 4; (4) user coordination ($X_uY_uZ_u$ Coordination in Fig 3.1 and Table 3.1): the user coordination is defined as a three dimensional space where the origin is the position of the head of the user; and it is used for estimating the tracking error of the 3D hand of the guitarist in Hand Pose Estimation Module in Chapter 6 and calculating features in Fingering Assessing Module in Chapter 7. (5) image coordination (XY Coordination in Fig 3.1 and Table 3.1): the image coordination is defined as a two-dimensional space where the origin is the left-bottom of the projected image of the camera; and it is used for calculating the mean error of 2D fingertip tracking in Chapter 5 and other calculation of image processing methods all over this thesis.

Besides, Table 3.1 also defines other codes which are used in this thesis.

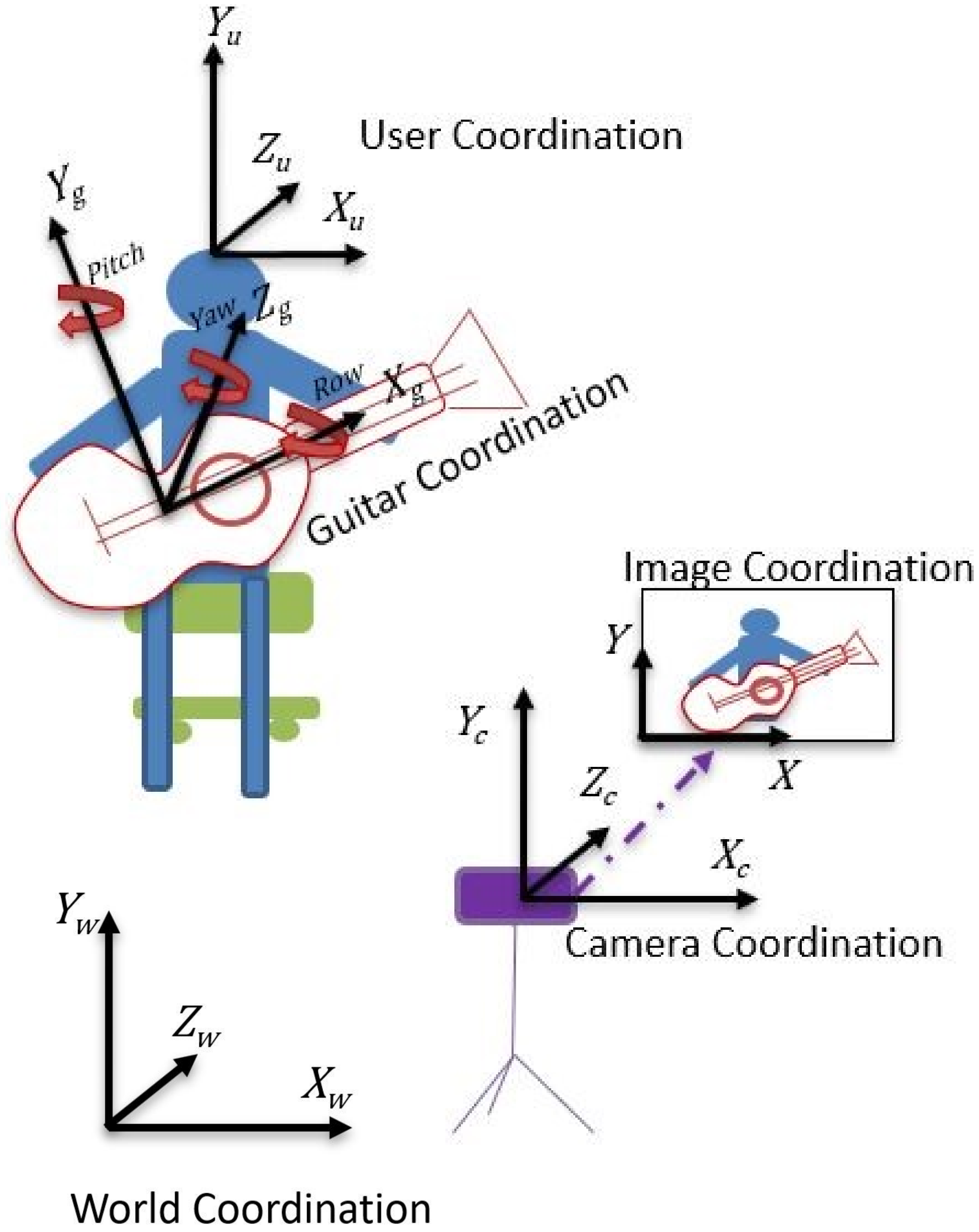


Fig 3.1 Coordination Systems of This Thesis

Table 3.1 Code Table of This Thesis

Code	**Explanation**	**Code**	**Explanation**
$X_w Y_w Z_{\mathrm{w}}$ Coordination	World Coordination	G	global accuracy
$X_c Y_c Z_c$ Coordination	Camera Coordination	R	Scoring Fucntion
$X_g Y_g Z_g$ Coordination	Guitar Coordination	$(x_{i.w}, y_{i,w}, \omega)^T$	Homographic Transform of the w-th Corner
$X_u Y_u Z_u$ Coordination	User Coordination	$(x_{1,w}, y_{1,w}, 1)^T$	Homographic Transform of the w-th Corner in Frame 1
$d(Yaw, Pitch, Row)$	Rotation Coordination	$Dis(aa, bb)$	Distance of aa and bb
XYCoordination	Image Coordination	G	global accuracy
(x, y)	Pixel Index at (x, y) in XY $Coordination$	$I(x, y)$	Pixel Intensity in XY $Coordination$
$E(u, v)$	Intensity Variation at (x, y)	C	class average accuracy
i	Frame Index	$mIoU$	mean intersection over union
$H \cdot X$	Inner Production	$P^{(j)}(t)$	$3J * T$ matrix
$\widehat{L_\iota}$	Line Set	$F(j\omega)$	Fourier Transform
H	RANSAC Homography	$F[n]$	DFT
$Fin_{(x,y)}$	Fingertip at (x, y)	$F[k, l]$	2D DFT

$T_{sum(x,y)}$	Template Matching Result at (x, y)	$G'_{t,i}$	i-th Grouping Result at Frame t
$R_{sum(x,y)}$	Reversed Hough Transform Result at (x, y)	$G''_{t,i}$	i-th Grouping Result at Frame t of Second Grouping
Fin_{Normal}	Fingertip Normalization	$Q^{(j)}$	Feature Matrix
T_t	T -Test value	D	Score Difference
s_p	Pooled Standard Deviation	ME	Mean Error
$s^2_{X_1}$ $s^2_{X_2}$	Unbiased Estimators of The Variances	V	Variance
$x\ (p)$	Original Input Pixel	M	Homography Matrix

Chapter 4 Tracking Guitar Neck

4.1 Introduction

In Chapter 4, the guitar neck tracking module of this system is discussed. As mentioned in Section 1.3 and Section 1.4, the guitar neck tracking module is the first step in the system, and all other modules including finger pose tracking module and fingering assessing module are implemented based on the result of guitar neck tracking; therefore, the guitar neck needs to be tracked accurately; otherwise, it would cause serious problems of the other modules in the system.

Figure 4.1 shows an example of the video inputted to the system. The whole guitar neck area must be captured in the video in order to analyze the fingering in the subsegment modules. The distance between each two adjacent strings is very close. Once inaccurate tracking happens, it would cause problems of the subsegment fingering assessing module. For example in Fig. 4.2, if the guitar neck is correctly tracked, the centroid of the guitar neck at each frame is projected to the center of each frame of a new image sequence, and the neck is placed horizontally (the top image of Fig. 4.2) to facilitate analyzing fingering; if inaccurate tracking happens, which means the guitar neck cannot be accurately projected (the right-bottom image of Fig. 4.2), it is very hard for the fingering assessing module to evaluate the fingering of the guitarist, because it must assess the fingering in a wrong manner as each string cannot be projected to its proper position even with several *milimeter* tracking error. For instance, the guitarist pressed the third string in real performance, while the system may detect that the guitarist presses the fourth string.

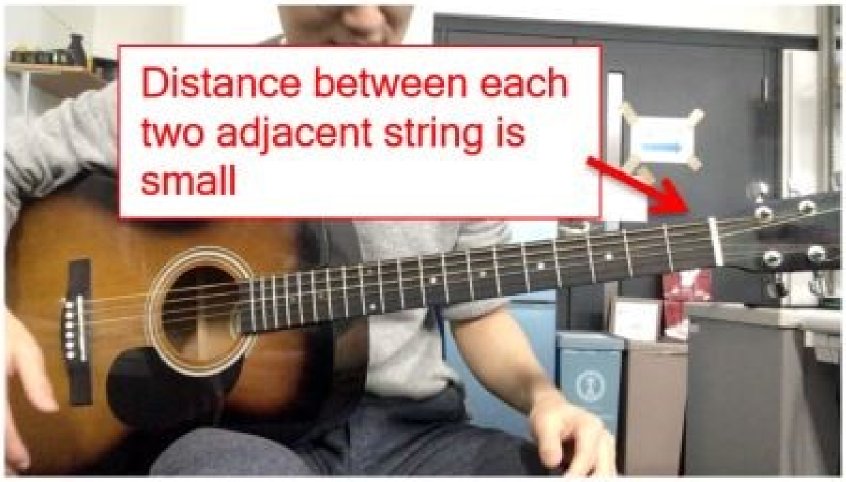


Fig 4.1 An Example of Input Video of the System

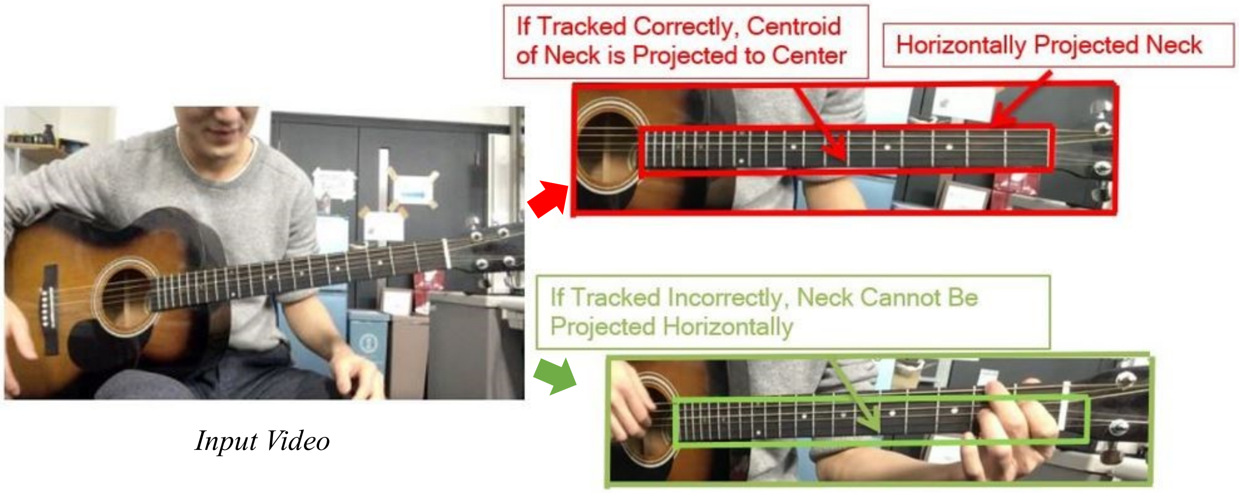
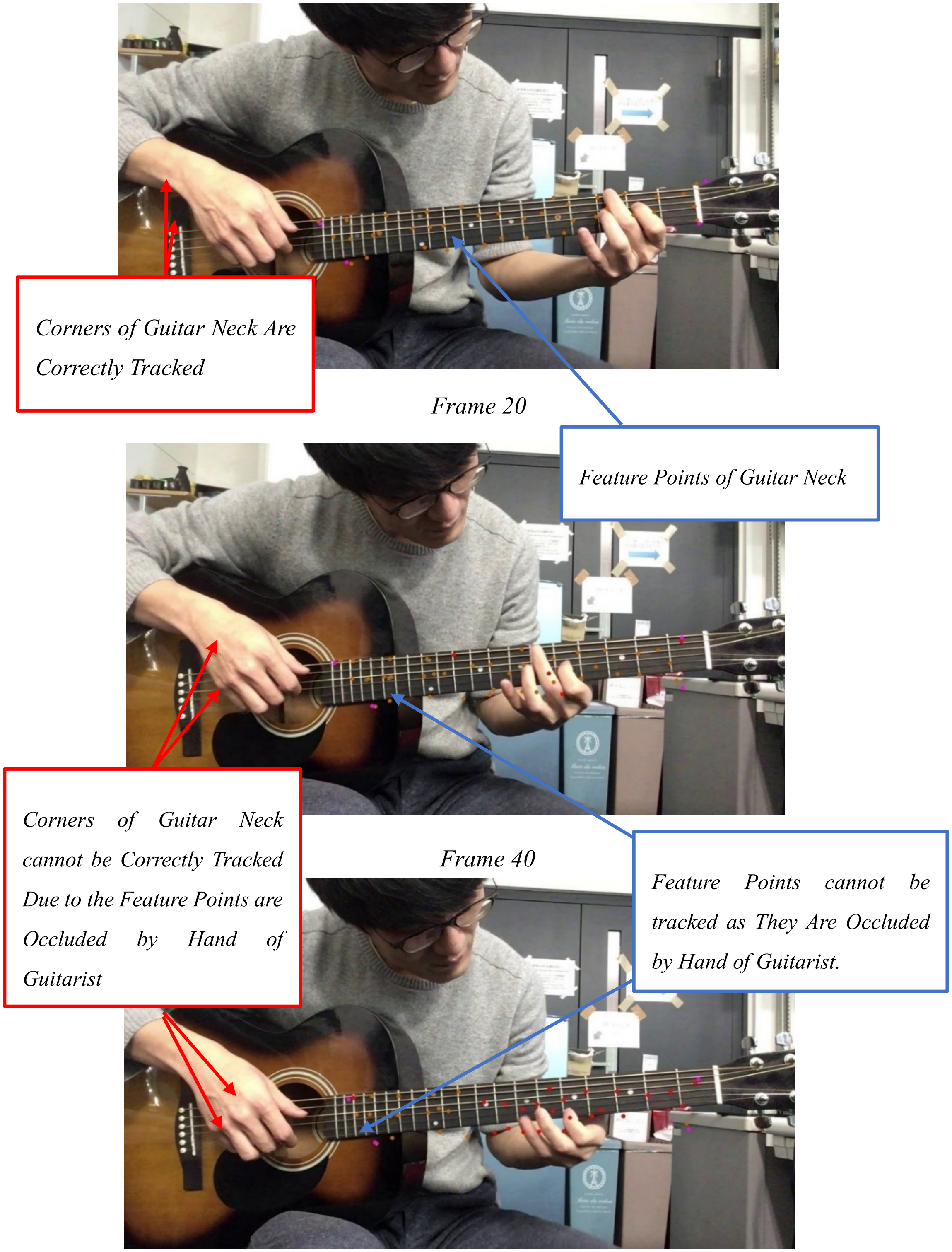

If Tracked Correctly, Centroid of Neck is Projected to Center
Horizontally Projected Neck
If Tracked Incorrectly, Neck Cannot Be Projected Horizontally
Input Video

Fig 4.2 Example of Output of The Guitar Neck Tracking Module: if the neck is correctly tracked, it is projected to a new image sequence that the guitar neck is always at the center (Right-Top image); when the inaccurate tracking happens, the guitar neck cannot be projected to the center even with few millimeters of tracking error.

Another challenge is that, as the guitarist may move the fingers everywhere on the guitar neck fretboard at a high speed; therefore, features of the guitar neck used for tracking maybe partially occluded by the hand of the guitarist as shown in Fig. 4.3, which makes the tracking extremely hard to process considering the high accuracy is need as mentioned before.

In this chapter, an accurate and robust guitar neck tracking module is proposed to solve the problems mentioned before. Specifically, (1) the guitar neck is automatically detected at the first frame of the input video by detecting the rectangles formed by the horizontal strings and vertical frets; (2) SIFT feature points are to be detected at every frame as it is invariant to rotation, illumination and scale changes in images a KD-tree searching based algorithm is utilized to match the SIFT features between the first frame and any other frame of the input videos; (3) a modified version of RANSAC (Random Sample Consensus) is proposed to overcome the above-mentioned occlusion issue. As mentioned earlier, feature points within the guitar neck area cannot be accurately tracked or matched, because it is overlapped and occluded by the guitarist's fingers. The proposed modified RANSAC-based filtering algorithm filters out and eliminates the mismatched feature points, and then calculates the homography between the correctly matched feature points at the first frame and any other frame to track the guitar neck. Besides, since the homography is calculated between the first frame and any other frame, this method does not need to concern about



Corners of Guitar Neck Are Correctly Tracked

Frame 20

Feature Points of Guitar Neck

Corners of Guitar Neck cannot be Correctly Tracked Due to the Feature Points are Occluded by Hand of Guitarist

Frame 40

Feature Points cannot be tracked as They Are Occluded by Hand of Guitarist.

Frame 60

Fig 4.3 Example of Occlusion (Feature points are occluded by the hand of guitarist) in Consecutive Frames

the tracking failure problem. (4) to suppress the effect of the guitar neck motion, the tracked guitar neck area at each frame is projected to the center of a new image sequence based on the calculated homography frame by frame. Owing to this projection, no matter how the guitar player shakes or swings the guitar neck while playing, the neck area at each frame is always horizontally projected to the centroid of a new image sequence to facilitate analyzing the fingering, where this analysis is the fingering assessing module described in Chapter 7.

As depicted in Fig. 4.4, after the video of guitar playing is input, first the guitar neck area is automatically detected at the first frame of the input video, and at the first frame, three types of feature points (SIFT, SURF, Shi-Tomasi) are detected respectively within that fretboard area. Then, from the second frame, SIFT/SURF features are detected at each frame and matched with the SIFT/SURF detected at each frame by using a KD-Tree based searching method to accelerate matching efficiency. On the other hand, Shi-Tomasi features are tracked frame by frame by using optical flows (since Shi-Tomasi feature points cannot be matched between each two frames). Furthermore, in order to solve the problems of occlusion due to the mismatched SIFT/SURF feature point pairs and the mis-tracked Shi-Tomasi between the first frame and the current frame, a modified RANSAC mechanism is proposed since the traditional RANSAC cannot handle the mis-match correctly. In addition, the perspective transform matrix based on correctly matched SIFT/SURF/Shi-Tomasi pairs is obtained to project the fretboard area to a new image sequence where the guitar neck is always horizontally projected to the centroid of a new image sequence to output the tracking result.

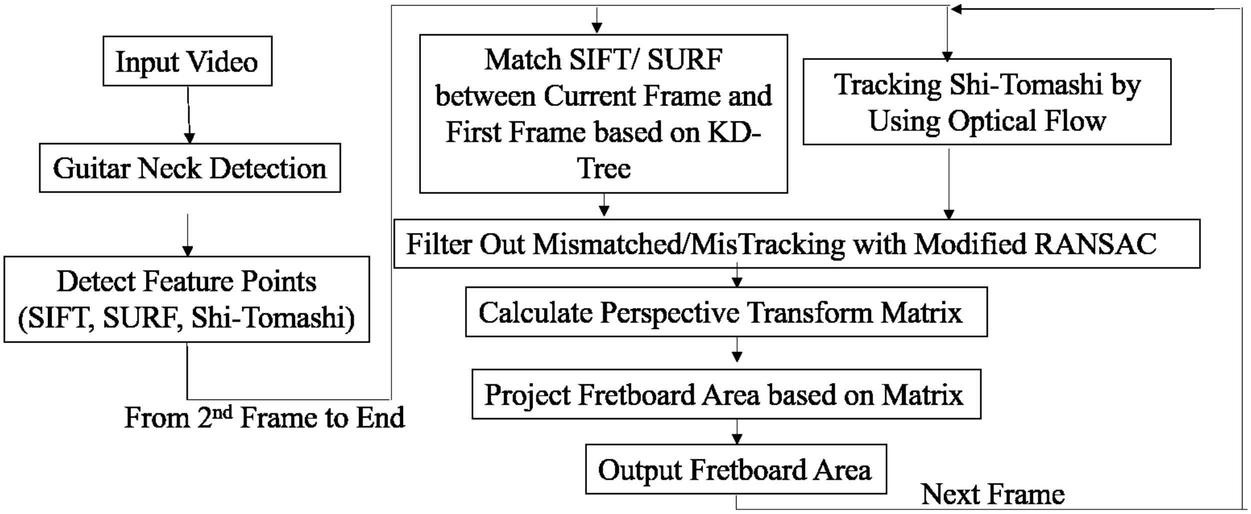
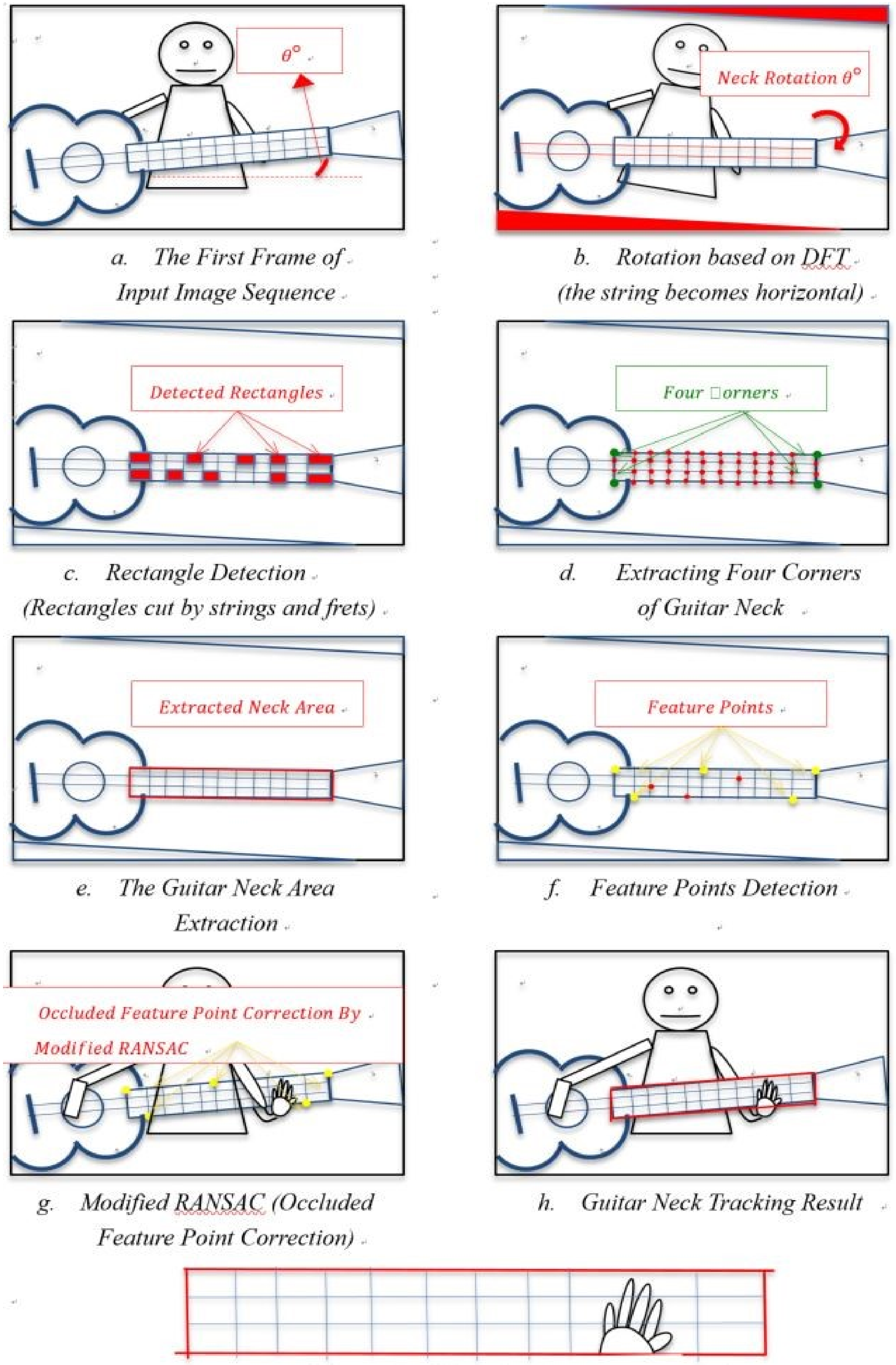

Input Video
Guitar Neck Detection
Detect Feature Points (SIFT, SURF, Shi-Tomashi)
From 2nd Frame to End
Match SIFT/ SURF between Current Frame and First Frame based on KD-Tree
Tracking Shi-Tomashi by Using Optical Flow
Filter Out Mismatched/MisTracking with Modified RANSAC
Calculate Perspective Transform Matrix
Project Fretboard Area based on Matrix
Output Fretboard Area
Next Frame

Fig 4.4 Outline of Guitar Neck Tracking



a. The First Frame of Input Image Sequence

b. Rotation based on DFT (the string becomes horizontal)

c. Rectangle Detection (Rectangles cut by strings and frets)

d. Extracting Four Corners of Guitar Neck

e. The Guitar Neck Area Extraction

f. Feature Points Detection

g. Modified RANSAC (Occluded Feature Point Correction)

h. Guitar Neck Tracking Result

i. Projection of Guitar Neck based on Perspective Transform

Fig 4.5 Concept of Guitar Neck Detection and Tracking

4.2 Guitar Neck Detection

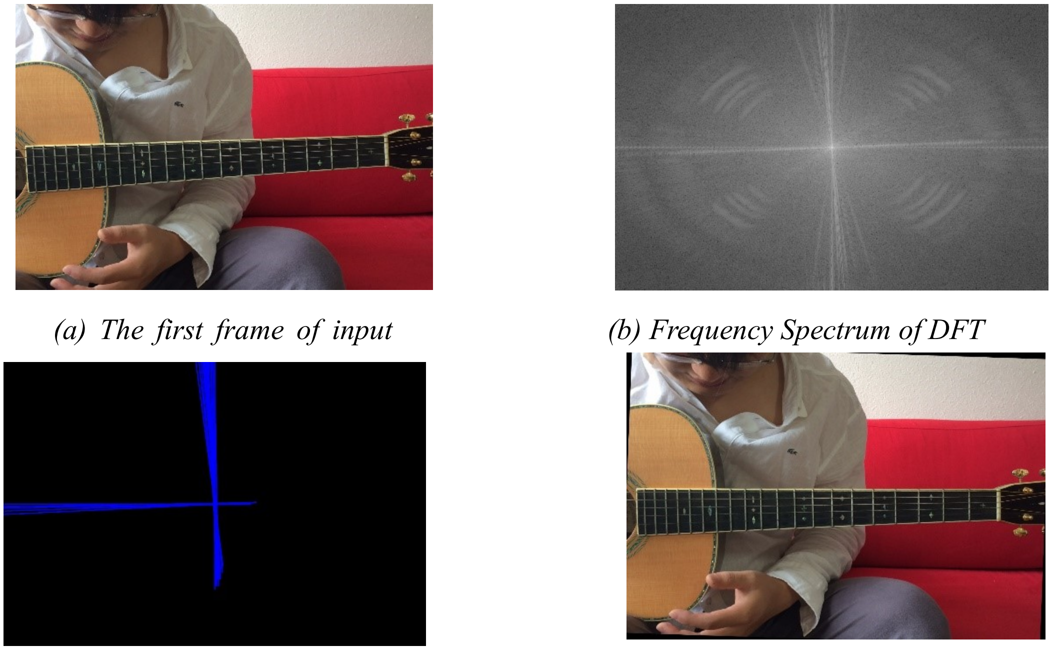
Overall, to automatically detect the guitar neck area at the first frame of the input video, as the concept of the guitar neck detection shown in Fig. 4.5. First one of the most distinctive features of guitar neck rectangles formed by guitar strings and frets (Fig. 4.5.c) need to be detected. Then, by filtering out the largest and the smallest coordinates of the rectangles detected before, the whole area of the guitar neck is detected.

4.2.1 Rotation Based on DFT

The Fourier Transform decomposes an image into its sinus and cosines components. In other words, it transforms an image from its spatial domain to its frequency domain. The idea is that any function may be approximated exactly with the sum of infinite sinus and cosines functions. Mathematically a two-dimensional image's Fourier transform is computed by [55, 84]:

$$F[k,l] = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(x,y) e^{-i2\pi\left(\frac{ki}{N}+\frac{lj}{N}\right)} \tag{4.1}$$

In Eq. 4.1, $I(x,y)$ is the image value at (x,y) in its spatial domain of the XY coordination and $F[k,l]$ is the frequency component at $[k,l]$ in the frequency domain. The result of the transformation $F[k,l]$ is complex numbers. Displaying this is possible either via a real image and a complex image or via



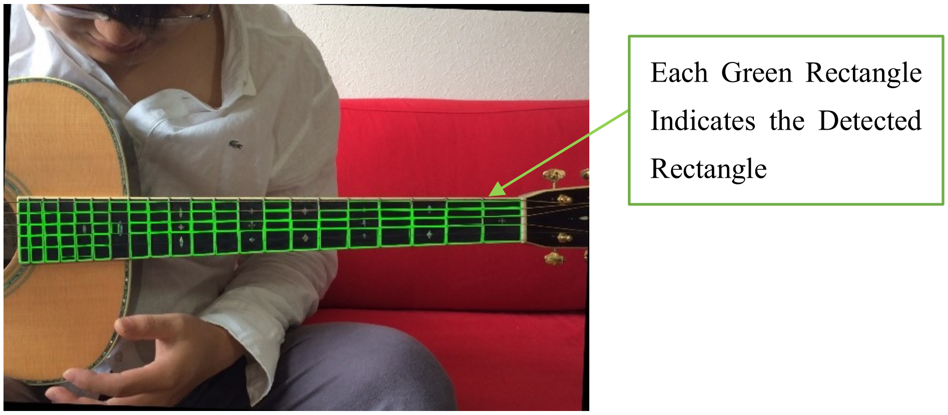
(a) The first frame of input

(b) Frequency Spectrum of DFT

(c) Hough Line Detection on Spectrum

(d) The Result of Guitar Neck Rotation

Fig 4.6 Rotation based on DFT



Each Green Rectangle Indicates the Detected Rectangle

Fig 4.7 Rectangles Detection (Green rectangles are the result)

a magnitude and a phase image [55, 84].

In the Guitar Neck Tracking module, after inputting the first frame of a guitar playing video file (Fig.4.6 (a)), DFT (Discrete Fourier Transform) [54] is performed to get the frequency spectrum (Fig.4.6 (b)). Then, Hough Transform is used to detect the largest cluster of lines on the transformed Fourier Domain. Since the angle between strings of the guitar (Fig.4.6 (c)) and the horizontal axis can be computed as the angle between the largest cluster of lines and the vertical axis of Fourier domain, by detecting the angle between that cluster and the vertical axis, the input frame is rotated so that the top-most string of the guitar becomes nearly horizontal (Fig.4.6 (d)) [84].

4.2.2 Detection of Rectangle

As described in Section 4.1, one of the most distinct features of the guitar neck is a set of rectangles formed by the horizontal strings and vertical frets. First, Canny edge detection algorithm is applied to the rotated first frame of the input video sequence. Then, after dilation and binarization at the first frame, closed contours are found. Only the contours satisfying the following conditions are extracted as shown in Fig. 4.7: (1) approximate quadrangle of each contour is a rectangle; (2) the absolute cosine value of the angle between the two lines that meet at a corner is less than 0.1; (4) the area of every quadrangle is between 50 pixels and 1000 pixels. Finally, the above-mentioned two steps are applied in the three channels (RGB) of the first frame to find all possible contours that could be rectangles. In Fig.4.7, green rectangle indicates the result of rectangle detection [84].

4.2.3 Guitar Corner Detection

The rectangles detected in Section 4.2.2 are saved in the computer's memory as four sequential vertexes. Therefore, in order to detect the whole area of the neck, all the vertexes are used to find the four

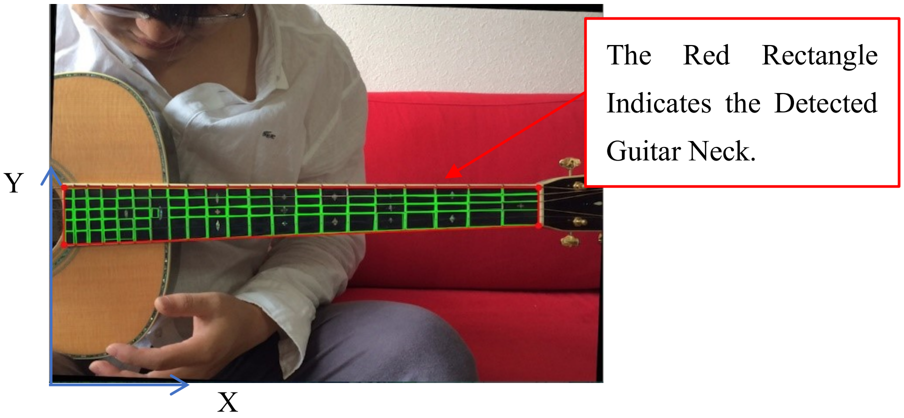


Figure 4.8 Guitar Neck Detection (The Red Rectangle is Detected Guitar Neck)

corners of the guitar neck. Since the image is rotated in Section 4.2.1 (Fig.4.5.b), the upper edge of the neck is horizontal and the left and the right edges are vertical, the four vertexes of the neck could be easily found by following the follow rules (Fig.4.5.d): (1) the smallest x and the smallest y coordinate among all the vertexes correspond to the upper-left corner of the neck; (2) the smallest x and the largest y coordinate correspond to the lower-left corner; (3) the largest x and the smallest y coordinate correspond to the upper-right corner; (4) among all the vertexes whose x have less-than-30- pixels distance to the largest x of all vertexes of rectangles, the largest y coordinate is extracted as the lower-right corner's vertical coordinate, and among all the vertexes of all rectangles, the largest x is extracted as the lower-right corner's x coordinate. Now, the four vertexes of the guitar neck are extracted (Fig.4.5.e). By connecting these four vertexes according to the correct order, the module for the guitar neck detection is finished. The result (red rectangle) of the guitar neck detection is shown in Fig.4.8 [84]. In the guitar tracking module, in order to track the guitar neck from the second frame to the end, the detection result of guitar neck is also manually corrected if the guitar neck is not detected correctly at the first frame.

4.3 SIFT Features and Shi-Tomasi Feature

4.3.1 SIFT [11] and KD-Tree [57 58]

As shown in Chapter 2, Scale Invariant Feature Transform (SIFT), which [56] is proved to be an effective feature for object or scene recognition with the highest accuracy result is used in the guitar tracking

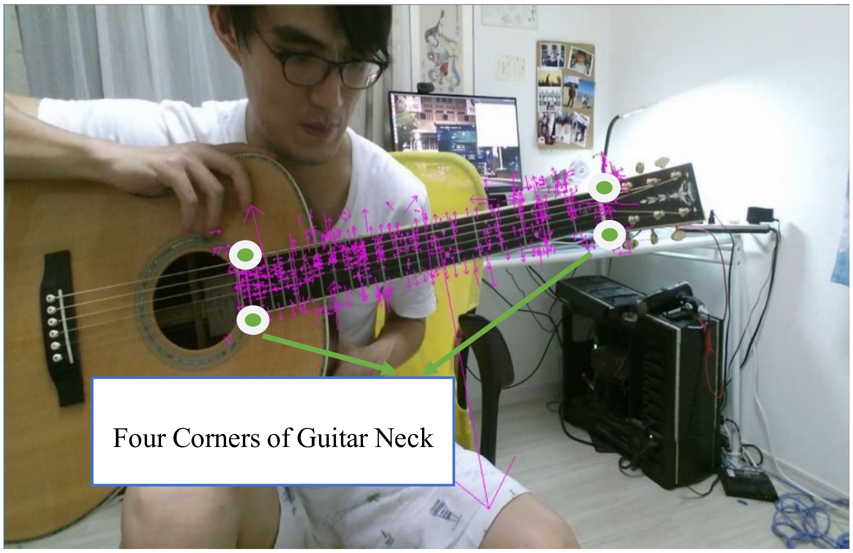


Fig 4.9 SIFT Feature (The Red Rectangle is Detected Guitar Neck)

module. After detecting the guitar neck at the first frame of an input video, SIFT feature points are detected within the guitar fretboard area, which is the output from the module explained in Section 4.2 as shown in Fig. 4.9. Some of the SIFT feature points exist outside the neck area as shown in Fig. 4.9, because the neck area is broadened by 20 pixels in order to detect the features near the border as many as possible. From the second to the end frame of the video, SIFT feature points at each frame are also detected and matched with the features detected at the first frame because each detected SIFT has two properties: the scale and the orientation. For accelerating the matching process, a KD-Tree based searching algorithm [57,58] is applied. The KD-tree is a binary tree in which every node is a k-dimensional point. Every non-leaf node generates a splitting hyperplane that divides the space into two subspaces. Points left/right to the hyperplane represent the left/right sub-tree of that node. The hyperplane direction is chosen in the following way: every node split to sub-trees is associated with one of the k-dimensions, such that the hyperplane is perpendicular to that dimension vector [58, 82].

4.3.2 Shi-Tomasi Feature and Optical Flow

(1) Shi-Tomasi Feature Point

The accuracy of SIFT-based tracking is compared with the Shi-Tomasi-based tracking in the experiment of this chapter. In this section, the property of Shi-Tomasi and the tracking algorithm for Shi-Tomasi: optical flows are discussed.

From the angle of mathematics:

Suppose a window $w(x, y)$ with the shifting movement u in x direction and v in y direction, then the variation of intensity is:

$$E(u, v) = \sum_{x,y} w(x, y)[I(x + u, y + v) - I(x, y)]^2 \tag{4.2}$$

where, $w(x, y)$ is the window position at (x, y) in image, $I(x, y)$, $I(x + u, y + v)$ is the intensity before window shifting and after window shifting.

In order to find corners, which represent large variations, the term below should be maximized:

$$\sum_{x,y}[I(x + u, y + v) - I(x, y)]^2 \approx \sum_{x,y}[I(x, y) + uI_x + vI_y - I(x, y)]^2$$

$$\approx [u, v]M \begin{bmatrix} u \\ v \end{bmatrix} \tag{4.3}$$

Where, $\sum_{x,y} = w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$

The scoring function R:

$$R = \min(\lambda_1, \lambda_2) \tag{4.4}$$

If it is larger a threshold value, it is considered as a Shi-Tomasi feature point [84]. The detection result of Shi-Tomasi feature points are shown in Fig. 4.10.

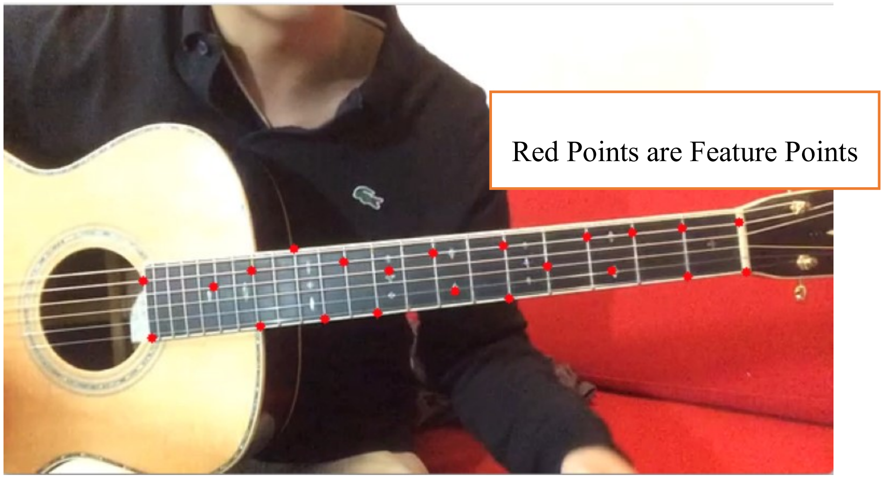
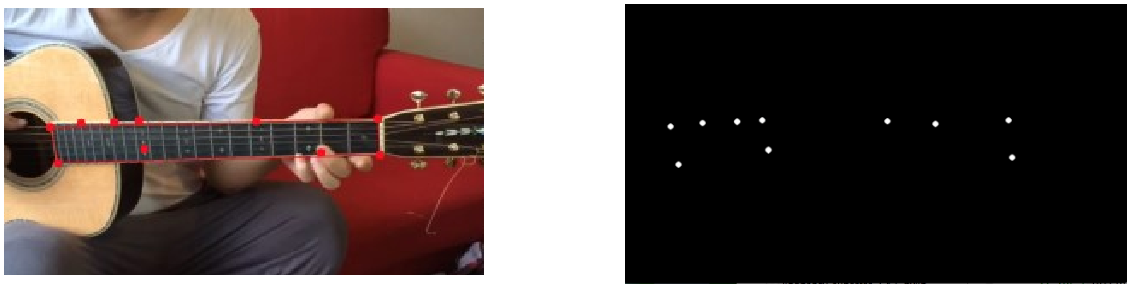


Fig 4.10 Shi-Tomasi Feature Point (The Red Rectangle is Detected Guitar Neck)



Left: Shi-Tomasi Feature Points *Right: Optical Fow of Feature*

Fig 4.11 Lucas-Kanade Based Optical Flow Tracking

(2) Optical Flows

Optical flows [59] or optic flows is a pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene [60] [61]. In computer vision, the Lucas–Kanade method [62] is a widely used differential method for optical flow estimation developed by Lucas and Kanade [63]. It assumes that the flow is essentially constant in a local neighborhood of the pixel under consideration and solves the basic optical flow equations for all the pixels in that neighborhood, by the least squares criterion. [84]

The Lucas–Kanade method has three hypotheses: (1) the points share same brightness before and after movement; (2) the movement should be as shortest as possible; (3) the pixels around the tracing point should move with tracing point.

In the guitar tracking module, Lucas-Kanade based optical flows method is used to track Shi-Tomasi Feature points, which are extracted in Section 4.2.4 between two successive frames, and the terminal point of each optical flow in the previous frame is seen as the origin of each optical flow in the next frame. According to the optical flow of those feature points, the whole area of the guitar neck is tracked frame by frame [84].

4.4 Modified RANSAC

As mentioned in the beginning of Chapter 4 and Fig. 4.3, because the partial guitar neck inevitably is occluded by the guitarist’s hand during guitar playing, the feature points cannot be correctly matched or tracked. Therefore, accurate tracking of the whole guitar neck area is very hard to perform. The strategy of this section is that the mis-matched or mis-tracked feature points need to be filtered out and corrected to the

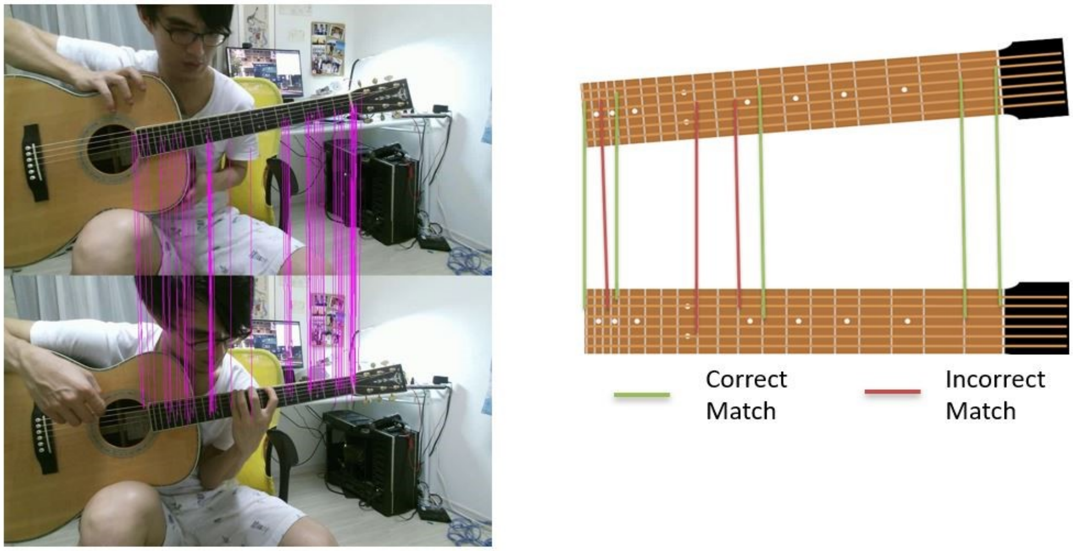


Fig 4.12 Mis-matches Present in Guitar Neck Tracking

proper position.

Traditional RANSAC (Random sample consensus) [85] is an iterative method to estimate parameters of a mathematical model from a set of observed data that contains outliers, when outliers are calculated no influence on the values of the estimates. Therefore, it also can be interpreted as an outlier detection method [86]. In the guitar tracking, the outliers are regarded as the mis-matches of feature point pairs between the first frame and subsegment frame due the hand occlusion issues, while the inliers indicate the correctly matched features point pairs.

Specifically, traditional RANSAC algorithm removes the mismatches by finding the transformation homography matrix of these feature points among all the matching result (both inliers and outliers). The mathematical RANSAC process of guitar neck tracking is shown in Eq. (4.5) and Eq. (4.6):

$$X_i = HX_1 \ , i \in (1,2,3 \dots I) \tag{4.5}$$

$$\begin{pmatrix} x_i \\ y_i \\ \omega \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}, i \in (1,2,3 \dots I) \tag{4.6}$$

where: $(x_i, y_i, \omega)^T$: homographic coordinates of the SIFT feature $(x,y)^T$ in *XY* coordination at the current frame (*Frame i*); $(x_1, y_1, 1)^T$: homographic coordinates of the SIFT feature $(x,y)^T$ at the first frame; *i* is the frame index. *H* is the homography matrix with 8 parameters *(a,b…h)*.

However, in the guitar tracking, as the guitar neck area has nearly identical vertical fret and horizontal string as Fig. 4.12 shows, the SIFT feature points in the guitar neck area are easily mis-matched. Therefore, it is difficult to calculate the homography matrix H in Eq. (4.5) due to too many wrong matches (the outliers in RANSAC), whose examples are shown in the left of Fig. 4.13. In other words, the traditional RANSAC could not filter out the mismatched feature points, and the correct matches of SIFT (called in inliers) could not be found either, as shown in the middle of the Fig. 4.13 [82].

The modified RANSAC [82] that aims at solving the above-mentioned problems as follows:

The detected SIFT features are matched between the first and other (second or later) frame as mentioned in Section 4.2. As the right image of the Fig. 4.13 shows, the matched features are connected between the two frames by line segments (pink line segments in Fig.4.13). The matching lines between the two frames are represented by:

$$\widehat{L_\iota} = \left(l_{1,\mathrm{i}}, l_{2,i} \dots l_{N,i}\right) \tag{4.7}$$

where, $\widehat{L_\iota}$ is the matching line vector at $Frame\ i$, N is the number of matching lines [82].

In $\widehat{L_\iota}$, matching lines that cross a large number of other lines are removed. Here, if a member $l_{n,i}$ in $\widehat{L_\iota}$ cross with four of the other lines in $\widehat{L_\iota}$, the method eliminates $l_{n,i}$. The method loops all the lines in $\widehat{L_\iota}$ and the remaining matching lines are defined in Eq. (4.8):

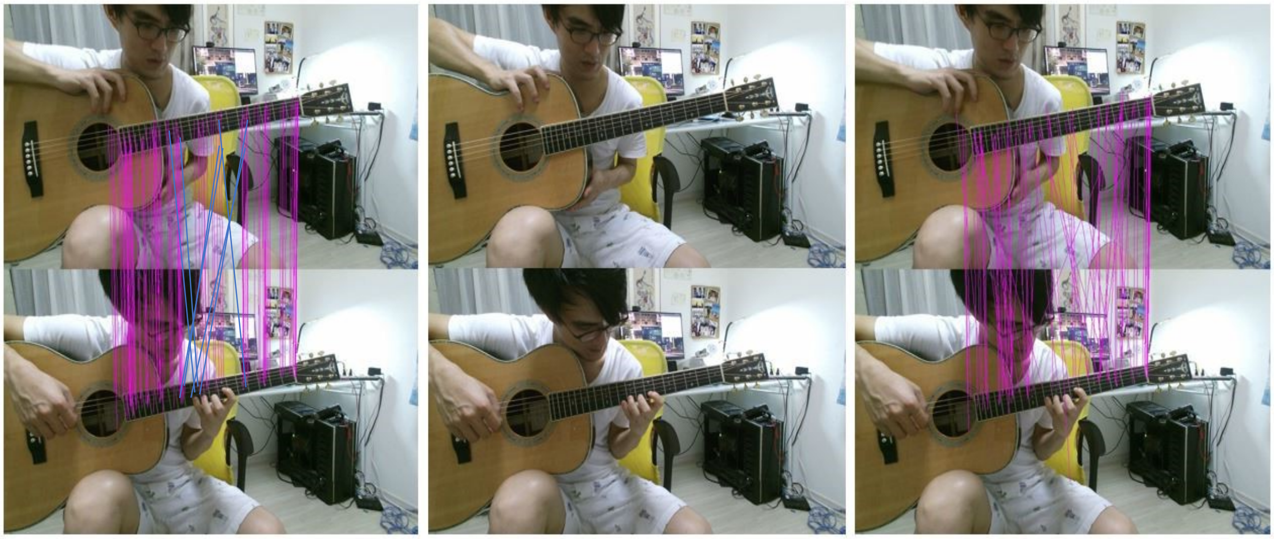


Fig 4.13 Mis-matches (Blue Lines) Present in Guitar Neck Tracking

Left: SIFT Matching Result; Middle: Traditonal RANSAC (cannot calculate Homography due to too many outliers); Right: Filtered Matching Result Based on Modified RANSAC

$$L_i = \left(l_{1,\mathrm{i}}, l_{2,i} \ldots l_{M,i}\right), M < N \tag{4.8}$$

In Eq. (4.8), L_i is called the remaining matching vector at *Frame i*, and M is the number of the remaining lines, where the matching line identifier *n=1,2,3..N in* Eq. (4.7) are re-arranged to *m =1,2,3..M* in Eq. (4.8) [82].

Based on the remaining matching vector L_i, the method applies RANSAC and calculate the homography matrix between the first frame and the current subsegment frame.

$$\begin{pmatrix} x_{i,m} \\ y_{i,m} \\ \omega \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix} \begin{pmatrix} x_{1,m} \\ y_{1,m} \\ 1 \end{pmatrix}, i \in (1,2,3 \ldots I), m \in (1,2..M) \tag{4.9}$$

where: $(x_{i,m}, y_{i,m}, \omega)^T$: homographic transform of the m-th SIFT feature $(x,y)^T$ in *XY* coordination at the current frame (*Frame i)*; $(x_{1,m}, y_{1,m}, 1)^T$: homographic transform of the m-th SIFT feature $(x,y)^T$ in *XY* coordination at the first frame; i is the frame number. M is the number of the remaining matching lines in Eq. (4.8). The inliers of the modified RANSAC (remaining matching lines) are shown in the right of Fig.4.13[82].

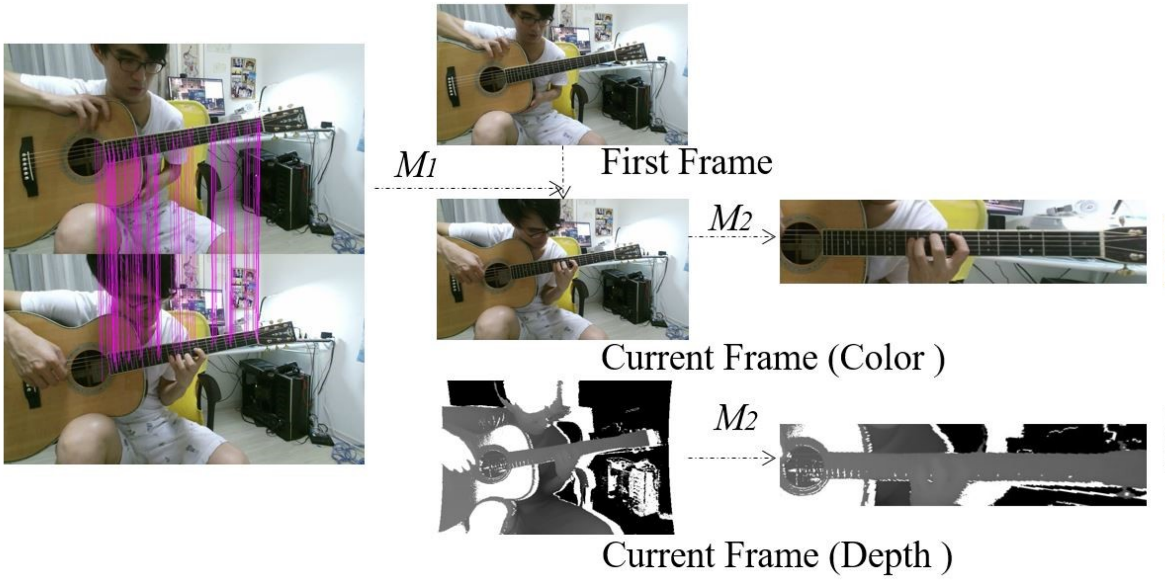


Fig 4.14 Projecting Fretboard Area (M_1 is the homography matrix, M_2 is the perspective transform matrix)

4.5 Projecting Fretboard Area

In Section 4.4, the homography matrix for the inliers (3*3-dimensional matrix in Eq. (4.10)) is the perspective transform matrix of the guitar fretboard area between the first frame and *Frame i* in Eq. (4.10). Given the coordinates of the four corners of the guitar fretboard at the first frame, the method calculates the coordinates of the four corners of the fretboard at frame *i* based on the homography matrix as follows [82]:

$$\begin{pmatrix} x_{i,w} \\ y_{i,w} \\ \omega \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix} \begin{pmatrix} x_{1,w} \\ y_{2,w} \\ 1 \end{pmatrix}, i \in (1,2,3 \ldots I), w \in (1,2..W = 4) \tag{4.10}$$

where: $(x_{i,w}, y_{i,w}, \omega)^T$: homographic coordinates of the w-th corner of the guitar neck at the current frame (*Frame i*); $(x_{1,w}, y_{1,w}, 1)^T$: homographic coordinates of the w-th corner of the guitar neck at the first frame: *Frame 1*; I is the frame number; W the is corner number of the guitar neck equals to 4 [82].

After the four corners of the guitar neck are tracked in Eq. (4.10), the method projects the tracked guitar neck area at each frame to the new image sequences, in which the guitar neck area is always centered. The whole projecting process is presented as [82]:

$$X_{i'} = M_2 X_i \ , i \in (1,2,3 \ldots I) \tag{4.11}$$

$$\begin{pmatrix} x_{i,w'} \\ y_{i,w'} \\ \omega \end{pmatrix} = \begin{pmatrix} i & j & k \\ l & m & n \\ o & p & 1 \end{pmatrix} \begin{pmatrix} x_{i,w} \\ y_{i,w} \\ 1 \end{pmatrix}, i \in (1,2,3 \ldots I), w \in (1,2..C = 4) \tag{4.12}$$

where M_2 is the projecting matrix with 8 parameters *(i,j...p)*; $X_i = (x_{i.w}, y_{i,w})^T$ is the tracked w-th corner of the guitar neck at the current frame (*Frame i)* in Eq. (4.10), $X_{i'} =(x_{i.w'}, y_{i,w'})^T$ is the position of w-th corner of the guitar neck in the new image sequence. As mentioned before, the guitar neck area should be constantly projected to the fixed center of the new image sequence no matter how the guitarist swings the guitar during playing. Given the X_i and $X_{i'}$ (X_i is the tracked corner of the guitar neck, $X_{i'}$ is the fixed position of the corner in the new image sequence), the method can easily calculate M_2 , and project the color image and the depth image to the new image sequence as shown in Fig. 4.14.

4.6 Experiments

4.6.1 Dataset and Experimental Condition

The dataset used for the guitar neck tracking module includes 50 videos of guitar playing with nearly 300 frames of the color images (also 300 frames of depth) taken by a depth sensor Microsoft Kinect.

The whole dataset includes three kinds of music pieces, which are the daily practices most frequently used by guitarists: (1) ***C*** major on the first fret and (2) symmetrical exercise, all of which are fundamental, classic practices, but best way to improve dexterity, speed, strength and stamina to help guitarists to overcome obstacles and become a better guitar player. All the data are taken under different illumination situations (day light, incandescent lights etc.) and complex backgrounds. All the videos of guitar playing are taken by 10 different guitars to show the generality of the proposed algorithm [82].

For the experiments, the system was implemented to a windows 10 desktop with a 3.0 GHz Intel Core i7 processor and DDR3 16GB memory without GPU acceleration. The camera is a Microsoft Kinect, which takes color image sequence and depth image sequence with the same resolution 1200*800. All the algorithms are implemented in Visual Studio 2013 with C++ and OpenCV 2.4.10 library.

4.6.2 Self-comparison

The experiment self-compares as follows: two features: SIFT and SURF are combined with two methods: the proposed method (SIFT + Modified RANSAC, SURF + Modified RANSAC) and traditional RANSAC (SIFT + Traditional RANSAN, SURF + Traditional RANSAC). As shown in Table 4.1, four combinations are experimentally compared. For each combination, the mean error of estimating the X_g, Y_g,

Table 4.1 Accuracy of Self-comparison (X_g *,* Y_g *and* Z_g are guitar coordinates in Fig 3.1 *) (mm)*

	Mean Error of Upper-left Corner			Mean Error of Lower-left Corner			Mean Error of Upper-right Corner			Mean Error of Lower-right Corner			Mean	Variance
	X_g	Y_g	Z_g	X_g	Y_g	Z_g	X_g	Y_g	Z_g	X_g	Y_g	Z_g		
SIFT + Modified RANSAC (mm)	2.3	2.4	5.6	3.3	2.3	6.0	2.7	2.8	6.3	4.4	4.5	7.4	4.2	1.5
SURF + Modified RANSAC (mm)	2.5	2.0	5.8	2.8	2.7	7.3	3.3	3.1	6.3	5.6	5.7	8.0	4.6	1.7
SIFT+RANSAC (mm)	5.3	5.7	8.9	4.4	3.9	9.2	5.5	5.6	10	6.4	7.0	11	6.9	3.44
SURF+RANSAC(mm)	6.2	6.6	9.0	6.4	6.4	9.8	4.7	4.3	7	8.8	7.9	13	7.5	3.98

Table 4.2 Self-comparison of Time Efficiency

	Processing Time Per Frame	FPS
SIFT+Modified RANSAC	2.5 s	0.4
SURF + Modified RANSAC	2.1 s	0.47
SIFT+RANSAC	2.3 s	0.43
SURF + RANSAC	1.7 s	0.58

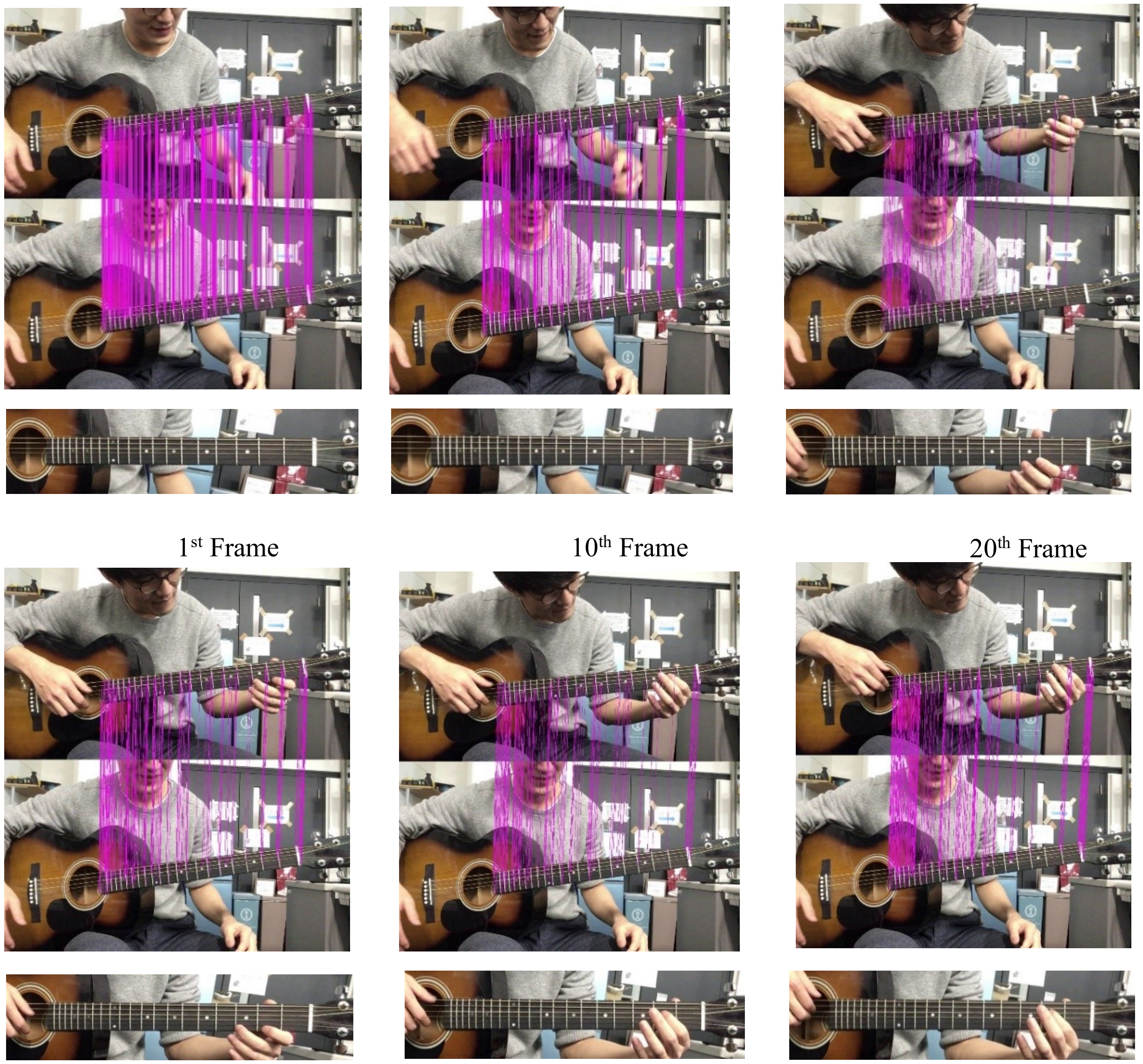
and Z_g coordinates for each of the four corners is listed. In addition, the mean and variance for the four corners are presented.

From Table 4.1, it turns out that the proposed method (SIFT + Modified RANSAC) outperforms the other combinations (SURF + Modified RANSAC, SIFT + Traditional RANSAN, SURF + Traditional RANSAC) with total mean error of 4.17 mm, variance of 1.5 mm. Also, the modified RANSAC either combined with SIFT or SURF (SIFT + Modified RANSAC, SURF + Modified RANSAC) highly outperform the traditional methods (SIFT + Traditional RANSAC, SURF + Traditional RANSAC), which indicates the proposed method is more effective than the traditional RANSAC in the guitar neck 3D tracking case while combining with SIFT or SURF. Other details of self-comparison could be found at Table 4.1.

As shown in Table 4.2, the experiment also compares the time efficiency for these combinations. The traditional RANSAC combined with SIFT meanly runs at 2.3 sec per frame while accelerating with KD-Tree searching. The modified RANSAC only needs extra 0.2 sec to filter out the mismatch, which means it takes 2.5 sec (0.4 FPS) to process a frame.

Two examples of the result of the Modified RANSAC for the neck tracking are shown in Fig. 4.15 and Fig. 4.16, respectively.

Fig. 4.17 shows tempered changes in the mean error of the four combinations for the neck tracking. Overall, the proposed method constantly achieves the smallest error.



1st Frame

10th Frame

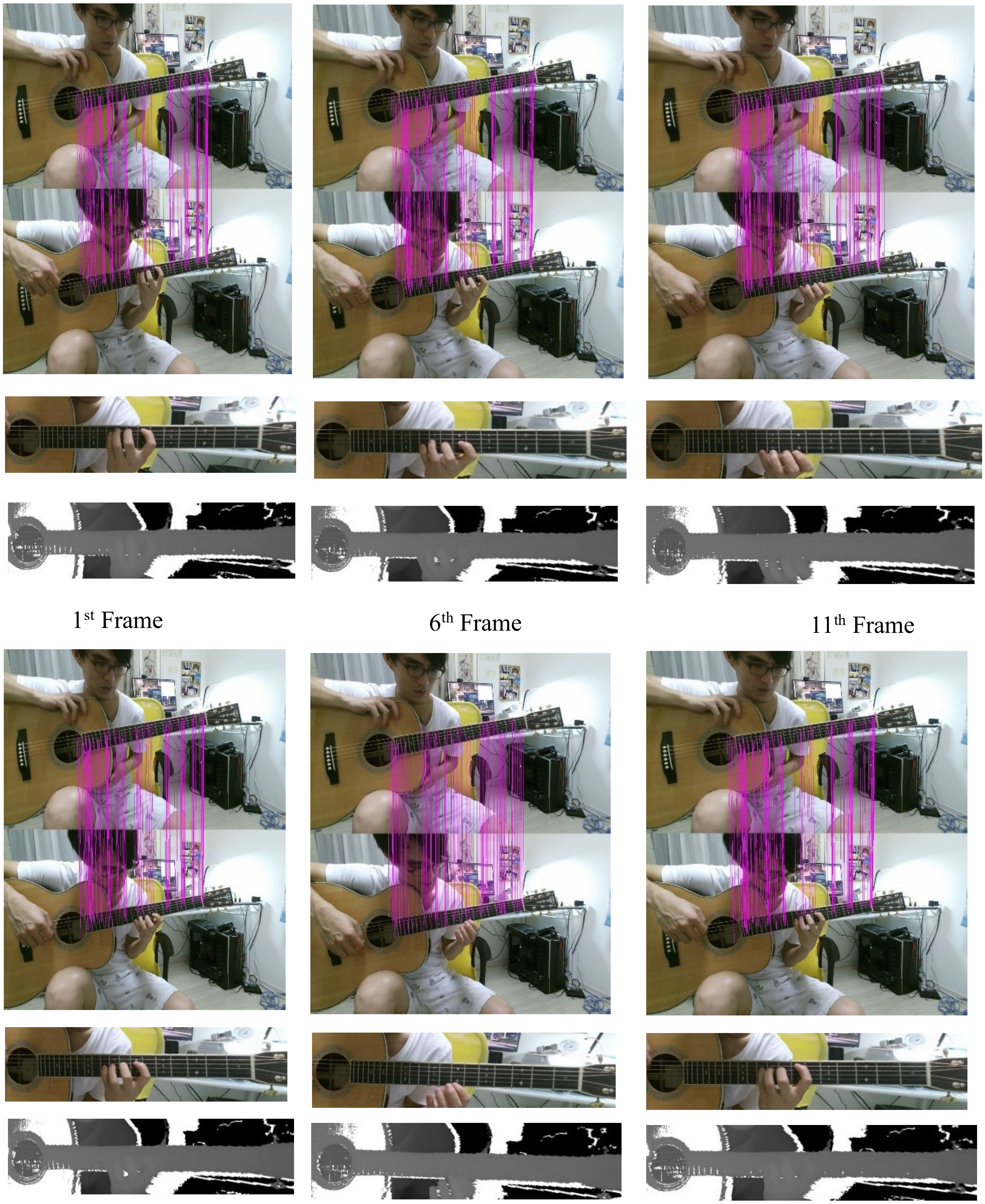
20th Frame

30th Frame

40th Frame

50th Frame

Fig 4.15 Example of Tracking Result Based on Modified RANSAC (1/2)



1st Frame

6th Frame

11th Frame

16th Frame

21th Frame

26th Frame

Fig 4.16 Example of Tracking Result Based on Modified RANSAC (2/2)

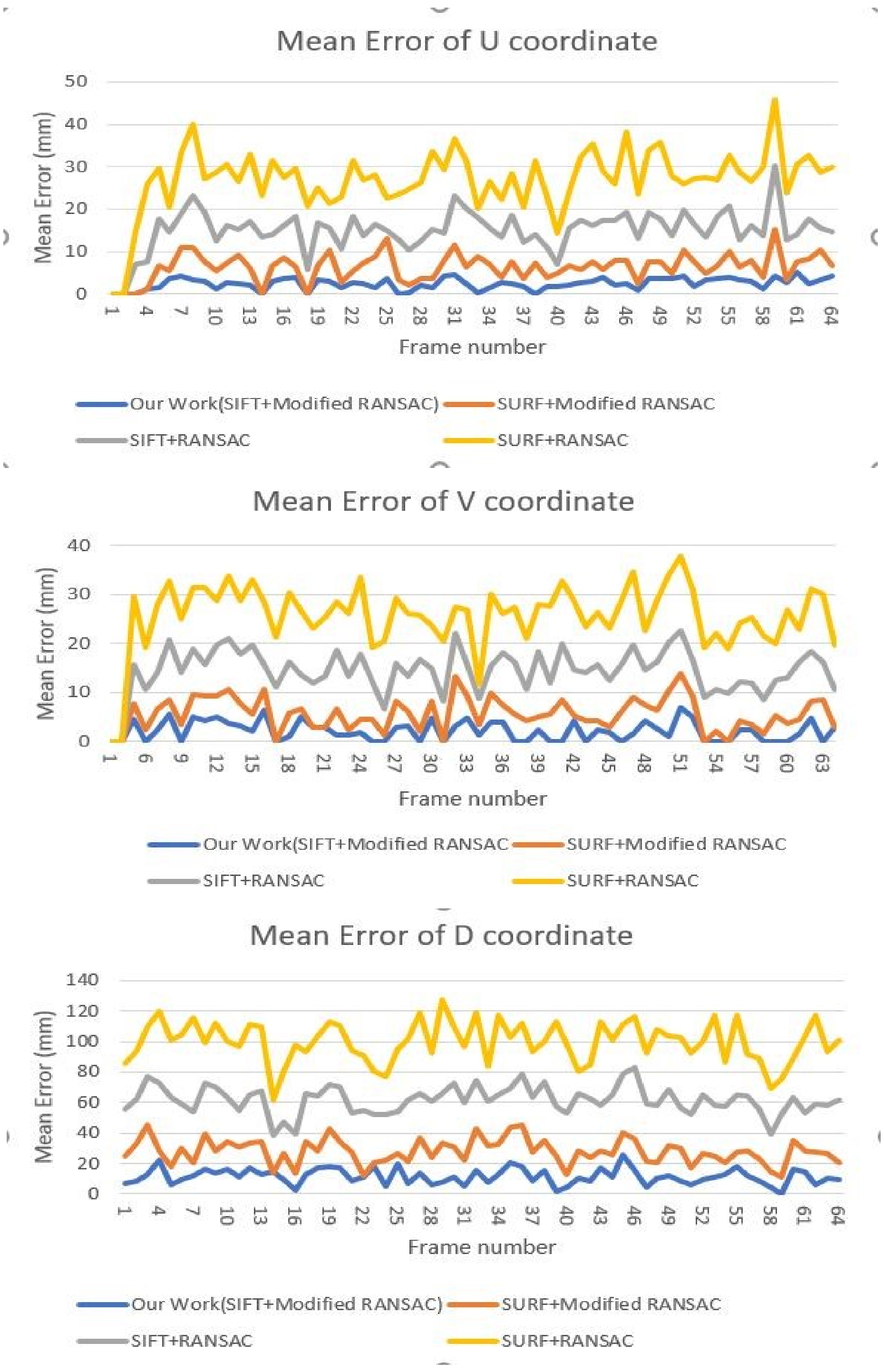


Fig 4.17 3D Tracking Error of An Example Image Sequence

4.6.3 Comparison with Related Works

The proposed method is experimentally compared with the related works [9, 83, 107] of guitar neck tracking algorithms, which do not need supporting tools such as AR tag or fixed cameras. Besides, the experiment also compares the work with state-of-art deep learning method, which is based on Fully-convolutional network [107]. For [9], it is easy to detect lines by Hough Transform and remain the largest cluster of lines that have the same slope; for [83], as the paper said, the experiment applies optical flow to track the 40 detected Shi-Tomasi Features (as [83] writes 40 points is the best performance); for [107], the experiment implemented a 7-level fully convolutional network, that is identical to VGG16, except replacing the last 3 fully-connected to 3 convolutions.

Figure 4.18 shows the comparison result. The experiment applies a general comparison method that is widely used in recent tracking research works. The horizontal axis indicates the threshold for the mean error of tracking, while the vertical axis means the accuracy of tracking for each threshold value. The proposed work outperforms others by achieving 100% when the threshold is set to 8 *mm* or smaller.

Table 4.3 gives a numerical comparison of time efficiency and mean tracking error with the works mentioned before. From Table 4.3, the proposed work (0.4 FPS) is much less efficient than Fully-convolutional net (35 FPS) [107], but Fully-convolutional net also needs at least 400 images to label and train, which would cost 10 hours to train with GPU acceleration. More importantly, the mean error of 4.2

Table 4.3 Comparison of Time Efficiency and Accuracy with Related Works

	Time Efficiency(FPS)	Mean Tracking error (mm)
Proposed Method	0.4	4.2
Optical flows based [10]	0.02	8.3
Fully Convolutional Net [107]	35	10.1
J. Scarr and R. Green [9]	0.89	13.3

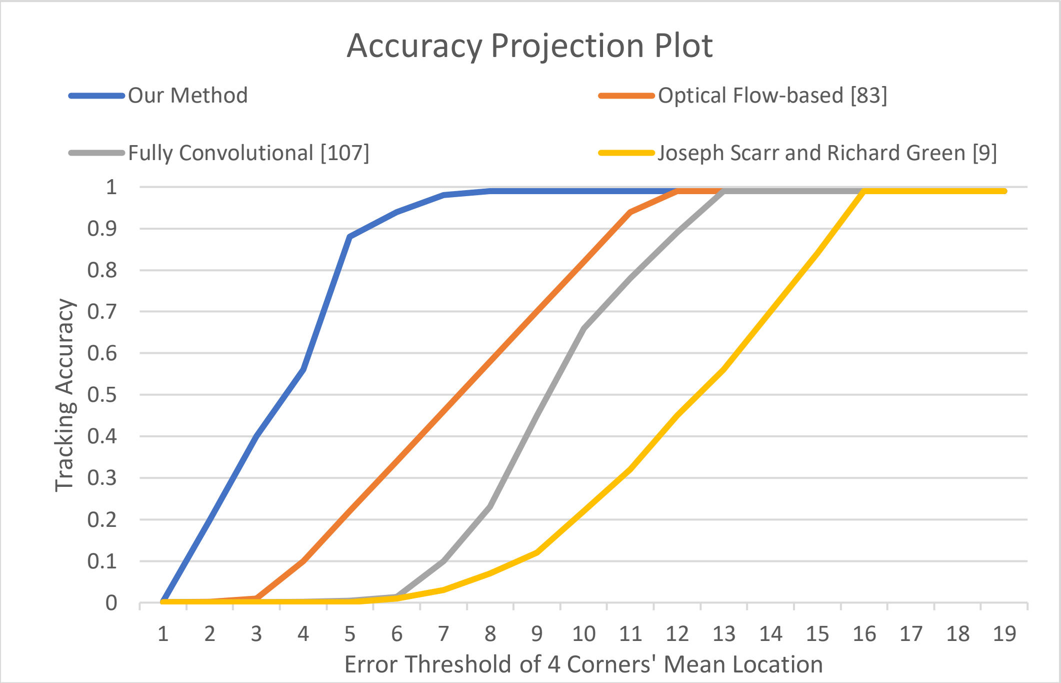


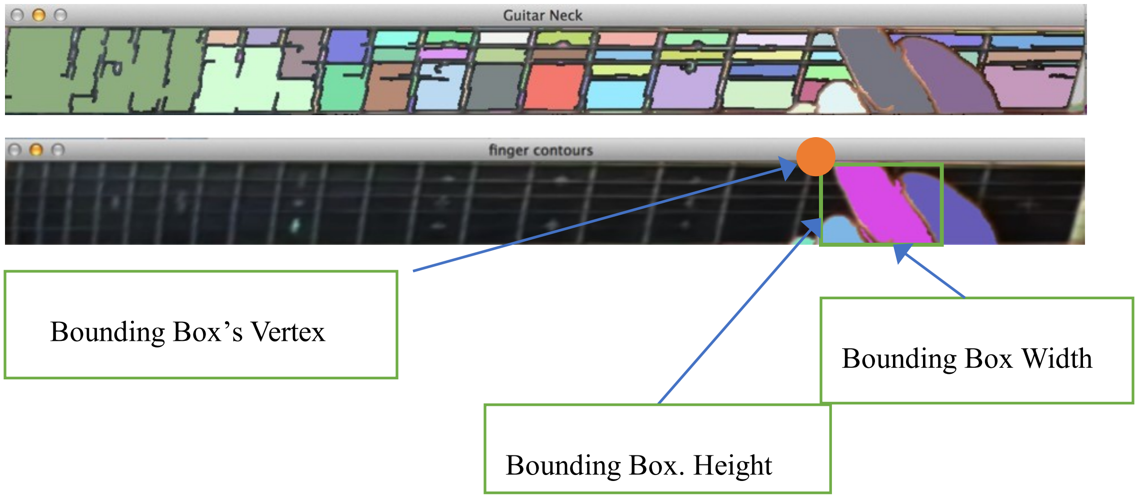
Fig 4.18 Comparison with Related Works [9, 83, 107]

mm by the proposed work is only 4/10 of the distance between adjacent strings on the guitar fretboard, while 10 *mm* (other related work's mean error) is almost the distance between the strings on guitar fretboard, which means if the mean tracking error is over 10 *mm*, it is very difficult to analyze which string is pressed by fingers in the future work.

4.6.4 Comparison by Guitar Chord Recognition

Related work [9] detects the pressed guitar chord by extracting finger contours according to the following four criteria (a) to (d): (a) at least 50% of the pixels of a finger contour must have skin color (R>95, G>40, B>20); (b) finger contours must intersect with the lower edge of the guitar neck; (c) a finger contour must be proportionate (the height of the bounding box (Fig. 4.19) of the finger contour must be shorter than the fourfold of the width) [2]; (d) the area of each finger contour must be larger than 36 pixels. Finally, based on the top-left corners of the detected four finger contours (as shown in Fig. 4.19), the 2D position of each fingertip could be detected on each Key Frame [9,116], which is the frame when the guitarist changes the hand shape to press the chord. Some examples of chord recognition by using SIFT and Modified RANSAC are shown in Fig. 4.19.

In this section, the same four criteria are used to detect the chord, and the only difference is that in related work [9], it uses a Hough Transform based method to detect the guitar neck at each frame; while in



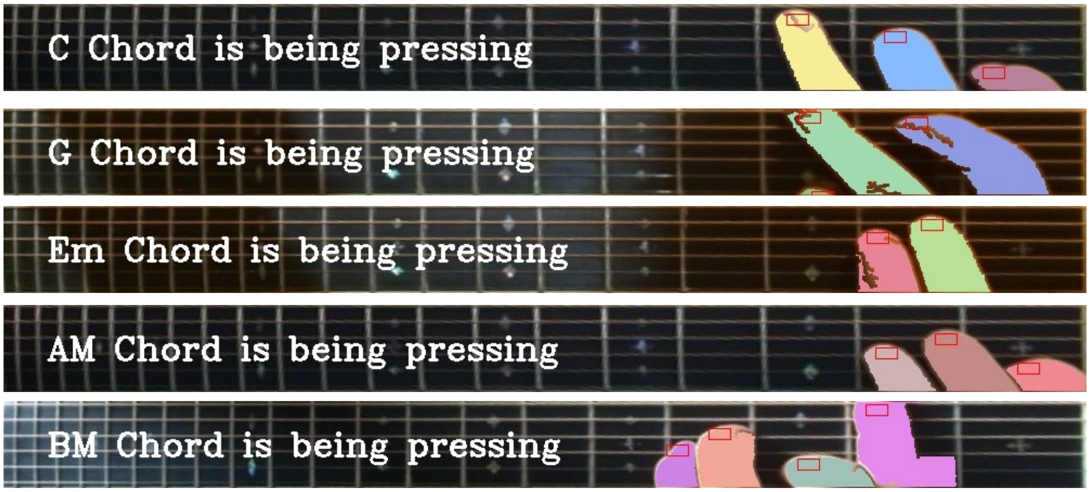
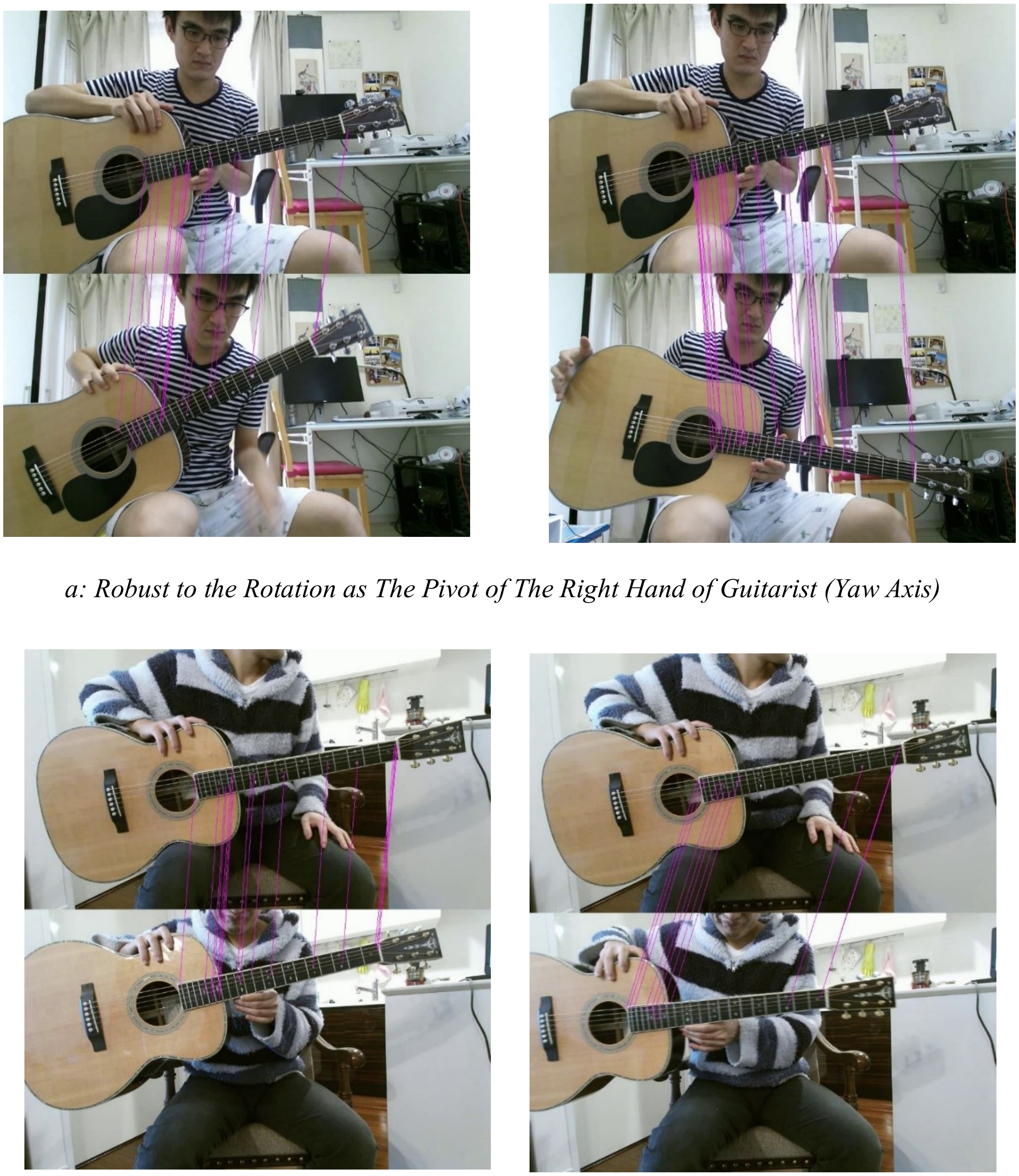

C Chord is being pressing
G Chord is being pressing
Em Chord is being pressing
AM Chord is being pressing
BM Chord is being pressing

Fig 4.19 Examples of Chord Recognition

this chord recognition, the SIFT and Modified RANSAC is used to track the guitar neck. The mean chord recognition rate of SIFT and Modified RANSAC based method is 93.8%. Compared with the mean chord recognition rate in related work [9], which is 52%, it demonstrates a significant increase in accuracy due to the accurate guitar neck tracking method.

4.6.5 Experiment on Robustness on Rotation and Translation

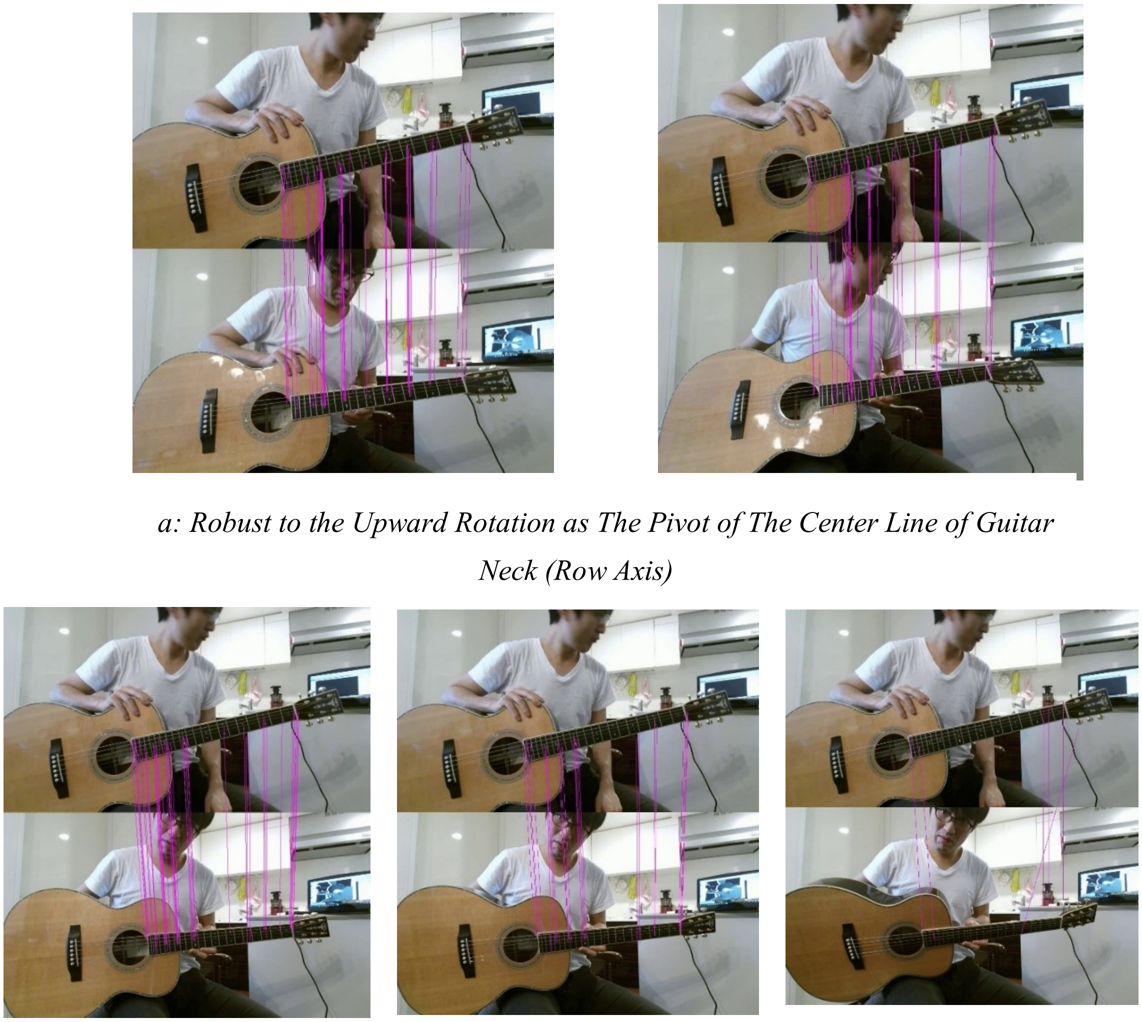
As guitarists may swing and move guitar during playing, the robustness of the guitar necking tracking over the rotation (a. rotating guitar about the right hand (Yaw rotation in Fig.3.2); b. rotating guitar about the neck center line (Row rotation in Fig.3.2) and c. rotating guitar about the horizontal center of guitar neck ($Pitch$ rotation in Fig.3.2)) and translation are tested. The results are show in Fig. 4.20, Fig. 4.21 and Fig. 4.22 respectively. Based on the experimental result, (1) it is very robust to the translation movement of guitar neck: as long as the guitar neck area is wholly in the image, no matter how it moves, guitar neck can be tracked; (2) it is robust to the rotation movement using the right hand of the guitarist as the pivot of the



a: Robust to the Rotation as The Pivot of The Right Hand of Guitarist (Yaw Axis)

b: Robust to the Translation Movement of Guitar Neck

Fig 4.20 Experiment on Robustness over Rotation (Yaw Axis) and Translation



a: Robust to the Upward Rotation as The Pivot of The Center Line of Guitar Neck (Row Axis)

b: not Robust to the Downward Rotation as The Pivot of The Center Line of Guitar Neck (Row Axis)

Fig 4.21 Experiment on Robustness over Rotation (Row Axis)

rotation (Yaw rotation in Fig. 3.1); on average, the limitation angle of the rotation is 20 degree: if the angle of the guitar necks of SIFT-matched two frames is larger than 20 degree, the guitar neck cannot be tracked anymore; (3) it is robust to the upward rotation using the center line of guitar neck as the pivot (Row rotation in Fig. 3.1) as shown in Fig. 4.21.a, the limitation angle of the rotation is on average 8 degrees, but it is not robust to the downward rotation using the center line of guitar neck as the pivot of the center line of the guitar neck (Row rotation in Fig. 3.1) at all as shown in Fig. 4.21.b. This is because when the guitar neck is rotated downward, the intensity of pixels greatly changes because the lighting source is always upon the guitarist; therefore, the SIFT features changes drastically and they cannot be matched anymore; (4) it is very robust over the rotation movement using the horizontal center of the guitar neck as the pivot of the

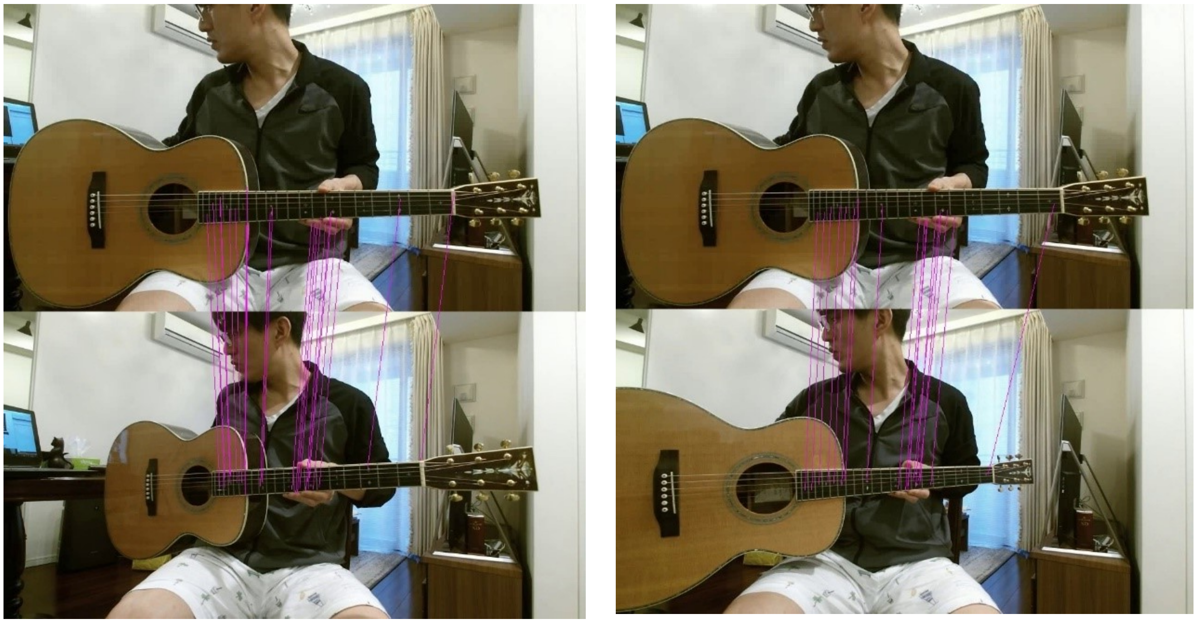


Fig 4.22 Experiment on Robustness over Rotation (Pitch Axis)

rotation ($Pitch$ rotation in Fig.3.2): the limitation angle of the rotation is averagely 15 degrees as shown in Fig. 4.22.

4.7 Discussion

In this chapter, the proposed guitar neck tracking module aims at accurately tracking the guitar neck during the guitar playing despite of occlusion-by-hand, rotation and translation movement, and different illumination and background conditions. The contribution of this chapter is as follows:

(1) Despite occlusions caused by the guitarist's hand during guitar playing, the guitar neck can be tracked much more accurately than related works [9, 10, 17]. The mean tracking error is only 4.2 *mm,* which is only 4/10 of the distance between adjacent strings on the guitar fretboard. Furthermore, the variance value 1.5 indicates that the proposed method is also stable and robust enough to further the subsequence research effectively.

(2) To deal with the case in which guitarists move the guitars during playing, the rectification method for the guitar neck area at each frame is proposed so that the neck is centered, and the neck's long and short sides are parallel to the horizontal and vertical axes at each frame. This method makes the subsequence process of this thesis easy to conduct, because once the guitar neck is tracked and rectified to the centered position, it is easy to recognize or assess the fingering of guitarist by only

analyzing the hand pose information of the guitarist. p

(3) Limitations of guitar motion are experimentally measured. The rotations about the right hand (Yaw rotation), the neck center line (Row rotation) and neck horizontal center ($Pitch$ rotation), as well as translations are measured. The limitation for Yaw rotation is 20 degrees; the limitation for upward Row rotation is 8 degrees, but it is not robust to downward Row rotation; the limitation of $Pitch$ rotation is 15 degrees. On the other hand, there is no limitation for the translation movement of guitar neck as long as the guitar neck area is wholly in the image. Limitation test shows the proposed method tracks the guitar neck effectively and robustly even under the situation that guitarist shakes or swing the guitar neck aggressively on purpose. Generally, the method is robust to the 3-dimensional rotations and translation movement of guitar.

The guitar neck tracking achieves the above contribution is because of the following reasons:

(1) SIFT feature points, which are invariant to rotation, illumination and scale are used as the accurate and robust feature in this chapter. Compared with other features, such as Shi-tomasi, SIFT feature achieves the most accurate tracking result is Section 4.6. Besides, by applying a KD-tree to accelerate the whole process, the proposed method tracks the guitar neck in an efficient manner.

(2) A modified version of RANSAC (Random Sample Consensus) is proposed in this chapter to overcome the occlusion-by-hand issue. As mentioned earlier, feature points within the guitar neck area cannot be accurately tracked or matched, because it is overlapped and occluded by fingers of guitar players. The proposed modified RANSAC-based filtering algorithm filters out and eliminates the mismatched feature points, and then calculates the homography between the correctly matched feature points on the first frame and on the any other frame to track the guitar neck. Besides, since the homography is calculated between the first frame and any other frame, the proposed methods do not need to concern about the tracking failure problem.

(3) To suppress the effect of the guitar neck motion, the tracked guitar neck area on every frame is rectified to the center of the image, where the horizontal and vertical guitar sides are parallel to the image axis. Owing to this rectification, no matter how the guitar player shakes or swings the guitar neck while playing, the neck area on every frame is always rectified in the rectified manner to facilitate analyzing the guitar fingering in Chapter 7.

4.8 Conclusion

Chapter 4 proposes an algorithm for tracking the 3D position of the fretboard from the video of guitar plays. Specifically, this module proposes a SIFT matching procedure to track the guitar neck in 3D. First, this module detects the SIFT features within the guitar fretboard and then match the detected points using KD-tree searching based matching algorithm frame by frame to track the whole fretboard. However, during the guitar plays, since the performer's fingers frequently overlap the fretboard, the feature points cannot always be matched accurately. Therefore, by using the modified RANSAC algorithm that filters out the tracking error of the feature points due to the overlapping issue mentioned before, a perspective transformation matrix called Homography is obtained between the correctly matched feature points detected at the first and other frames. Consequently, the guitar neck is tracked correctly based on the perspective transformation matrix.

Experiments under different conditions show promising results of the proposed method. High accuracy: the total mean tracking error is only 4.2 *mm* and variance are 1.5 *mm* for tracking the four corners of the guitar fretboard is obtained. This result outperforms related tracking works including state-of-art Fully-convolutional Network.

Chord Recognition is performed. The experiments for pressing the proper chord is measured. The accuracy of chord recognition is 93.8%, which is much better than a conventional method.

Chapter 5 Tracking Fingertip of Guitarist

5.1 Introduction

As depicted in Fig.5.1, first, after inputting a video of guitar play, a CNN-based hand segmentation net is used to discriminate the hand area from the background. Then, the template matching and reversed Hough Transform are performed to the hand areas so that the count map for fingertip candidates is generated using the segmentation result, where the results of the template matching and reversed Hough Transform are used as weighted features to extract the fingertip candidates. Furthermore, a temporal grouping is applied to remove noise and group the same four fingertips (index finger, middle finger, ring finger, little finger) on the successive count maps. Then, an ROI-association algorithm is utilized to associate the four fingertips with their individual trajectories on the frame-by-frame count maps. Here, for this ROI association algorithm, three patterns for tracking fingertips movement during the whole process are defined: the active pattern, adding pattern, vanishing pattern. All the tracked trajectories of fingertip candidates are fitted into these three patterns in order to solve the problem such as self-occlusion etc. Finally, this module use the ROI associated particle filter to track the fingertips by distributing particles within the associated ROIs of fingertips at every two adjacent frames of the video.

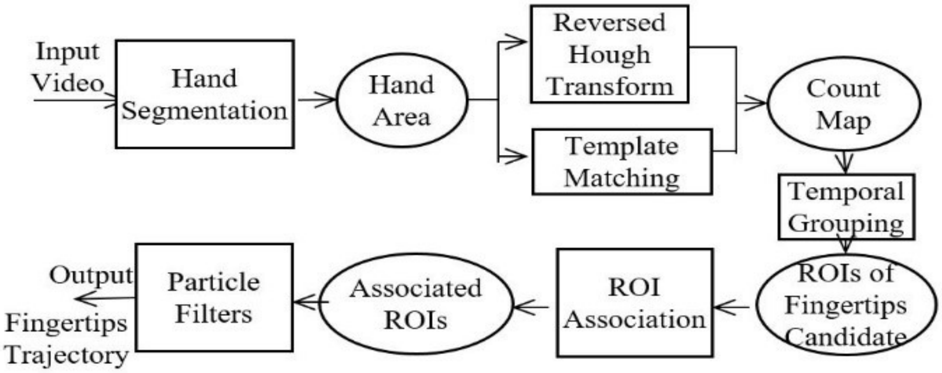
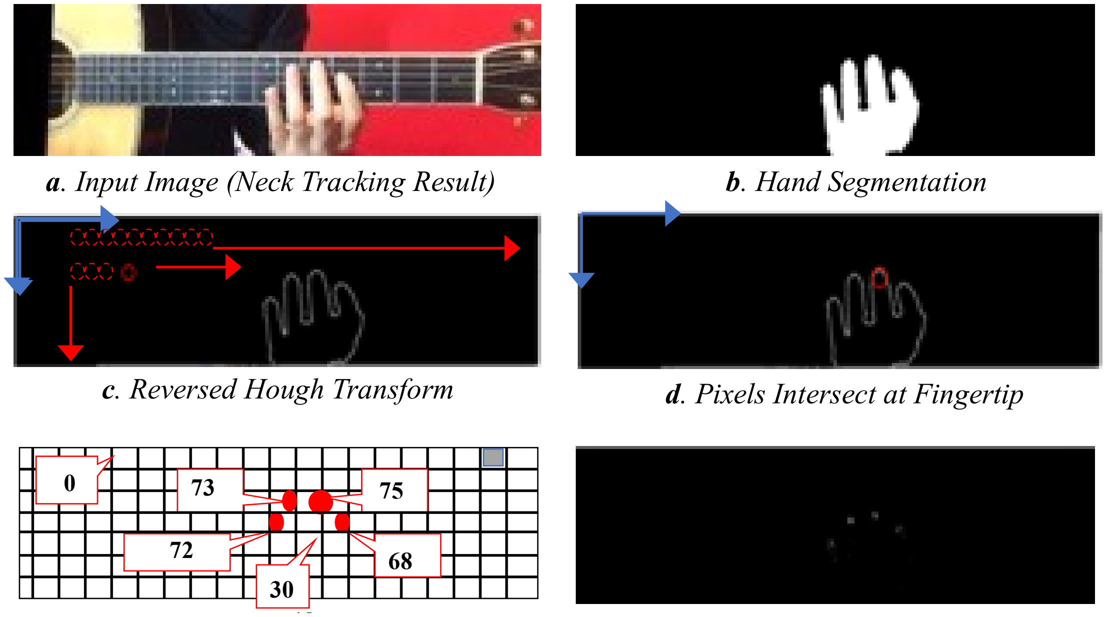


Fig.5.1 Diagram of the Method (□ *:Module* ○ *: Data)*

5.2 Features for Localizing Fingertips

After the hand segmentation (as the hand segmentation is deep learning-based method, introduced in Section 6.2), this module detects candidates of the fingertips in the hand segmentation result by applying both reversed Hough Transform and template matching as the features of the fingertips, because the most distinctive feature of the fingertip is its semi-circle shape.

The reversed Hough Transform (RHT) is performed for the segmented hand area (Fig.5.2. b). Specifically, the segmentation result, whose size is same as the input image (Fig. 5.2.a), is traversed pixel by pixel (from left to right, from top to bottom as the blue arrows in Fig. 5.2.c show), and a set of circles, whose radius ranges between 15 and 20 pixels, is generated at each pixel (Fig. 5.2.c). The counts of the pixels at which the set of circles and the hand contour intersect are saved in the accumulator (Fig. 5.2. d) that has the same size with the hand segmentation result (Fig. 5.2. b). Figure 5.2.e shows an example of how the counts are saved in the accumulator In Fig. 5.2.e, each cell indicates a pixel in the accumulator, and all the cells store the count of the intersection. From Fig. 5.2.d, it is clear that the pixels around fingertips own higher counts than other pixels, because fingertip contours, which have semi-circle shape, intersect the circles more frequently.



a. *Input Image (Neck Tracking Result)*

b. *Hand Segmentation*

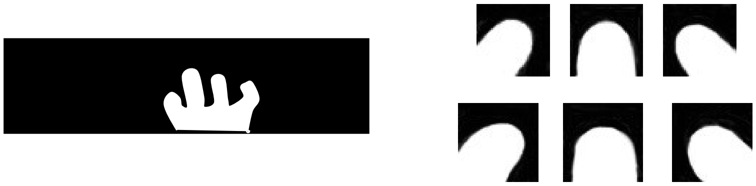
c. Reversed Hough Transform

d. *Pixels Intersect at Fingertip*

e. *A Same Size Accumulator Recording the count of the Intersection*

f. *Count map (Brighter Pixels Indicate High Probability)*

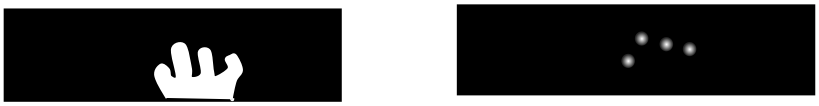
Fig.5.2 Reversed Hough Transform



***a**. Hand Segmentation Result*

***b**. Six Templates with different Sizes and Orientations of Fingertips*

Fig.5.3 Template Matching



***a**. Hand Segmentation*

***b**. Generated Count map of Fingertip*

Fig.5.4 Count map based on Template Matching and Reversed Hough Transform

For the Template Matching (TM), instead of using the three templates used by Kerdvibulvech's method[9)], this module uses six templates (Fig. 5.3.b) to localize the fingertip candidates' positions considering the directional variance of fingertips during guitar plays. As detailed in Eq. (5.3), the accumulator cells at which the template matches the finger contours frequently store large counts.

Both of the RHT and TM features are weighted as indicated by Eq. (5.1), because based on the test results, this combined weighted method works much better than either one.

$$Fin_{(x,y)} = \alpha\, T_{sum(x,y)} + (1-\alpha)R_{sum(x,y)} \tag{5.1}$$

where $Fin_{(x,j)}$ indicates the fingertip candidate detection result at pixel *(x,y)* in the segmentation result, where *(x,y)* is the 2D coordinates in *XY* coordination system; $T_{sum(x,y)}$ and $R_{sum(x,y)}$ indicate the results of applying TM and RHT at pixel *(x,y)*, respectively; α indicates the weight that ranges between 0.0 and 1.0. In Eq. (5.1), $R_{sum(x,y)}$ and $T_{sum(x,y)}$ are calculated by Eq.(5.2) and Eq.(5.3), respectively.

$$R_{sum(x,y)} = \sum_{r=15}^{25} \sum_{(x',y')} \{(x',y') \in l \cap (x'-x)^2 + (y'-y)^2 = r\} \tag{5.2}$$

where, *(x',y')* is the circle centered at *(x,y)* with the radius of *r, l* is the set of hand contour pixels.

$$T_{sum(x,y)} = \sum_{i=0}^{N_0} \frac{\sum_{x',y'}[T(x',y') - H(x+x',y+y')]^2}{\sqrt{\sum_{x',y'}(T(x',y')^2) \sum_{x',y'} H(x+x',y+y')^2}} \tag{5.3}$$

where *T(x,y)* indicates the template at *(x,y)* and *H(x, y)* is the current frame at which TM is performed, N_0 is the number of the fingertip templates (Fig. 5.3.b) and is equal to six

Finally, *Fin(x,y)* in Eq.(5.1) is normalized as follows:

$$Fin_{Normal} = \frac{Fin_{(x,y)}}{Fin_{(x_{max},y_{max})}} \times 255 \tag{5.4}$$

where (x_{max}, y_{max}) is the position of the largest score at the current frame.

The accumulator that stores Fin_{Normal} is called the count map, which has the same resolution as the hand segmentation result shown in Fig.5.4.a. As the count map is displayed in Fig.5.4.b, high intensities correspond to large counts, in other words, to fingertips at high probabilities. Then, this module converts the gray-scale count map to a binary image, in which non-zero and zero pixels of the count map are converted to white and black, respectively. Furthermore, this module detects “fingertip candidates” by applying the traditional contour tracing method for non-zero (white) pixels in the binary image.

5.3 Temporal Grouping and ROI Association

5.3.1 Basic Idea

(1) Particle Filter-Only

Traditional particle filter cannot track multiple targets, because it randomly distributes particles over

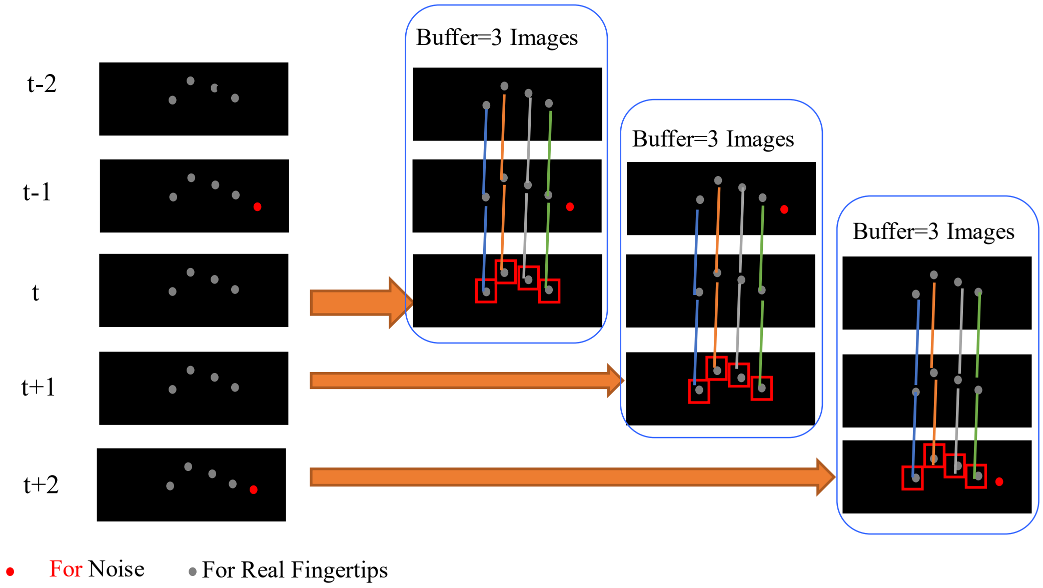


Fig.5.5 Conceptual Image of Temporal Grouping

the whole image; thereby, it cannot discriminate different fingertips, because all the fingertips own very similar appearances, which may vary frequently cause loss or confusion of particles.

(2) Temporal Grouping and Particle Filter

One of the classic methods for multiple target tracking is (1) using ROIs to locate the rough position of each target in the frame, (2) group those ROIs that belong to same target in consecutive frame and (3) distributing particles to accurately track each target. In the work, after this module detects fingertip candidates, which could include noise also, this module generates ROIs, each of which encloses a fingertip candidate by the square based on the grouping algorithm explained in Sec. 5.2.2. Particles are distributed only in the ROIs in order to prevent the mis-distribute-problem described in (1) Particle Filter-only. However, as the movement of fingertips during guitar playing is extremely complex such as joint finger issue or self-occlusion, the generated ROIs are not accurate enough, for example, since the self-occlusion of fingertips happens, the ROI of the occluded finger cannot be generated, this module finally design this algorithm as follows:

3) ROI Association, Temporal Grouping and Particle Filter

On consecutive frames, the ROIs that are associated by the process explained in Sec. 5.3.2: after Temporal Grouping-based generation of ROIs in each frame explained in Sec. 5.3.1. Then, the filtering process explained in Section 5.3.3 is performed to distribute the particles between the associated ROIs.

5.3.2 Temporal Grouping

The reason why this module propose temporal grouping is based on the following two reasons: (1) false detection of fingertips caused by wrong segmentation etc. may appear in temporal sequences randomly (the red dot in the left of Fig.5.5); in contrast, real fingertip candidates (the gray dot in the left of Fig. 5.5) continuously appear during a longer time span, compared with noises that appear randomly; (2) each ROI should correspond to a single fingertip candidate over the temporal sequence (the red squares in the left of Fig. 5.5)

Therefore, at each frame, the proposed temporal grouping focuses on a certain number of the previous frames (also called as buffer): for instance, in Fig.5.5, at Frame t, this module focuses on three frames (*Frame t, Frame t-1, Frame t-2*) if the buffer size is three.

The temporal group is obtained according to the following procedure that obtains the first and second temporal group candidates, as follows. In the buffer with the size B, at frame t, the first temporal group candidate $G'_{t,i}$ for the fingertip candidate i is represented by Eq.(5.5):

$$G'_{t,i} = \{f_{t,i}, f_{t-1,i}, f_{t-2,i} \cdots f_{t-B+1,i}\} \tag{5.5}$$

In Eq. (5.5), $f_{t\text{-}j\text{-}1,i}$ is found at Frame *t-j-1* as the closest fingertip candidate to $f_{t\text{-}j,i}$ at Frame *t-j*, where $0 \leq j \leq$B-2. From the first temporal group candidate $\boldsymbol{G'_{t,i}}$ for *i* at *t*, the second temporal group $\boldsymbol{G''_{t,i}}$ for i at t is obtained by Eq. (5.6).

$$G''_{t,i} = \{f_{t,i}, \{f_{t-t',\ i} | Dis(f_{t-t',\ i}\ ,\ f_{t,\ i}) < Thr,\ t' \epsilon\ (1,2 \ldots B-1),\}\} \tag{5.6}$$

In Eq. (5.6), *Dis(aa, bb)* is the 2D Euclid distance between the centroids of the contour pixels of fingertip candidates *aa* and *bb*; *Thr* is the threshold for the 2D Euclid distance. Note that if $Dis(f_{t,i}, f_{t\text{-}t',i})$ is larger than *Thr*, then $f_{t\text{-}t',i}$ is not included in $G''_{t,i}$.

Finally, at any Frame *t,* for each fingertip candidate $f_{t,i}$, if the number of the members of $G''_{t,i}$ is larger than the threshold *g_Thr*, $G''_{t,i}$ is judged to be a temporal group $G_{t,i}$, and ROIs are given to each member of $G_{t,i}$; otherwise (if smaller than *g_Thr*), $G''_{t,i}$ is judged to be a noise and removed. For example, the red squares in Fig. 5.5 are the temporal groups' members at *t, t-1* and *t-2*, and each group's members in the buffer are linked by differently colored line segments. In Fig. 5.5, the red dot is a fingertip candidate that are judged to be a noise and removed.

5.3.3 ROI Association

As a result of performing the temporal grouping explained in Sec. 5.3.2, fingertip candidates are temporally grouped, and an ROI is generated for each candidate at each frame. However, the process of the temporal grouping does not have a function for tracking each fingertip. Therefore, the ROIs should be linked (associated) temporally: in other words, the associated ROIs consistently correspond to the fingertips, but the number of the associated ROIs and visible fingertips do not always match. To address this issue, this module analyzes and classify each ROI at each frame into the following three patterns: (1) active-pattern, (2) adding-pattern, and (3) vanishing-pattern, as shown in Fig.5.6. Generally, the ROIs are classified by (a) comparing the number of ROIs in the current frame and the number in the previous frame; (b) the distance between each ROI in the current frame and the correspondent ROI in the previous frame. The details of the ROI association are shown as follows:

By comparing the numbers this module classifies the grouped ROIs into three patterns by (a) comparing the number of ROIs in the current frame and the number in the previous frame; (b) calculating the distance between each ROI in the current frame and the correspondent ROI in the previous frame in order to classify all the ROIs at each frame into the above-mentioned three patterns (1) to (3).

(1) active-pattern

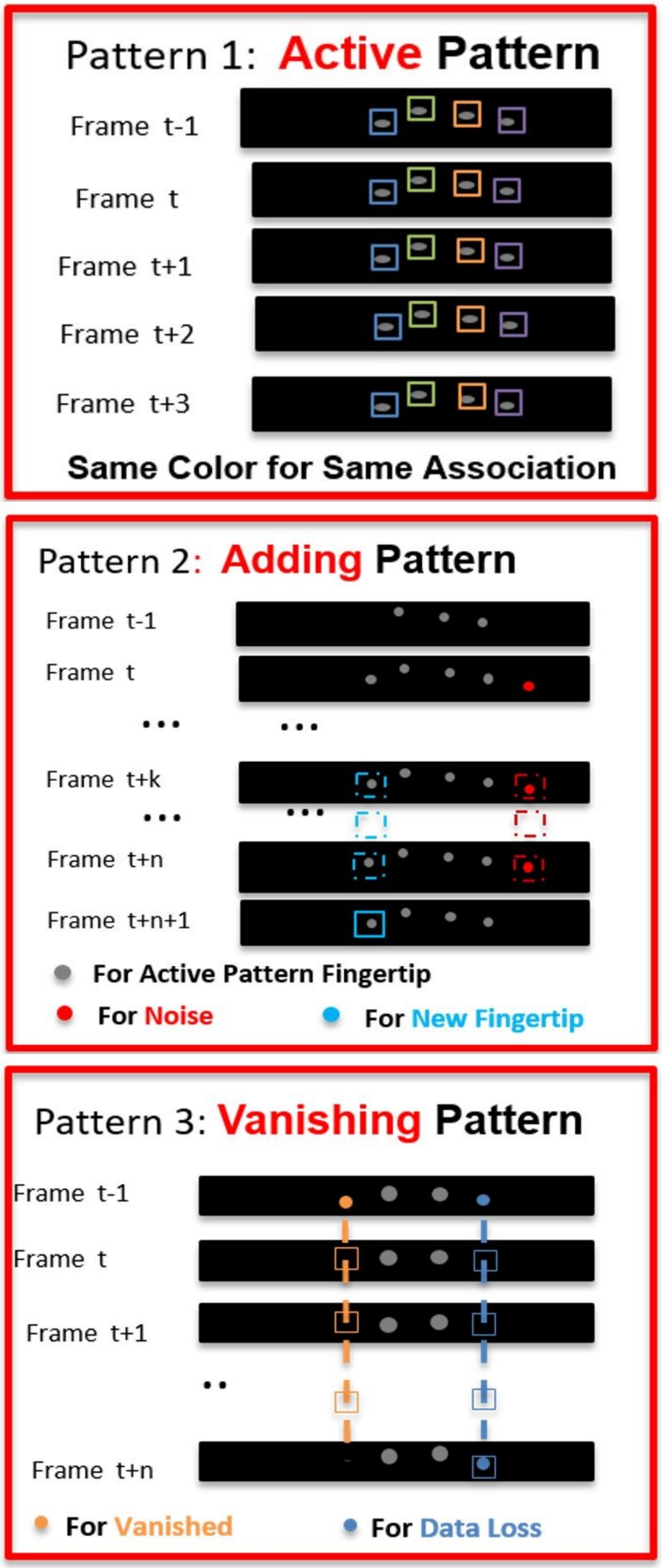


Fig.5.6 Three Patterns of the ROI Association

This pattern corresponds to the case in which all the fingertip candidates' ROIs are associated over two consecutive frames. Specifically, if the following two conditions (i) and (ii) are satisfied in two consecutive frames (Frame *t-1* and *t*), the ROIs in the two frames are judged to be classified into the active patterns: (i) if the number of ROIs at Frame t is equal to the number of ROIs at Frame *t-1*; otherwise, the ROIs are treated as adding or vanishing patterns. (ii) For each ROI in the two frames, the closest (in two-dimensional Euclid distance in the 2D coordinate system defined for each of two consecutive frames) ROI is searched in the other frame, and if the closest distance is smaller than the threshold *r_Thr*, the closest ROIs (one is in Frame *t*, the other is at Frame *t-1*) are associated; otherwise (larger), ROIs in t and t-1 are treated as an adding and vanishing pattern respectively. For instance, in the active-pattern in Fig.5.6, the same colored ROIs indicate the associated ROIs in the two consecutive frames.

(2) adding-pattern

The adding-pattern corresponds to the case in which a new finger gets into a frame (frame-in), or a noise exists continuously (curved finger). In case of the "frame-in", the previous frame has no ROI that can be associated with the frame-in's ROI. A typical example of the above-mentioned continuous noise is that if the guitar player curves his finger for a long time, the curved joint of the finger (not the fingertip) might keep showing a semi-circle shape. The process for finding adding-pattern(s) is conducted according to the following two steps: (i) if the number of ROIs at Frame t is larger than the number of ROIs at Frame t-1, Step (ii) is performed; otherwise (smaller), it should be treated as a vanishing-pattern. (ii) For each ROI in each frame, the closest (in two-dimensional Euclid distance in two consecutive frames) ROI is searched in its previous frame, and if the closest distance is smaller than the threshold *r_Thr,* these two ROIs are associated and treated as active-patterns; otherwise (larger than *r_Thr*,), the unassociated ROIs in t and t-1 are treated as adding-patterns and vanishing-patterns, respectively. The adding-patterns are further checked if they correspond to either "frame-in" (new finger) or noise (curved finger). As an example of the adding-pattern is shown in the Pattern 2 of Fig.5.6, if the adding-pattern(s) last(s) long, the pattern(s) are judged to be (a) new finger(s); otherwise, noise (curved finger) to be removed. More specifically, in n frames after frame t, if the closest distance(s) of the ROIs of the adding pattern(s) between each two adjacent frames are smaller than *r_Thr*, the ROIs are associated, and if this association lasts for the n frames, the associated ROIs are recognized as "new finger(s)".

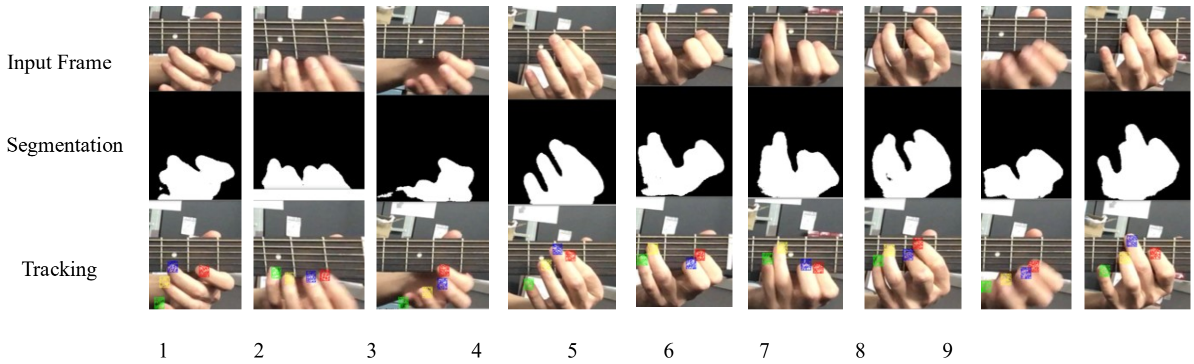
(3) vanishing-pattern

The vanishing-pattern corresponds to the case in which a finger moves out of a frame (frame-out), or a temporary data loss (self-occluded). In case of the "frame-out", no ROI at the current frame can be associated with the ROI at the previous frame. The data loss is that fingertips are occluded by other fingers. The process for finding vanishing-pattern(s) is conducted according to the following two steps: (i) if the number of ROIs at Frame t is smaller than the number of ROIs at Frame t-1, Step (ii) is performed; otherwise (larger), it should be treated as an adding-pattern. (ii) For each ROI in each frame, the closest ROI is searched in its previous frame, and if the closest distance is smaller than the threshold *r_Thr*, these two ROIs are associated and treated as active-patterns; otherwise (larger than *r_Thr*), the unassociated ROIs in t and t-1 are treated as adding-patterns and vanishing-patterns, respectively. The adding-patterns are further checked if they (whether the number of ROIs at Frame t-1 is larger than t) correspond to either "frame-out" or "self-occluded". As an example of the vanishing-pattern is shown in the Pattern 3 of Fig.5.6, if the vanishing-pattern(s) keep(s) vanishing long, the pattern(s) are judged to be (a) frame-out finger(s); otherwise, to be a "self-occluded". More specifically, in n frames after frame t, if the fingertip(s) keep(s) vanishing for more than n frames, this module consider it(them) as frame-out finger(s); or after n frames, it(they) return(s) to show in the image sequence, this module consider it(them) as data loss.

5.3.4 ROI associated particle filtering

Non-linear, non-Gaussian particle filtering system can accurately describe multiple fingertip tracking in actual complex guitar playing scenes. In fingertip tracking algorithm, as detailed in Section 5.1, to all the associated ROIs, particles are distributed so as to track the fingertips. The reason why the particle filtering processing is needed even though the ROIs of the fingers are already generated and associated in the previous stage is that the fingertips are not always placed at the center of the ROIs. In the following, this module outlines the state transition and likelihood calculation of this fingertip tracking system; any other details such as particle resampling, are same as traditional work [119].

Considering fast movements of fingers during guitar plays, particles are transited randomly under standard Gaussian Distribution and under no physical movement pattern. The likelihood of all the particles are calculated by using the feature described in Section 4.1; namely, RHT is performed at each particle's position, and that particle's likelihood is given by Eq. (5.2). By calculating the likelihood of each particle distributed only between associated ROIs, four fingertips of the guitarists can be tracked accurately.



1,3 are little finger frame-out and frame-in tracking; 4,5,6,7,9 are jointed fingers issues such as index and middle fingers joint together; 3,8 are fast movement tracking because of blurring; 2,5 are distortion of fingers. Under any circumstances, this work can track the fingertips effectively.

Fig.5.7 Some Examples of Segmentation and Fingertip Tracking Result

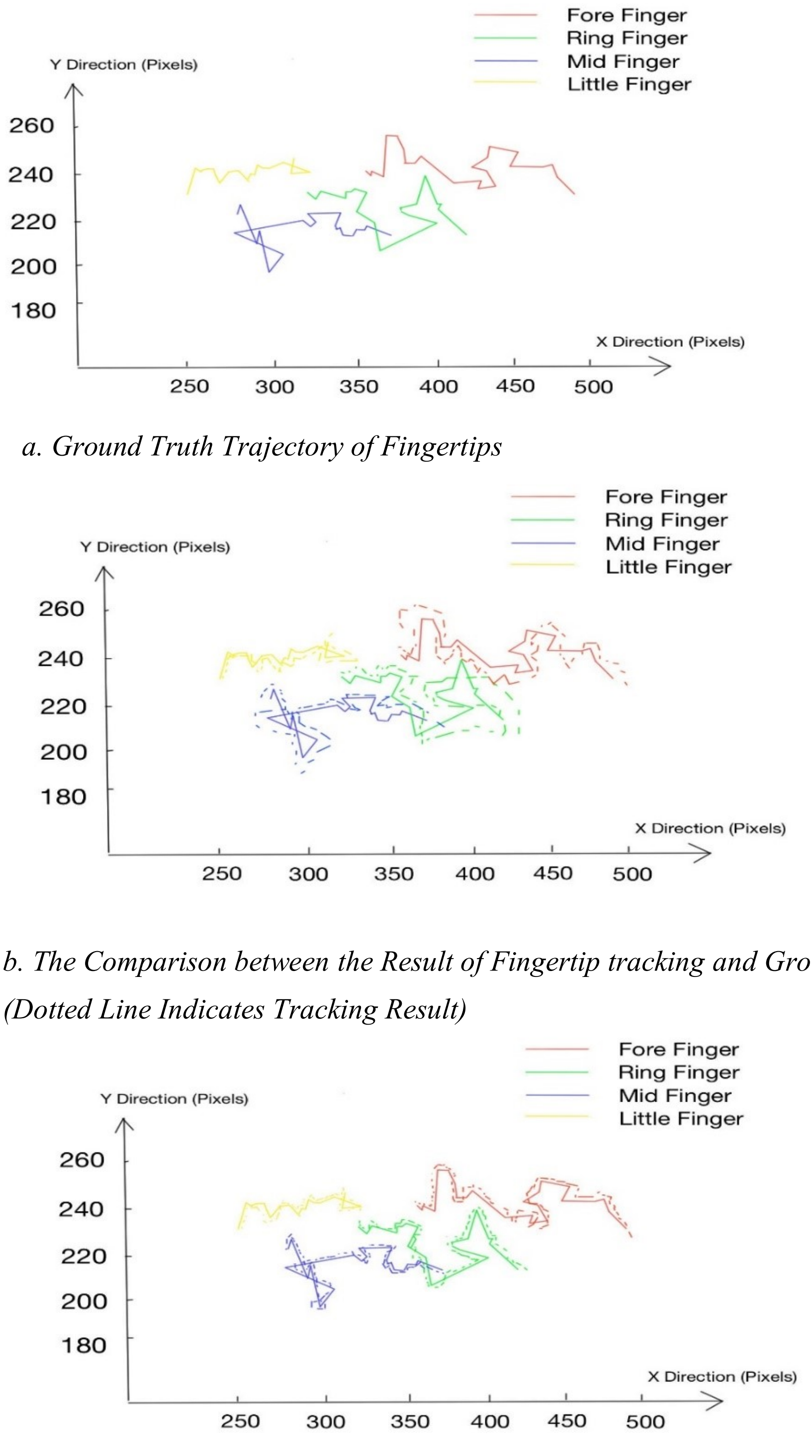
5.4 Experiments

In this experiment, this module uses the guitar playing videos collected from six experimental participants, and the duration of each video is nearly 15 seconds~25 seconds. Totally, 15 videos (7534 frames) are used to test the validity of the proposed algorithm.

All the test data are taken with iPhone front camera with the compressed resolution of 480 pixels by 300 pixels. All the videos were taken under different illumination conditions (daylight, incandescent lamp

Table 5.1 Comparisons of the Proposed Work and Previous Related Works for Fingertips Tracking Errors (Pixel)

Tracking Error (pixels)	Forefinger	Middle finger	Ring finger	Little finger	Mean
Proposed Work (Hough + Template)	**6.5**	**3.3**	**4.9**	**6.0**	**5.2**
Proposed Work (Hough Transform only)	8.8	9.7	9.9	7.8	9.1
Proposed Work (Template Matching Only)	7.5	6.1	7.2	8.4	7.3
The Previous Work [83]	7.3	5.2	6.7	8.3	6.9
Kerdvibulvech [12]	17.1	13.3	10.8	6.7	12.0
A.Burns [10]	150.3	173.5	124	188.3	159.0



a. Ground Truth Trajectory of Fingertips

b. The Comparison between the Result of Fingertip tracking and Ground Truth (Dotted Line Indicates Tracking Result)

c. The Comparison between the Previous Method [83] and Ground Truth (Dotted Line Indicates Tracking Result)

Fig.5.8 The Visual Trajectories Comparison

and etc.) with complicated background and different subjects' clothing. The system used for testing was a Macbook Pro machine with a 2.7 GHz i7 processor and an Nvidia GeForce 650m GT graphics card. The algorithm was implemented in C++, using Xcode 6.4 and version 2.8 of the OpenCV library.

In these 15 test videos, this module let the participants play three musical pieces: (1) Symmetrical Exercise and (2) Scales of 24 majors or Minor (3) a practicing piece of notes because it has been proved these daily guitar exercises help the players to develop their stamina and rhythm abilities, meanwhile developing valuable muscle memory in their picking hand, and no matter novice or professional player yielded significant positive results by practicing these two pieces of notes.

Table 5.1 shows the comparison of fingertip tracking accuracy between the related works and this work. The ground truth is manually given by the observation of human eyes. The proposed method achieves the mean error 6.5, 3.3, 4.9, 6.0 pixels for the four fingertips: fore finger, middle finger ring finger and little finger respectively, totally mean error of 5.2 pixels in *XY* coordination system. This proposed work outperforms the related works [10, 12, 83]. The previous work [83] tracks the fingertips with temporal grouping but without ROI association; A.Burns [10] detected fingertips at each frame without linking or associating correspondent fingertips on consecutive frames, thus the tracking error is fairly high. C.Kerdvibulvech [12] also implements an ROI associated particle filter to track fingertips, and without classifying ROIs to patterns such as the three patterns, it cannot handle self-occlusions or fast-movement, and it is competitive with the work only when no occlusion happens. Details of the comparisons are found in Table 5.1, and some examples of the tracking result are shown in Fig.5.7.

Figure 5.8 shows the vision comparison between the proposed work and the previous methods [83] for the same image sequence with 518 consecutive frames. From Fig. 5.8 it turns out compared with the related work, the tracking result (dotted colorful line) tracks the real fingertips (full line) more accurately.

5.5 Discussion

In this Chapter, the proposed ROI association particle filter-based fingertip tracking method applies a temporal-grouping for the candidates based on ROI (region of interest) association to group the same fingertip candidates on consecutive frames and distribute particles in the surrounding area centered at each of the associated fingertip candidates to address the fast movements and self-occlusions of the fingertips. The contribution of the proposed ROI association particle filter-based fingertip tracking algorithm is summarized as follows:

(1) The proposed method tracks four fingertips in a very high accuracy. Compared with related works [10, 12, 83], four fingertips are tracked with a low tracking error of 5.2 pixel on average. The proposed method even outperforms the deep learning-based states-of-art [21] in the tracking

accuracy of fingertips.

(2) The ROI association-based idea makes the proposed method robust to the joint-finger, self-occlusion, frame-out problems, which are the most difficult issues in fingertip tracking and guitar playing situation.

(3) Compared with other deep learning-based tracking algorithm, the proposed tracking method does not need off-line training process, which causes huge amount of human resources to label the training data; therefore, it is very efficient in processing time and usage of manual labor.

The ROI association particle filter-based fingertip tracking method achieves the above contribution is because of the following reasons:

(1) Two features of fingertips are proposed in this method to accurately fingertip candidates in consecutive frames: reversed Hough Transform and template matching. The reversed Hough Transform is performed to detect the semi-circle shape of fingertips, while six templates are utilized to locate the fingertips of guitarist from different finger directions and finger sizes. Then by utilizing the count map described in Section 5.2, which has the same resolution as the input video, further processes of fingertip tracking in this chapter are easy to conduct to achieve accurate result.

(2) Compared with traditional particle filter that cannot track multiple targets as it randomly distributes particles over the whole image, the proposed method generates a consecutive ROI sequence by proposing a temporal grouping and discriminates different ROIs of the different fingertips by associating the ROIs in consecutive frames. By utilizing the mentioned temporal grouping and ROI association, the proposed method can eliminate noises and solve the self-occlusion of fingertips and joint-finger issues that frequently happen in guitar playing to achieve the accurate result.

However, The ROI associated particle filter-based fingertip tracking algorithm only tracks fingertips in 2D space without depth information. As mentioned in Section 5.4, although the proposed method tracks the fingertips of guitarists more accurately than CNN based algorithm, it lacks the depth information of fingertips, which leads to the problem of inaccuracy for the fingering assessing described in Chapter 7 when compared with 3D CNN based method. Furthermore, because the associated particle filter-based fingertip tracking algorithm only tracks four fingertips of guitarist’s hand, the movement of fingertips cannot fully represent the movement of the hand of the guitarist during guitar playing, which also leads to the problem of inaccuracy for the fingering assessing.

5.6 Conclusion

Chapter 5 proposes a 2D fingertip tracking algorithm within the automatic guitar fingering assessing system. First a machine learning-based Bayesian Pixel Classifier is used to segment the hand area on the test data Then, the probability map of fingertip is generated on segmentation results by counting the voting numbers of the Template Matching and Reversed Hough Transform, and on the probability map, the higher intensity pixels indicate the pixels with higher probability to be the fingertips. Furthermore, a Grouping algorithm, which is a geometry analysis for buffer images, is applied to remove noise and group the same fingertips (index finger, middle finger, ring finger, little finger) at the successive frames, and based on the Grouping results, this module draws the ROIs only at the position of the clustered fingertip candidates. Then, an ROI association algorithm is utilized to associate 4 tracked fingers with their correspondent tracked result frame by frame. This module applies this data association by defining three patterns for fingertip movements during the whole process: the adding pattern, the vanishing pattern, and the active pattern, and every tracked trajectory of fingertip is fitted into these three patterns in order to solve the problem such us self-occlusion and frame-out mentioned in Section 5.1. Finally, particle filter is utilized to track the fingertip of the guitarist in case of complex and volatile hand gesture change by continuously distributing the particles between associated ROIs of fingertips.

Experiments are conducted using videos of guitar plays under different conditions. For the fingertip tracking, the proposed method outperforms the current state-of-art tracking algorithm with high accuracy, the mean error 6.5, 3.3, 4.9, 6.0 pixels for fore finger, middle finger ring finger and little finger respectively.

Chapter 6 Estimating Hand Pose of Guitarist

6.1 Introduction

Accurate hand pose estimation is another task in guitar fingering system as the fingering assessing module needs to combine the spatiotemporal information of both the guitar neck and finger pose. In guitar playing, whether the player is correctly pressed on the fretboard of the guitar or he is just putting the fingertips upon the string without pressing the fretboard is hard to be detected by only using RGB image sequence as the camera is placed in front of the player. Therefore, the 3D hand pose of guitarists is required to detect the "correctly pressing" to achieve a higher assessing result. Besides, not only the fingertips but also the other joints of the hand need to be detected frame by frame. Besides, as mentioned in Section 2.2, traditional methods of hand segmentation based on color [94, 95] or depth value [75] cannot segment hand region as the first step of hand pose estimation module as [1] color and texture vary according to the lighting conditions, presence of shadows, and human hand may appear many shapes depending on the angle of image capture, posture of the hand [74]; Depth cameras bring now an easy solution to handling occlusions, however, they provide a poorly accurate 3D reconstruction of the boundaries of the hand, when users hold something in their hand (in the guitar playing case, guitar players hold guitar at every frame during their playing), as the depth value of the hand and the object are same. [75].

Recently, with the development of neural network, semantic segmentation [96, 97] and human hand pose estimation [13, 15, 18] have been attracting significant attentions of academic research. However, traditional deep learning-based hand pose estimation [13, 15, 18] assume the hand area is the nearest object of the camera and extract the hand by only cropping the pixels of the smallest region with the smallest depth value (they use depth sensor), However, in guitar playing, the guitarist holds the guitar neck during playing, which cause the depth value of the hand and guitar neck are nearly same, and it is impossible to extract hand area in the first step of the approach. Besides, they [13, 15] require a huge training database (over 50,000 images) due to viewpoint quantization and cannot be universally used as it is overfitted to their own training dataset.

To obtain the accurate 3D coordination of all the joints of guitarists during their playing, in Chapter 6, the deep learning-based method for estimating hand pose of the guitarist is proposed. More specifically, first, after inputting a video of guitar play, a CNN-based hand segmentation net is used to discriminate the hand area from the background; second, at every segmented frame of hand region the16 joints of the hand of the guitarist (four joints for index finger, four joints for middle finger, four joints for ring finger and four

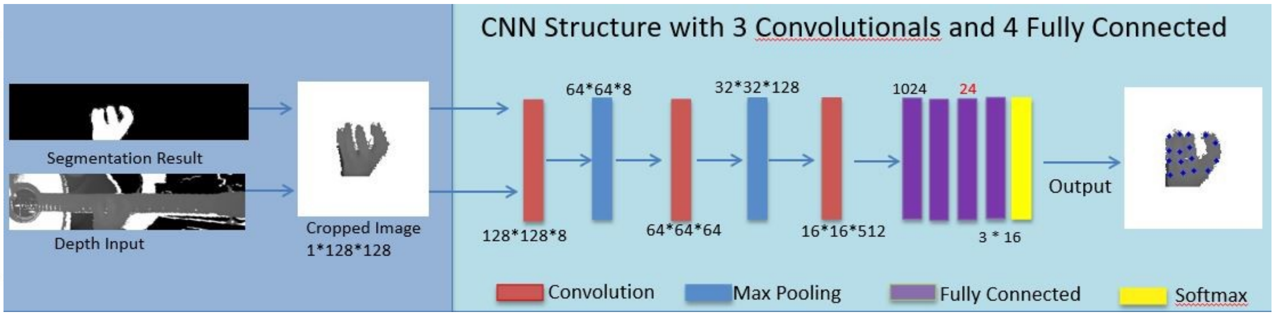


*Fig 6.1***.** *Process Outline of CNN based Hand Segmentation:*

After inputting both RGB image sequence and depth image sequence, hand segmentation based on FCN is proposed to segment hand region on RGB image sequence; then based on the segmented mask, the hand region in depth image sequence is extracted; Furthermore, by cropping the hand region on depth image sequence, the 3D coordination of 16 joints of hand are predicted by the proposed CNN network.

joints for little finger) are estimated by implementing a pretrained neural network model to acquire the 3D coordinates of all 16 joints.

In this chapter, as shown in Fig.6.1, a data-driven deep learning-based framework that outputs the 3D position of joints of the guitarist is proposed. First, after inputting the guitar neck tracking result, which is an image sequence with the projected guitar neck area at the center of each frame, hand area is segmented by using the FCN (Fully Convolutional Neural Network) based hand segmentation algorithm described in Section 6.2. then pre-processing of the module is to extract the hand contours, normalize the hand contour

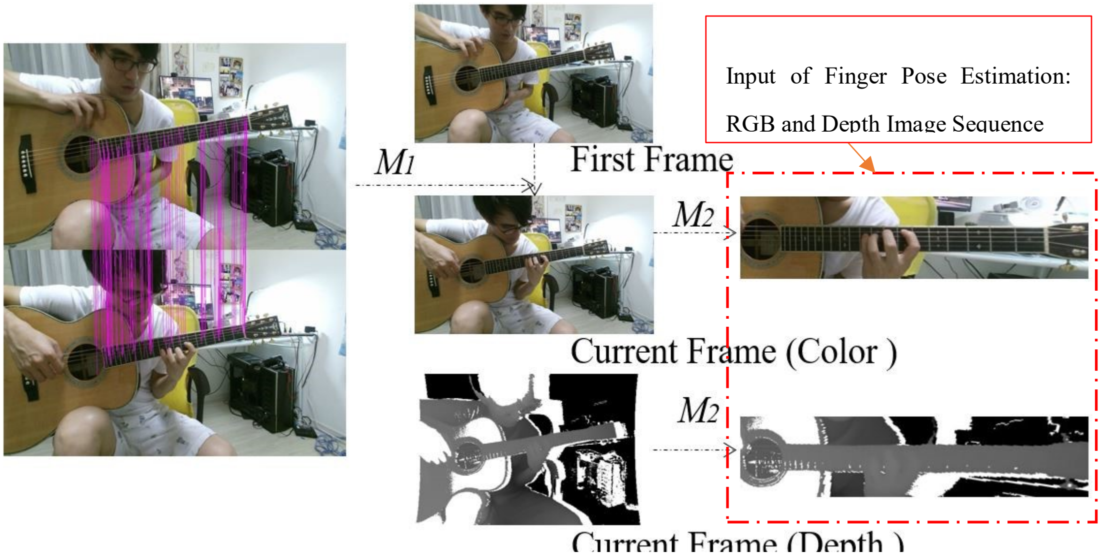
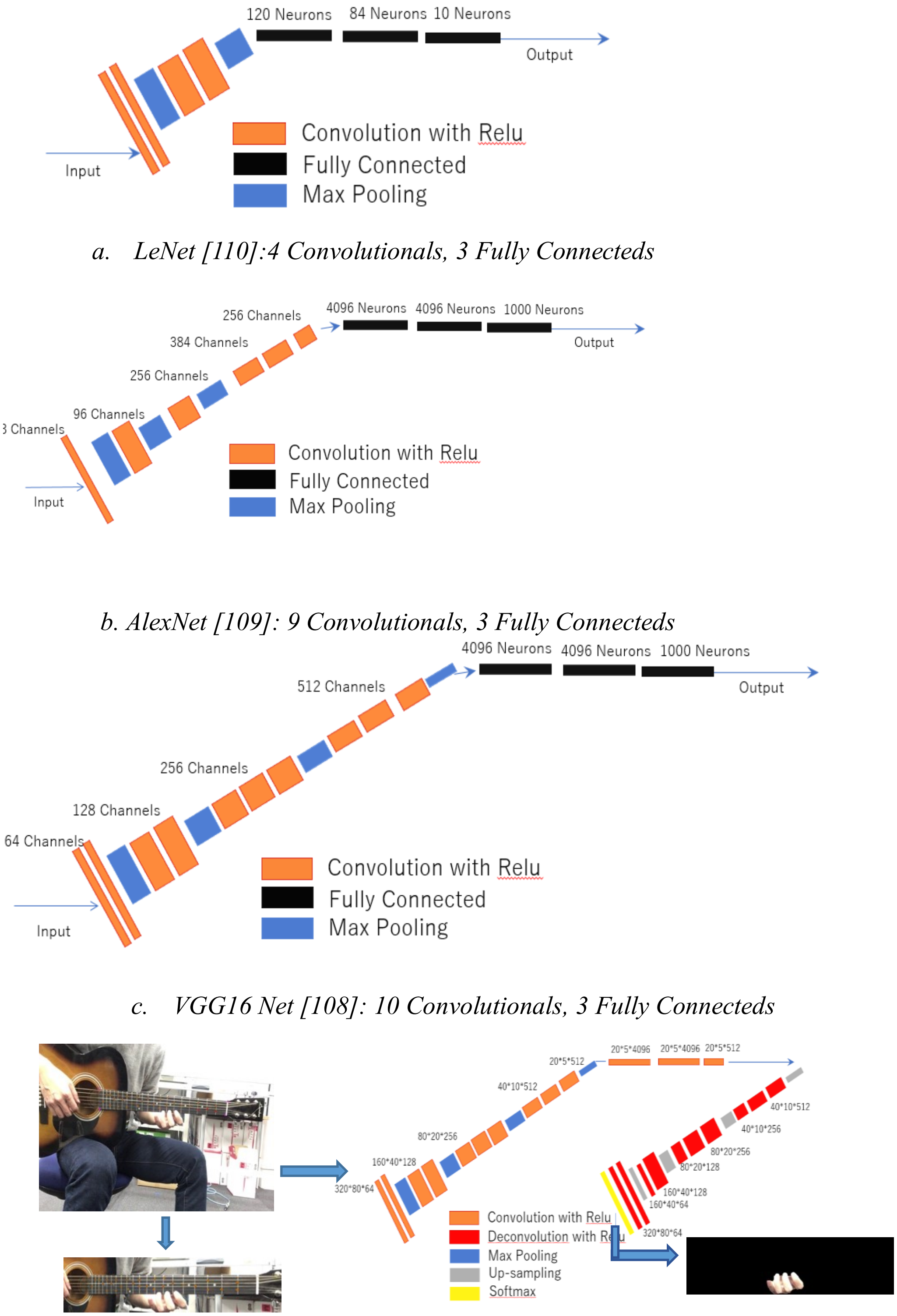

Input of Finger Pose Estimation: RGB and Depth Image Sequence
M1
First Frame
M2
Current Frame (Color)
M2
Current Frame (Depth)

Fig 6.2 Input of Finger Pose Estimation (Also Output of Guitar Neck Tracking)

to a certain resolution image with a certain range of depth value. Furthermore. The CNN (Convolutional Neural Network) based regression model is learned in order to predict the 3D coordinates of the hand of the guitarist.

6.2 Input

As described in Chapter 1, the input of 3D finger pose estimation is the output of the guitar neck tracking, which are two image sequences of RGB image sequence and depth image sequence shown in Fig 6.2. Both of them are the projected result of the original video captured by depth sensor: first M_1 in Fig 6.2 is the result of Modified RANSAC described in Section 3.4 and Eq. (3.10), and M_2 is the result of the projecting the matrix described in Section 3.6 and Eq. (3.12).



a. LeNet [110]:4 Convolutionals, 3 Fully Connecteds

b. AlexNet [109]: 9 Convolutionals, 3 Fully Connecteds

c. VGG16 Net [108]: 10 Convolutionals, 3 Fully Connecteds

d. Hand Net (Proposed Hand Segmentation Network)

Fig 6.3 Net Structure Comparisons with LeNet[110], AlexNet [109]and VGG16 [108]

6.3 Hand Segmentation Based on CNN

Compared with traditional deep learning nets, such as LeNet [110], AlexNet [109], and VGG-16 [108], which must take fixed sized inputs and generate recognition result without spatial coordinate information, this proposed method uses and transfers these fully connected layers into fully convolutional, which could make pixel-wised, no input-size-constrain segmentation happens [97,108]. As shown in Fig 6.3, LeNet [110], AlexNet [109], and VGG-16 [108] use fully connected layers before output; therefore, they lose spatial information on images, and cannot generate the pixel-wise segmentation result.

As a solution, in the proposed hand segmentation algorithm, (1) this module convert the three fully connected layers of VGG-16 [108] to three convolutional layers with 4096, 4096, 512 channels, respectively, as shown in Fig.6.3.d**.** As the fully connected layers of these nets have fixed dimensions and throw away spatial coordinates [96], casting the fully connected layer to the convolutional layer actually can also be viewed as convolutions with kernels that cover the entire input regions [96, 97]. Thus, it can take any input size, max-pool (down-sample) it and recover it to the original size when doing segmentation; (2) instead of implementing linear up-sampling, this module adopt a symmetrical network [96] where the number of deconvolutional layers equals with the number of convolutional layers (both of them are 10 in the case) as shown in Fig.6.3.d. The advantage of the symmetry is the deconvolutional layer could always find its correspondent convolutional layer, so the non-linear up-sampling is performed by memorizing its corresponding max-pooling without learning to linearly up-sample [96,97].

The other detailed information of the net structure is shown in Fig.6.3.d**.** The structure of the net is constructed with two parts: encoder (convolutional layers and max pooling layers) and decoder (up-sampling layers and deconvolutinal layers). In the encoder network, it performs convolution with a filter bank to produce a set of feature maps and then batch normalized [111]. Then an element-wise rectified-linear non-linearity (ReLU) ***max(0, x)*** is applied. Following that, max-pooling with a 2 × 2 window and stride 2 (non-overlapping window) is performed and the resulting output is sub-sampled by a factor of 2. Max-pooling is used to achieve translation invariance over small spatial shifts in the input image; in the decoder network, this module adopts Segnet[20)] up-sampling with deconvolutional layers and softmax layer. Each decoder filter has the same number of channels as the number of up-sampled feature maps. A smaller variant is one where the decoder filters are single channel, i.e. they only convolve their corresponding up-sampled feature map. This variant reduces the number of trainable parameters and inference time significantly [96].

6.4 Hand Region Extraction

The input of the hand pose estimation is the FCN based segmentation result described in Section 6.3. After this module segments the hand region, this module crops the hand region to a 128*128-pixel region like recent deep learning -based approaches [13, 112] do. However, unlike they assume the hand is close to the object, the method uses the FCN-based segmentation described in Section 6.3 to crop the hand region automatically: the method extracts from the depth input and the segmentation mask shown in Fig.6.1, the centroid of the cropped hand region image is also the centroid of the hand. The method implements it by using the segmentation result of FCN-based method (the binarized image), and the method traces the longest contour on the segmentation result to get the hand region area. Furthermore, the method normalizes depth values to [-1,1]: the deeper scene in the cropped image, i.e background is set to 1 (white area in the cropped image of Fig.6.1), and the nearest pixel in the hand region is set to -1.The reason the method does this normalization is that the method needs to assure the work is invariant to (1) the distance between the camera and hand, (2) the position of the hand in the segmentation result. The process is shown in light blue area of Fig.6.1.

6.5 CNN based Hand Pose Estimation

The proposed network is shown in dark region of Fig.6.1. After extracting the hand region to a 128*128 region, first three Convs layers and two max-pooling layers output 512 channels of feature maps; then the method uses two fully-connected layers with 1024 notes respectively; furthermore, instead of directly estimating the 3D position of each joint, the method predicts a lower parameter space because there is a strong relation between each joint of a hand concerning the physical constraint of the hand. Also, related work [13] has shown a low dimensional embedding of hand parameters is sufficient to parameterize the hand's 3D pose. Thus, in this case, the method implements a fully-connected layer with only 24 notes (red number is Fig. 6.1) after the two 1024-note-FC layers; finally, a fully-connected layer with 3*J (J is the number of the joints, in this case, J=16) notes output the 3D position of hand pose.

6.6 Experiments

6.6.1 FCN-Based Hand Segmentation

For training section, this experiment uses 420 images of 13 guitar players' guitar plays. This experiment manually labels the hand area of each image, which is a binary image with the same resolution as the training image. This experiment implements with the Caffe module on a NVIDIA Titan Black GPU with CuDNN acceleration. To train all the variants, this experiment uses stochastic gradient descent (SGD) with a fixed learning rate of 0.1 and momentum of 0.9, and this experiment train the variants until the training loss converges. The training set is randomly shuffled for one epoch and batch size is 12 images.

For test, this experiment uses three commonly used performance measures as follows: (1) global accuracy *(G)* which measures the global percentage of pixels correctly segmented in the dataset; (2) class average accuracy *(C)* is the mean of the segmental accuracy over all classes and (3) mean intersection over union (mIoU) over all classes as used in the related works. The*(mIoU)* metric is a more stringent metric than class average accuracy since it penalizes false positive predictions.

The definition of global accuracy *(G)*, class average accuracy *(C)* and mean intersection over union *(mIoU)* are as follows:

global accuracy *(G):*

$$G = \frac{N_{correct}}{N_{all}} \qquad (6.1)$$

class average accuracy *(C):*

$$C = mean\left(\sum_{i=0}^{P} \frac{N_{correct}}{N_{all}}\right) \qquad (6.2)$$

mean intersection over union *(mIoU):*

$$mIoU = mean\left(\sum_{i=0}^{P} \frac{N_{tpi}}{N_{tpi}+N_{fni}+N_{fpi}}\right) \qquad (6.3)$$

where *N* indicates the number of pixels, *P* is the class number, i is the class index; *tp,fn,fp* indicate true positive, false negative and false positive; $N_{correct}$ indicates the number of the pixels that are correctly segmented, N_{all} indicates the number of the pixels in the dataset, N_{tpi}, N_{fni}, N_{fpi} indicate the number of the truth-positive, false-negative and false-positive segmented pixels respectively.

Table 6.1 Accuracy and Efficiency of CNN-based Segmentation

G is Global Accuracy, ***C*** stands for class average accuracy, ***mIoU*** indicates mean intersection over union in Section 6. ***Tra Time*** is the time cost for training all 420 images, ***Tes Time*** is the time cost for segment one image. ***V1*** is the input version of 320*80 while ***V2*** is 1300*300

	Simple Net		320*80 Net		Hand Net		Related Work[24)
	V1	***V2***	***V1***	***V2***	***V1***	***V2***	
G	0.69	0.91	0.77	0.92	0.88	**0.98**	**0.93**
C	0.62	0.85	0.71	0.88	0.82	**0.95**	**0.87**
mIoU	0.60	0.74	0.69	0.81	0.74	**0.89**	**0.82**
Tra (hour)	13h	18h	15.2h	25h	18.5h	28 h	30.5h
Tes (ms)	21ms	207ms	30ms	275ms	45ms	320ms	400 ms

The whole testing dataset includes 15 videos with 7534 sequential images, and the challenge is to segment hand area from the whole image.

For both training and testing, this experiment uses two versions of input data set, which are totally same except for the resolution: one is 1300*320, the other is 320*80 in order to show how resolution affects the results. Like state-of-arts20,21), this experiment also adopts the three similar networks to evaluate how the deepness of the work affects the segmentation results. This experiment shows the segmentation accuracy and the time efficiency of the three different net structures for both of the two versions of the input data in Table 6.1. The Hand Net (13 convolutional layers) outperforms the 320*80 Net (10 convolutional layers) and the Simple Net (7 Convolutional layers) with 98% accuracy for global accuracy. On the other hand, the data set with the higher resolution (V2 in Table 1) produce the higher test accuracy. This experiment also totally compares the proposed work with state-of-art24), which is the only work using CNN to segment the hand area, which is also the only work achieving above 90% accuracy among all the hand segmentation works. From the Table 6.1, this experiment could say this proposed work outperforms the state-of-arts not only in terms of accuracy, but also time efficiency. The training loss and accuracy information is shown in Fig 6.5. In Fig 6.5, the horizontal axis is the interratation number, while the vertical axis is the training loss (the left vertical axis) and segentation accuracy (the right vertical axis), which indicates that the proposed method converges in an effectively and efficiently manner during the training process of this FCN based method.

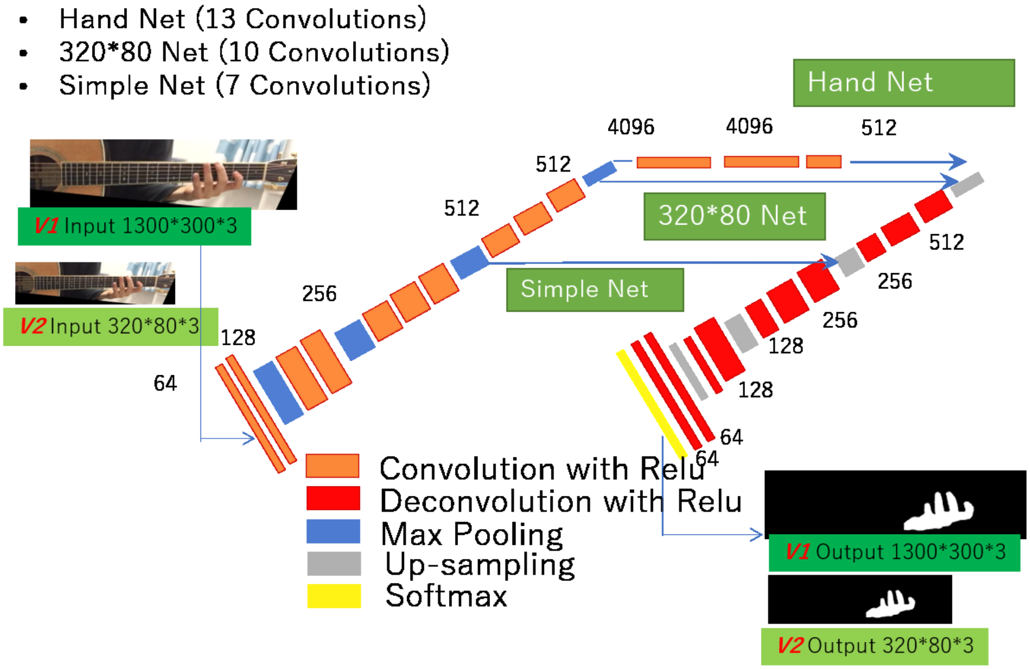


Fig 6.4 Three Network Structure in Experiment

*(1)HandNet with 13 Convolutional layers (Two Versions of Input, V1 is the input resolution of 1300*300, V2 is 320*80) (2) 320*80 Net with 10 Convolutional layers (The Same Two Version of Input Resolution) (3) Simple Net with 7 Convolutional layers (The Same Two Version of Input Resolution)*

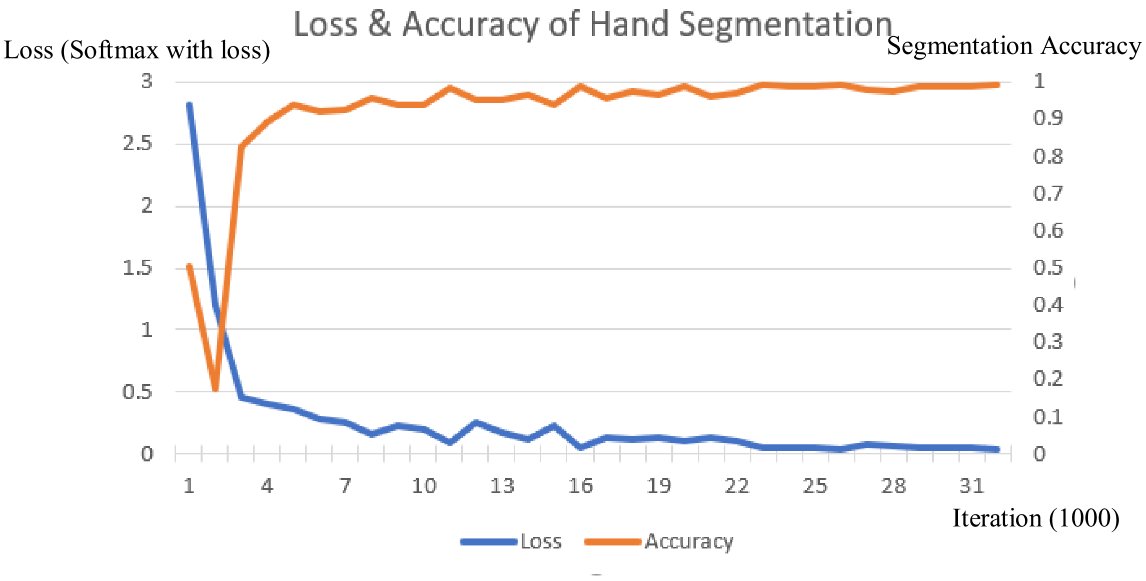
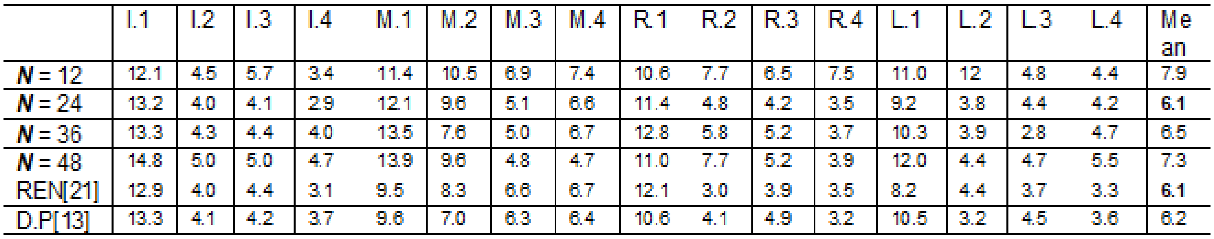


Fig 6.5 Training Loss and Accuracy of Hand Segmentation

Table 6.2. Per-joint mean error of hand pose estimation for Self-Comparison (First Four Rows) and Comparison with State-of-arts (Last Two Rows). I, M, R, L Indicate Index Finger, Middle Finger, Ring Finger and Little Finger; 1, 2, 3, 4 Indicate Finger Joint from Tips to Root.



	I.1	I.2	I.3	I.4	M.1	M.2	M.3	M.4	R.1	R.2	R.3	R.4	L.1	L.2	L.3	L.4	Mean
N = 12	12.1	4.5	5.7	3.4	11.4	10.5	6.9	7.4	10.6	7.7	6.5	7.5	11.0	12	4.8	4.4	7.9
N = 24	13.2	4.0	4.1	2.9	12.1	9.6	5.1	6.6	11.4	4.8	4.2	3.5	9.2	3.8	4.4	4.2	6.1
N = 36	13.3	4.3	4.4	4.0	13.5	7.6	5.0	6.7	12.8	5.8	5.2	3.7	10.3	3.9	2.8	4.7	6.5
N = 48	14.8	5.0	5.0	4.7	13.9	9.6	4.8	4.7	11.0	7.7	5.2	3.9	12.0	4.4	4.7	5.5	7.3
REN[21]	12.9	4.0	4.4	3.1	9.5	8.3	6.6	6.7	12.1	3.0	3.9	3.5	8.2	4.4	3.7	3.3	6.1
D.P[13]	13.3	4.1	4.2	3.7	9.6	7.0	6.3	6.4	10.6	4.1	4.9	3.2	10.5	3.2	4.5	3.6	6.2

6.6.2 CNN-Based Finger Pose Estimation

This experiment applies CNN-based Hand Pose Estimation on 524 images in this thesis's own dataset. As this experiment described before, this experiment first crops the hand region area centered at the centroid of hand, resize it to 128*128 pixels, normalize the depth value to [-1,1]. This experiment implements with the Caffe module on a NVIDIA Titan Black GPU with CuDNN V4 acceleration. This experiment uses stochastic gradient descent (SGD) with a mini-batch size of 12. The learning rate is fixed to 0.0001, and this experiment train the variants until the training loss converges. In the meanwhile, this experiment uses a weight decay of 0.0005 and a momentum of 0.9. this experiment manually labels the position of each 16 joints of cropped hand images to assure the accuracy of the annotation of the training data.

The performance is evaluated by two metrics: (a) per-joint mean error of Euclidean distance (in millimeters) and (b) percentage of frames in which all errors of joints are below a threshold. States-of-arts [13,112,113] all calculate the (b) as this metric is generally regarded very challenging, as a single dislocated joint deteriorates the whole hand pose [13,113]. First, this experiment self-compares the net with baseline and different ensemble settings on the net structure. The self-comparison result based on Metric (a) and (b) are shown in Table 6.2 and Fig.6.6 respectively.

Second, this experiment compares this proposed work with several state-of-the-arts [13,113] methods on the dataset not only in accuracy of estimation (Table 6.2) but also in time efficiency of training and testing (Table 6.3). This work shows a competitive accuracy with state-of-arts [13,113] but outperform them in time efficiency for both training and testing. An example of hand pose estimation result on a video is shown in Fig.6.7. In Fig 6.8, the horizontal axis is the interratation number, while the vertical axis is the 3D Euclid loss of the training and validation (blue curve is for validation, orange curve is for training), which indicates that the proposed method converges in an effectively and efficiently manner during the training process of this CNN based method.

Table 6.3. Comparison of Time Efficiency with Related Works [13, 113].

	Training Time (h)	Testing time per Image (ms) with GPU
This Proposed Work	4	0.19
REN [113]	21	0.84
Deep Prior [13]	16	0.56

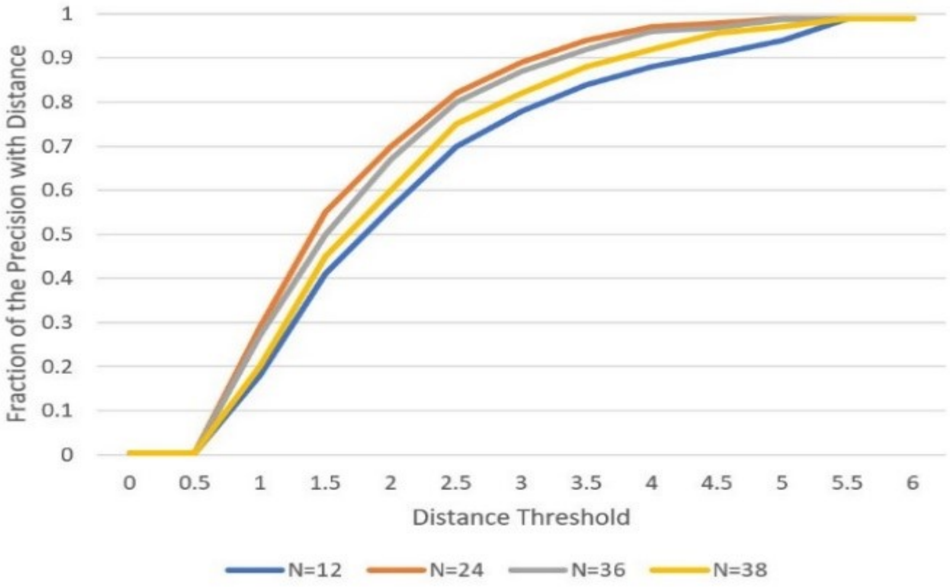
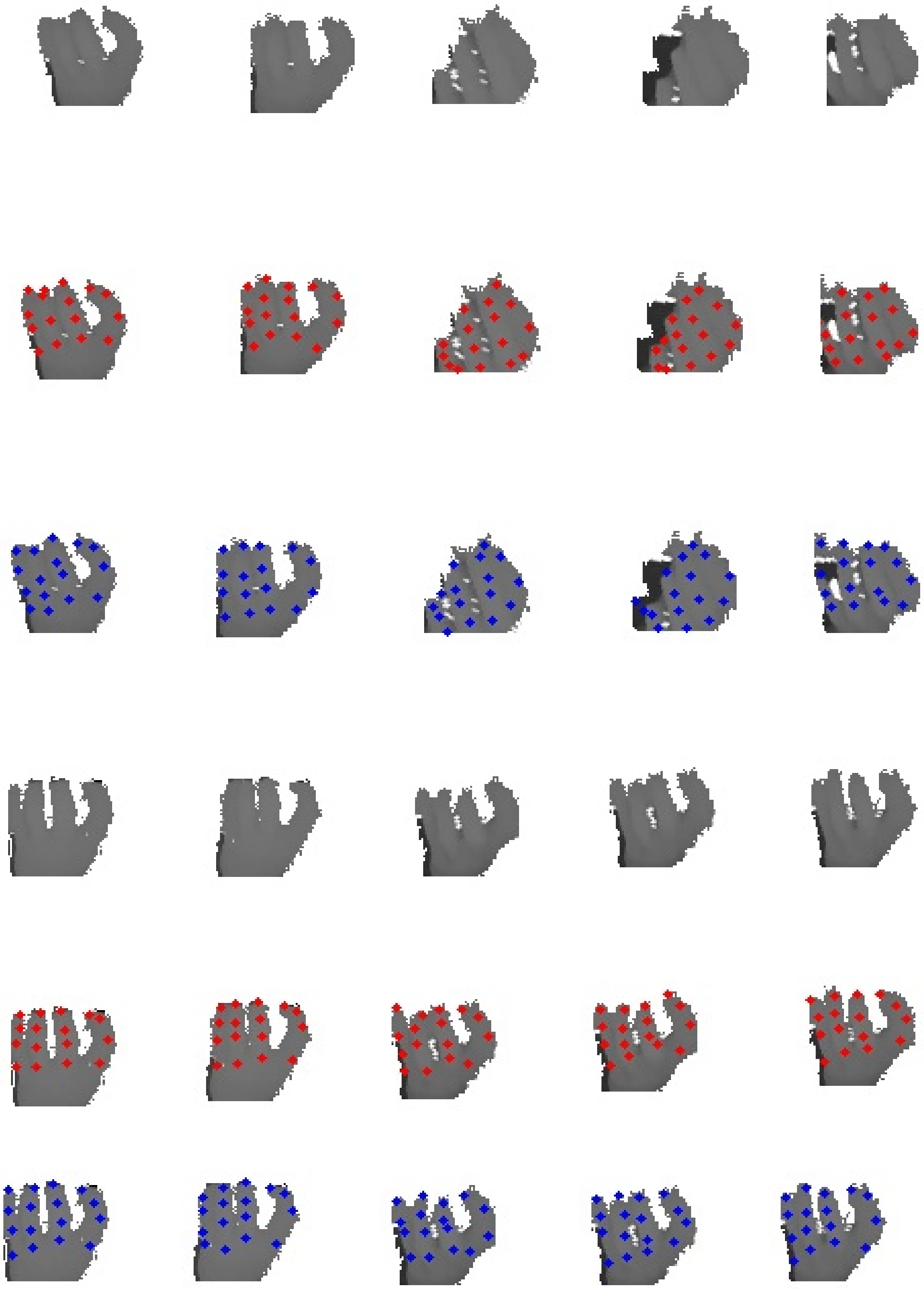


Fig 6.6. Self-Comparison. The horizontal axis indicates the thresholds of mean error of tracking, while the vertical one means the precision of tracking when each threshold on the horizontal axis is set. N=24 achieves best performance within the work.



Red: Ground True *Blue: Prediction*

Fig 6.7 Some Example Result of 3D Finger Pose Estimation

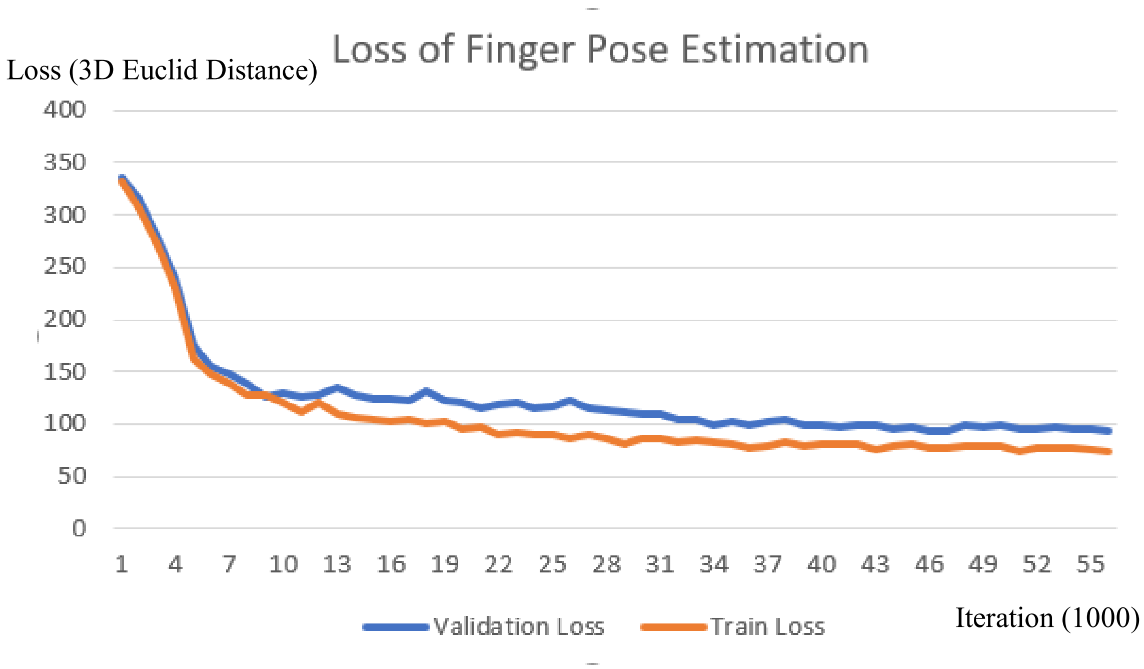


Fig 6.8 Training Loss 7 of Finger Pose Estimation

6.7 Discussion

In this chapter, FCN-based hand area segmentation and CNN-based had pose estimation are proposed to track the hand pose of the guitarist during guitar playing. The contribution of the proposed ROI association particle filter-based fingertip tracking algorithm is summarized as follows:

(1) Because the FCN hand segmentation achieves 98% accuracy for correctly segmenting the hand area in the first step, compared with other deep learning-based methods [13, 15, 19], the proposed method can cope with the situation while the users are holding an object, such as guitar neck in their hand; while the states-of-arts [13, 15, 19] can only estimate the hand pose when users hold nothing in their hand.

(2) The proposed CNN structure adopts a low dimensional embedding of hand parameters. More specifically, the proposed method implements a fully-connected layer with only 24 notes in the middle of all fully-connected layer, which makes the network achieves early same accuracy with states-of-art [21], but highly outperforms it in training and test efficiency.

(3) The proposed CNN based method is also robust to the joint-finger, self-occlusion, frame-out problems, which are the most difficult issues in fingertip tracking and guitar playing situation.

(4) As there are 13 peoples' data in the training dataset, the proposed method is not sensitive to the gender, the personal difference, such as the bone length of the hand; while in the related works, such as hand model-based method [11], it highly depends on the parameters of hand model, which makes difficult to estimate the hand pose for different people.

Compare with ROI association based 2D fingertip tracking described in Chapter 5, although the proposed 3D CNN-based hand pose estimation lower accuracy for four fingertip tracking, it tracks 16 joints of the guitarist in 3D frame by frame. The merit of 3D tracking in guitar playing is 3D finger pose estimation can comprehensively represent the movement of the hand of the guitarist during guitar playing, such as whether the guitarist presses on the fretboard or not by acquiring the depth information; furthermore, hand pose consisting of 16 joints can also comprehensively represent the movement of the hand of the guitarist during guitar playing compared with only four joints (fingertips) are tracked in 2D ROI association-based tracking method.

6.8 Conclusion

In this chapter, two deep learning-based methods are proposed to output the 3D position of the hand pose of the guitarist: first, an FCN based network, which is constructed by converting the three fully connected layers of VGG-16 to three convolutional layers with 4096, 4096, 512 channels respectively, is trained to output the segmentation result of the hand area. With a correct segmentation rate of 0.98 for global accuracy, the proposed network outperforms state-of-art significantly. Furthermore, a CNN based network, which is constructed of 3 convolutional layers and 3 fully-connected layers works almost equally accurate as the state-of-arts, while outperforms them in better training and testing time efficiency. More specifically, instead of directly regressing and estimating the 3D position of 16 joints of the guitarist's hand, the proposed work predicts a lower parameter space as there is a strong relation between each joint of a hand concerning the physical constraint of hand.

Chapter 7 Guitar Fingering Assessing

7.1 Introduction

As mentioned in Chapter 1, fingering assessing of guitar plays involves not only detecting finger pose at some specific timing; more importantly the transition of finger movement at every single timing during guitar playing needs to be evaluated.

As mentioned in Section 2, traditional research works [89, 90, 91] assess human actions by manually designing evaluation functions, for instance trajectory of the player of gymnast [89] can be used to evaluate the performance of one specific gymnast action. As each action needs a unique evaluation function, obviously the human design function cannot be generalized into other actions.

As one of the most important originalities of this thesis, in this section, the algorithm of assessing the transition of finger movement is proposed. More specifically, the spatio-temporal feature that can represent the finger movement in guitar playing video is proposed, and by fitting the feature to a linear SVR (support vector regression) model, the module outputs the automatically predicted score for assessing the transition of the finger movement in guitar playing video.

First, the input of the fingering assessing from transitions of finger movements is the result of the 3D hand pose estimation on each frame of guitar playing video described in Section 7.2; then by calculating the proposed 3D DCT (Discrete Cosine Transform) as the feature of the hand pose on consecutive frames, which can be used to represent the whole transition movement of the guitarist's hand; furthermore, the training-based regression model is utilized to assess fingering from the transition of finger movement. Note that, the process of finger movement assessing does not use any human designed evaluation function. The whole process is shown in Fig. 7.1.

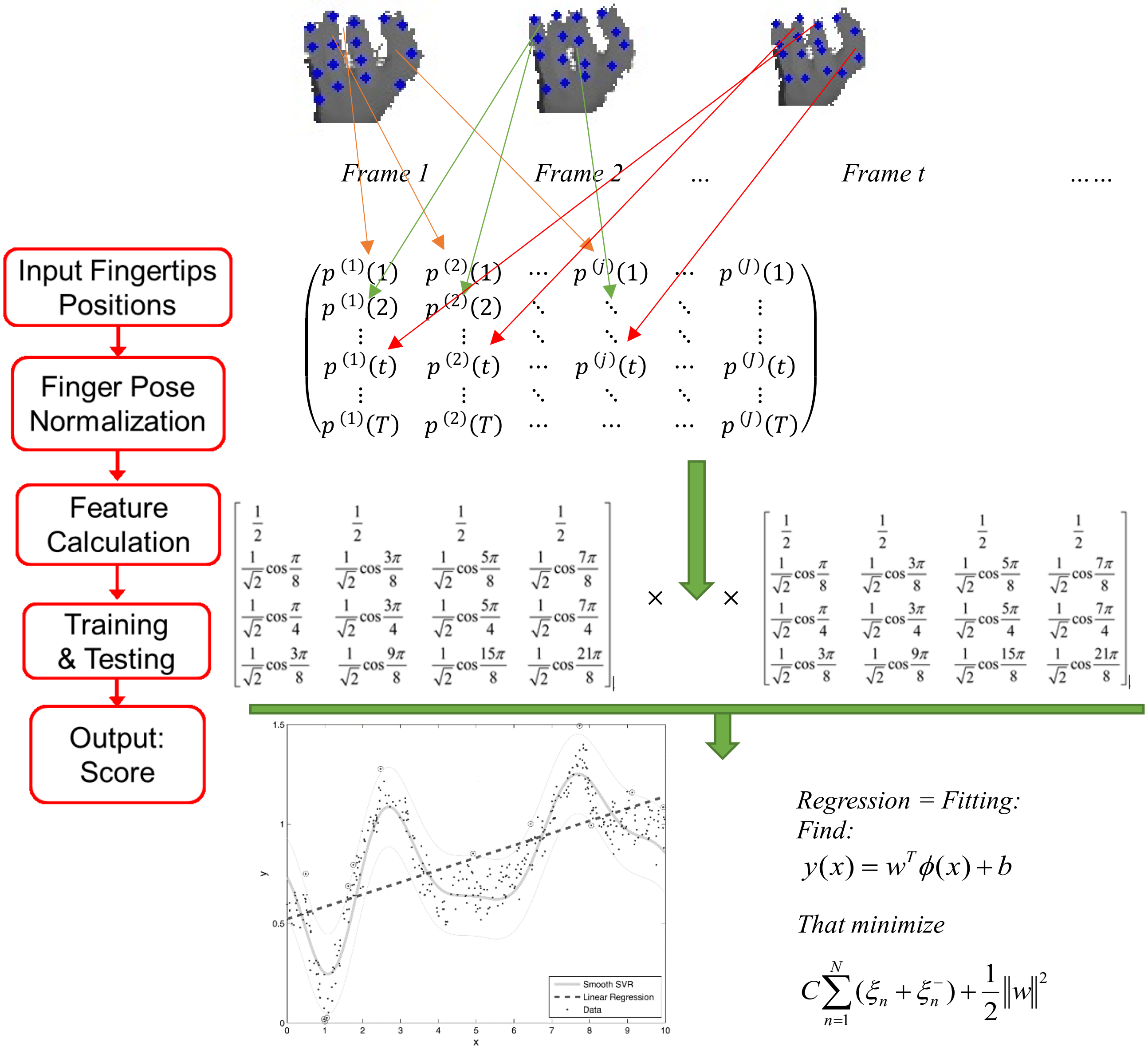

Input Fingertips Positions
Finger Pose Normalization
Feature Calculation
Training & Testing
Output: Score
Frame 1
Frame 2
...
Frame t
……
$\begin{pmatrix} p^{(1)}(1) & p^{(2)}(1) & \cdots & p^{(j)}(1) & \cdots & p^{(J)}(1) \\ p^{(1)}(2) & p^{(2)}(2) & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ p^{(1)}(t) & p^{(2)}(t) & \cdots & p^{(j)}(t) & \cdots & p^{(J)}(t) \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ p^{(1)}(T) & p^{(2)}(T) & \cdots & \cdots & \cdots & p^{(J)}(T) \end{pmatrix}$
$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{\sqrt{2}}\cos\frac{\pi}{8} & \frac{1}{\sqrt{2}}\cos\frac{3\pi}{8} & \frac{1}{\sqrt{2}}\cos\frac{5\pi}{8} & \frac{1}{\sqrt{2}}\cos\frac{7\pi}{8} \\ \frac{1}{\sqrt{2}}\cos\frac{\pi}{4} & \frac{1}{\sqrt{2}}\cos\frac{3\pi}{4} & \frac{1}{\sqrt{2}}\cos\frac{5\pi}{4} & \frac{1}{\sqrt{2}}\cos\frac{7\pi}{4} \\ \frac{1}{\sqrt{2}}\cos\frac{3\pi}{8} & \frac{1}{\sqrt{2}}\cos\frac{9\pi}{8} & \frac{1}{\sqrt{2}}\cos\frac{15\pi}{8} & \frac{1}{\sqrt{2}}\cos\frac{21\pi}{8} \end{bmatrix}$
×
×
$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{\sqrt{2}}\cos\frac{\pi}{8} & \frac{1}{\sqrt{2}}\cos\frac{3\pi}{8} & \frac{1}{\sqrt{2}}\cos\frac{5\pi}{8} & \frac{1}{\sqrt{2}}\cos\frac{7\pi}{8} \\ \frac{1}{\sqrt{2}}\cos\frac{\pi}{4} & \frac{1}{\sqrt{2}}\cos\frac{3\pi}{4} & \frac{1}{\sqrt{2}}\cos\frac{5\pi}{4} & \frac{1}{\sqrt{2}}\cos\frac{7\pi}{4} \\ \frac{1}{\sqrt{2}}\cos\frac{3\pi}{8} & \frac{1}{\sqrt{2}}\cos\frac{9\pi}{8} & \frac{1}{\sqrt{2}}\cos\frac{15\pi}{8} & \frac{1}{\sqrt{2}}\cos\frac{21\pi}{8} \end{bmatrix}$
1.5
1
0.5
0
y
0
1
2
3
4
5
6
7
8
9
10
x
Smooth SVR
Linear Regression
Data
Regression = Fitting:
Find:
$y(x) = w^T \phi(x) + b$
That minimize
$C\sum_{n=1}^{N}(\xi_n + \xi_n^-) + \frac{1}{2}\|w\|^2$

Fig 7.1 Outline of Chapter 7

7.2 Input of Fingering Assessing

After estimating the hand pose of the guitarist frame by frame, this module formulates the hand pose of the guitarist in single video as a two-dimensional matrix $P'^{(j)}(i)$:

$$
P'^{(j)}(i) \\
= \begin{pmatrix}
x_u^{(1)}(1) & y_u^{(1)}(1) & z_u^{(1)}(1) & x_u^{(2)}(1) & y_u^{(2)}(1) & z_u^{(2)}(1) & \cdots & x_u^{(J)}(1) & y_u^{(J)}(1) & z_u^{(J)}(1) \\
x_u^{(1)}(2) & y_u^{(1)}(2) & z_u^{(1)}(2) & x_u^{(2)}(2) & y_u^{(2)}(2) & z_u^{(2)}(2) & \ddots & & \vdots & \\
 & & & \vdots & & & \ddots & & \vdots & \\
x_u^{(1)}(i) & y_u^{(1)}(i) & z_u^{(1)}(i) & x_u^{(2)}(i) & y_u^{(2)}(i) & z_u^{(2)}(i) & \ddots & x_u^{(J)}(i) & y_u^{(J)}(i) & z_u^{(J)}(i) \\
 & & & \vdots & & & \ddots & & \vdots & \\
x_u^{(1)}(I) & y_u^{(1)}(I) & z_u^{(1)}(I) & x_u^{(2)}(I) & y_u^{(2)}(i) & z_u^{(2)}(I) & \cdots & x_u^{(J)}(I) & y_u^{(J)}(I) & z_u^{(J)}(I)
\end{pmatrix}
$$

$$
j \in (1,2 \ldots J = 16),\ i \in (1,2 \ldots Index = 16) \qquad (7.1)
$$

where $P'^{(j)}(i)$ is a $3J * Index$ matrix, J is the total number of the finger joint, $Index$ is the total frame number of the video. Each set of $x_u^{(j)}(i)$ $y_u^{(j)}(i)$ $z_u^{(j)}(i)$ indicates the three-dimensional coordination of j th joint of finger at *Frame* i in $X_uY_uZ_u$ coordination system.

By replacing a set of $x_u^{(j)}(i)$ $y_u^{(j)}(i)$ $z_u^{(j)}(i)$ as $p^{(j)}(i)$, $P^{(j)}(i)$ is:

$$
P^{(j)}(t) = \{ p^{(j)}(t) \} = \begin{pmatrix}
p^{(1)}(1) & p^{(2)}(1) & \cdots & p^{(j)}(1) & \cdots & p^{(J)}(1) \\
p^{(1)}(2) & p^{(2)}(2) & \ddots & \ddots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
p^{(1)}(i) & p^{(2)}(i) & \cdots & p^{(j)}(i) & \cdots & p^{(j)}(i) \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
p^{(1)}(Index) & p^{(2)}(Index) & \cdots & \cdots & \cdots & p^{(J)}(Index)
\end{pmatrix}
$$

$$
j \in (1,2 \ldots J = 16),\ i \in (1,2 \ldots Index = 16) \qquad (7.2)
$$

where $P^{(j)}(t)$ is a $3J * Index$ matrix, J is the total number of the finger joint, $Index$ is the total frame number of the video. Each set of $P^{(j)}(t) = (\, x_u^{(j)}(i), y_u^{(j)}(i),\ z_u^{(j)}(i))$ indicates the three-dimensional coordination of j th joint of finger at Frame i.

7.3 Spatio-Temporal Feature Calculation

Equation (7.2) lists all the 3D coordinates of the 16 finger joints of any single input video. Therefore, the movement of all finger joints in a video can be represented by a matrix like Eq. (7.2). However, there are two drawbacks of the representative form in Eq. (7.2) of the movement of the fingers:

(1) As the movement of fingers in a video is actually an analog signal, in Eq. (7.2) the matrix is a digital discrete signal sampled by camera FPS and etc.;

(2) The 3D value of the coordinates of fingers in Eq. (7.2) is the predicted result of 3D hand pose estimation in Chapter 6. Therefore, it is not the perfect ground truth of finger pose but estimated result of the algorithm (it contains error of estimation mentioned in Chapter 6).

Based on these two reasons, the thesis proposes a feature calculation process that calculates a more accurate representative form for the transition of finger movement in guitar fingering assessing.

7.3.1 Discrete Fourier Transform & Discrete Cosine Transform

The Discrete Fourier Transform (DFT) is one of the most powerful tools in digital signal processing; it enables us to find the spectrum of a finite-duration signal. The mathmatical process of DFT is shown as follows:

Let $f(i)$ be the continuous signal that can be considers as original data (ground truth). In digital signal processing, first this module samples the original data by using N datas denoted as:

$$f[0], f[1], f[2] \ldots\ldots f[n] \ldots\ldots f[N-1] \tag{7.3}$$

The Fourier transform of the original data $f(i)$ is:

$$F(j\omega) = \int_{-\infty}^{\infty} f(i)\, e^{-j\omega i} di \tag{7.4}$$

By applying in each sampled data $f[n]$ since the integrant exists only at the sampled point:

$$F(j\omega) = \int_{0}^{(N-1)Index} f(i)\, e^{-j\omega i} di$$

$$= f[0]e^{-j0} + f[1]e^{-j\omega Index} + f[2]e^{-j\omega 2Index} + \cdots + f[n]e^{-j\omega nIndex} + \cdots + f[N-1]\, e^{-j\omega(N-1)Index}$$

$$ie.\quad F(j\omega) = \textstyle\sum_{0}^{N-1} f[n]e^{-j\omega nIndex} \tag{7.5}$$

As any sample signal with N sampled data can be interpreted as a periodic signal by assuming $f[N]$ to $f[2N-1]$ is as same as $f[0]$ to $f[N-1]$, by setting the basis frequency component as $\frac{1}{NT}\, Hz$ or

$\frac{2\pi}{NT}\ rad/sec.$

By setting $\omega = 0, \frac{2\pi}{NT}, \frac{2\pi}{NT} \times 2, \ldots . \frac{2\pi}{NT} \times n \ldots . \frac{2\pi}{NT} \times (N-1)$, the $F[n]$ (DFT result of the sampled signal $f[n]$ is:

$$F[n] = \sum_0^{N-1} f[n] e^{-j\frac{2\pi}{N}n} \qquad n = 0{:}N-1 \tag{7.6}$$

Re-write the Eq. (7.4) in matrix form:

$$\begin{pmatrix} F[0] \\ F[1] \\ F[2] \\ \vdots \\ F[N-1] \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & W & W^2 & W^3 & \cdots & W^{N-1} \\ 1 & W^2 & W^4 & W^6 & \cdots & W^{N-2} \\ 1 & W^3 & W^6 & W^9 & \cdots & W^{N-3} \\ \vdots & & & & \cdots & \vdots \\ 1 & W^{N-1} & W^{N-2} & W^{N-3} & \cdots & W \end{pmatrix} \begin{pmatrix} f[0] \\ f[1] \\ f[2] \\ \vdots \\ f[N-1] \end{pmatrix} \tag{7.7}$$

where, $W = \exp(-j\frac{2\pi}{N})$.

For two dimensional signals (M*N), such as the finger pose matrix shown in Section 7.2.1, the 2D DFT is defined as follow:

$$F[k,l] = \frac{1}{\sqrt{MN}} \sum_0^{N-1} [\sum_0^{M-1} f[m,n]\, e^{-j2\pi\frac{mk}{M}}\ ]\, e^{-j2\pi\frac{nl}{N}} =$$

$$\frac{1}{\sqrt{N}} \sum_0^{N-1} X'[k,n]\, e^{-j2\pi\frac{nl}{N}} \qquad k = 0{:}M-1; l\ = 0{:}N-1 \tag{7.8}$$

where $X'[k,n] = \frac{1}{\sqrt{M}} \sum_0^{M-1} X[m,n]\, e^{-j2\pi\frac{mk}{M}}$

Eq. (7.8) shows that the two-dimensional DFT process can be considered as a two-step one dimensional DFT: first the $X'[k,n]$ is the one-dimensional DFT result of row signal M in n th colomn:

$$X'_n = W_M^* x_n, \qquad n = 0{:}N-1;$$

where W_M^* is the FFT matrix shown in Eq. (7.7) with *N*N* members.

By calculating all the N columns of DFT:

$$X' \quad = \left[{X'}_0, {X'}_{1,} {X'}_2 \ldots {X'}_{N-1}\right] = W_M^* \left[{x'}_0, {x'}_{1,} {x'}_2 \ldots {x'}_{N-1}\right] = \; W_M^* \, x \tag{7.9}$$

On the other hand, the one -dimensional DFT for the colomn signal N while the row vector is considered as parameter. Therefore, the second step of 2D DFT is a one-dimensional DFT of the k th row:

$$X_k^T = (W_N^* {X'}_k)^T = {X'}^T{}_k \, W_N^{*T} = {X'}^T{}_k \, W_N^* \qquad k = 0{:}\, M-1 \tag{7.10}$$

By calculating every row:

$$\begin{pmatrix} X_0^T \\ X_1^T \\ X_2^T \\ \vdots \\ X_{N-1}^T \end{pmatrix} = \begin{pmatrix} X_0'^T \\ X_1'^T \\ X_2'^T \\ \vdots \\ X'^{\,T}_{N-1} \end{pmatrix} \; W_N^* \tag{7.11}$$

$$X = X' \, W_N^* \tag{7.12}$$

Since $X' = \; W_M^* \, x$ in Eq. (7.11),

The 2D DFT can be written as:

$$X = W_M^* \, x \, W_N^* \tag{7.13}$$

Similarly, the coefficient of 2D DCT (Discrete Cosine Transform) matrix (N*N) is defined as:

$$C_{k,l} = \begin{cases} \frac{1}{\sqrt{N}} & if\ k = 0; \\ \frac{\sqrt{2}}{\sqrt{N}} \cos[\frac{(2l+1)k\pi}{2N}] & if\ k > 0; \end{cases} \tag{7.14}$$

The 2D DCT can be written as:

$$X = C_M^* \, x \, C_N^*$$

7.3.2 Spatio-temporal Feature of 3D Finger Pose

In the guitar playing, after estimating the hand pose of the guitarist frame by frame, this module formulates the hand pose as:

$$q^{(j)}(t) = \; p^{(j)}(t) - p^{(0)}(t) \;\; j \in (1,2 \ldots J) \tag{7.15}$$

where J is the total number of the finger joint, t is the frame index of the video, $p^{(j)}(t)$ is the index finger's tip at frame t. The reason this module does this normalization is to let the feature be invariant to any input. Then the module transforms the normalized position of joints from spatial domain to frequency domain by using DCT:

$$Q^{(j)} = \; C_M^* \, q^{(j)}(t) \, C_N^* \tag{7.16}$$

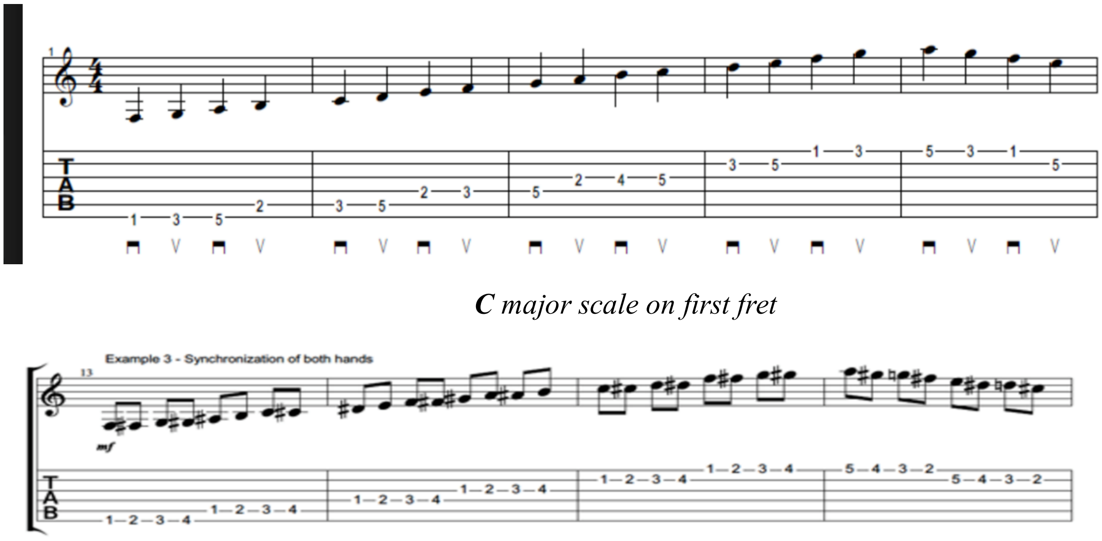
where M is number of total rows of $q^{(j)}$(t), N is number of total columns of $q^{(j)}$(t). The module computes the features for every joint for 3D (x_u, y_u, z_u) component respectively and concatenate them all to final feature vector:

$$A = |Q| \tag{7.17}$$

where A is absolute value of the result of cosine transform, which is the final features of finger pose.

7.4 Support Vector Regression

This module utilizes a supervised regression SVR model for the finger pose assessing system. Note feature vector $A = |Q|$ is the frequency feature of hand pose, and each video of guitar playing is a single feature vector which horizontal component is the joint index of the guitarist, while vertical one is the first k component of cosine transform. The ground truth of the scoring of each video is obtained by specialist of guitar playing. The module uses a linear support vector regression (L-SVR) to predict the score of guitar playing by using lib-SVM [79, 80]. Any details of SVR could be found at [81].



C major scale on first fret

Symmetrical Excise

Fig 7.2. 2 music pieces in the guitar fingering assessing module: (1) ***C*** *major scale on first fret and (2) symmetrical excise*

7.5 Experiments

7.5.1 Experimental Object and Condition

In the fingering assessment evaluation, similar but not exactly same as Section 5.4, the experiment chooses two kinds of music pieces: (1) ***C*** major scale on the first fret and (2) symmetrical exercise as the object of fingering assessment. The musical score of two exercises are shown in Fig.7.2. Each music piece contains 50 videos taken by Microsoft Kinect, and all the 99 videos are taken by 11 guitar players including 3 expert players, 4 mid-level players and 4 beginners. Each video is scored ranging from 0 to 100 (full mark) by 3 experts of guitar playing. The experiment cross validates the work by randomly selecting 40 videos as training, 10 as testing for 5 times.

7.5.2 Mean Rank Correlation of Self-Comparison and Comparison with States-of-arts

The experiment self-compares this proposed work by: (1) the experiment utilizes three features: a. 3D-DCT as described in Eq. (7.17) , b. 3D DFT as described in Eq.(7.13), c.3D Space-time interest points (STIP) as described in Eq.(7.1) d.2D-DCT as the result of the fingertip tracking result described in Chapter 5, which is the tracking result of 2D fingertip tracking; (2) the experiment compare the ridge regression with this proposed SVR with three features mentioned in (1) respectively [79]. Table 7.1 and 7.2 show the detailed comparison results on C major scale and symmetrical exercise, respectively. The experiment uses mean rank correlation to evaluate the proposed work like states-of-arts did [15]. The mean rank correlation is defined as follows:

$$\rho = 1 - \frac{6\sum_{1}^{N}(x_i - y_i)^2}{n(n^2-1)} \tag{7.18}$$

where *N* is total number of the testing data, x_i is the regression score of *i*-th data, y_i is the ground truth score of *i*-th data. The proposed method (3D-DCT+SVR) outperform others with a mean rank correlation of 0.68 for C major scale and symmetrical exercise (0.67 in Table 7.1 and 0.69 in Table 7.2).

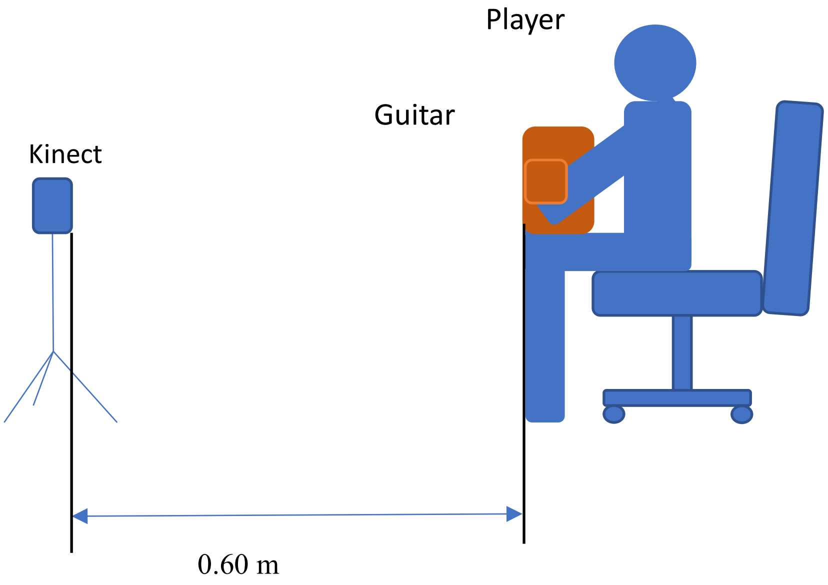


Fig 7.3. Experimental Conditions of Guitar Fingering Assessing

a. *Three Expert Players*

b. *Four Mid-Level Players*

c. *Four Beginners of Guitar*

Fig 7.4 100 Experiment Videos Taken by 11 Players

Table 7.1. Mean Rank Correlation of Fingering Assessing for C Major Scale (the first row is the mean ran correlation value of the proposed SVR; while the second row is the mean ran correlation value of Ridge Regression. The different columns indicate the value of mean ran correlation for each different features: DCT, DFT etc.)

	3D-STIP	3D-DCT	3D-DFT	AQA[79]	2D-DCT
SVR	0.33	**0.67**	0.45	0.34	0.41
Ridge Regression	0.29	0.45	0.38	0.27	0.3

Table 7.2. Mean Rank Correlation of Fingering Assessing for Symmetrical Excise

	3D-STIP	3D-DCT	3D-DFT	AQA[15]	2D-DCT
SVR	0.31	**0.69**	0.57	0.49	0.43
Ridge Regression	0.35	0.48	0.49	0.44	0.22

In the meantime, the experiment compares the proposed fingering assessment with the AQA [79], which is the only work for automatically evaluating human action on 2D video. From Table 7.1 and 7.2, it figures out the proposed method outperform it with the mean rank correlation 0.68 (the mean rank correlation of AQA [79] is 0.415 tested on the dataset of C major scale and symmetrical exercise).

7.5.3 T-Test Validation

T-test is one of the most frequently used procedures in statistics to determine whether the mean of a population significantly differs from a specific value (called the hypothesized mean) or from the mean of another population. T value of two samples is calculated by Eq. (7.19) [114]:

$$T_t = \frac{\overline{Pro} - \overline{Oth}}{s_{\mathrm{p}}\sqrt{\frac{2}{n}}}$$

$$\text{while } s_{\mathrm{p}} = \sqrt{\frac{s_{Pro}^2 + s_{Oth}^2}{2}} \tag{7.19}$$

where n is the number of test data, Pro is the prediction sequence value of all data by using 3D-DCT as the feature and SVR as the regression model (the proposed method), Oth is the prediction sequence value of all data by using other methods in Table 7.1 and Table 7.2 (3D-STIP, 3D-DFT and AQA [15]); s_p is the pooled standard deviation for $n = n_1 = n_2$ and s_{Pro}^2 and s_{Oth}^2 are the unbiased estimators of the variance

Table 7.3. Comparison between The System Prediction and mid-level / beginner human players

	Score	Difference	Mean Error	Variance
Ground Truth	50	0	0	0
Prediction of The System	43.8.	6.2	6.3/100	54.3
Prediction of Mid-level Player	38.7	11.3	11.4/100	501
Prediction of Beginners	31.1	18.9	19.1/100	1323

of the two samples [114].

Based on the Eq. (7.19), the p value of the T-test between 3D-DCT and 3D-STIP is **0.029** at statistical significant at an alpha level of 0.05; the p value of the T-test between 3D-DCT and 3D-DFT is **0.0095** at statistical significant at an alpha level of 0.05; the p value of the T-test between 3D-DCT and AQA [15] is **0.01598** at statistical significant at an alpha level of 0.05. It is proved that the experiments between the proposed method and other methods are statistically effective with the significances all smaller than **0.05**.

7.5.4 Comparison with Human Judge

As mentioned in Chapter 1, the whole research goal of this thesis is that, the score prediction of the guitar fingering assessing should be at least better than mid-level human player. As the ground truth data is manually labelled by expert players in this module, the comparison between the system prediction and the prediction of mid-level / beginner human players are shown in Table 7.3.

The Difference *(D),* Mean Error *(ME)* and Variance *(V*) is Table 7.3 is calculated in Eq. (7.20), Eq. (7.21) and Eq. (7.22) respectively:

Difference *(D):*

$$D = \overline{Pre} - \overline{Lab} \tag{7.20}$$

Mean Error *(ME):*

$$ME = \frac{\sum_{i=0}^{N} |Pre_i - Lab_i|}{N} \tag{7.21}$$

Variance *(V):*

$$V = \frac{\sum_{i=0}^{N} (Pre_i - Lab_i)^2}{N} \tag{7.22}$$

where, $\overline{Pre}$ is the mean value of all the system prediction value, $\overline{Lab}$ is the mean value of all the label value; *N* is the total data number in the dataset.

In Table 7.3, the row of ground truth (first row) shows the score used in training data labelled by expert players of guitar. From the second row to the last row, it shows the prediction of the system, the prediction

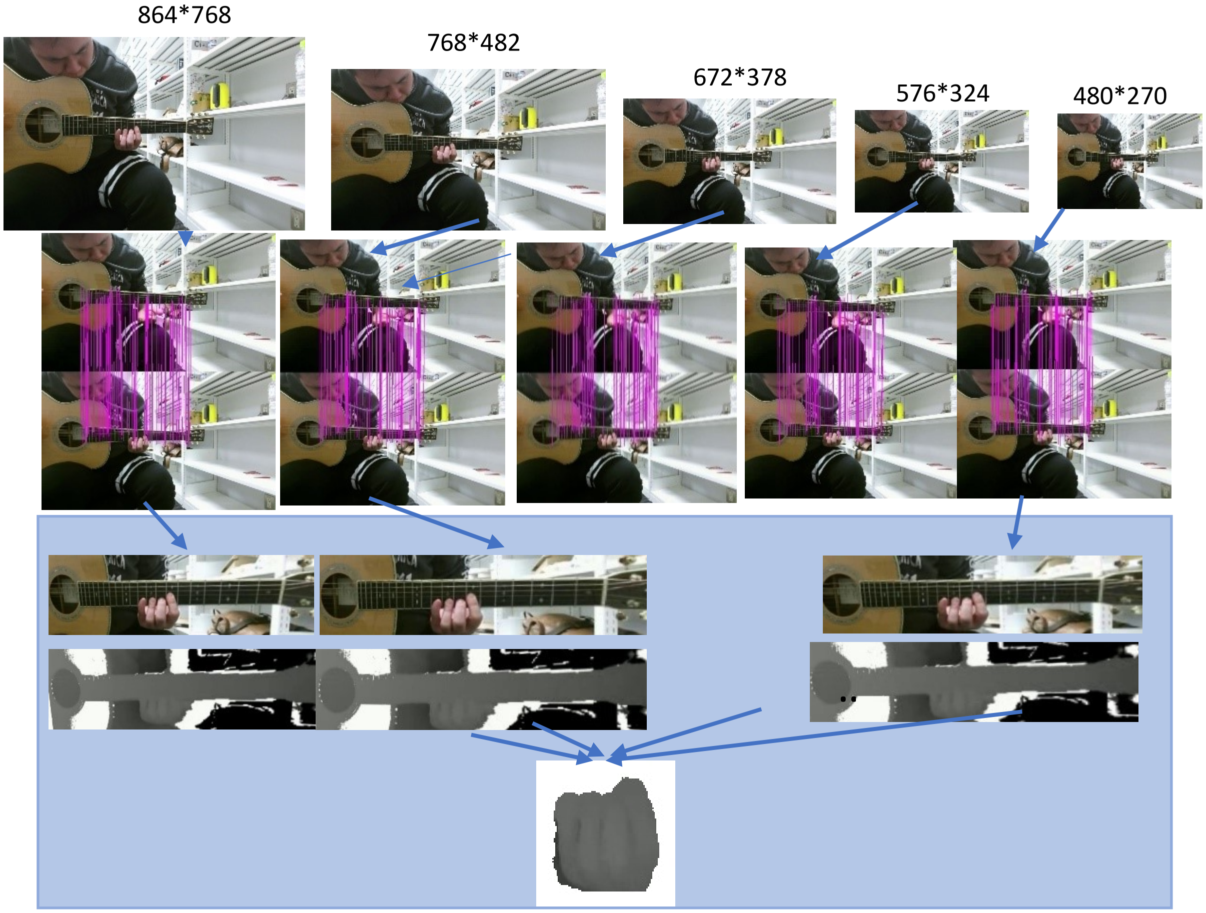


Fig 7.5 The Example of the Test for the Robustness against Resolution Changes

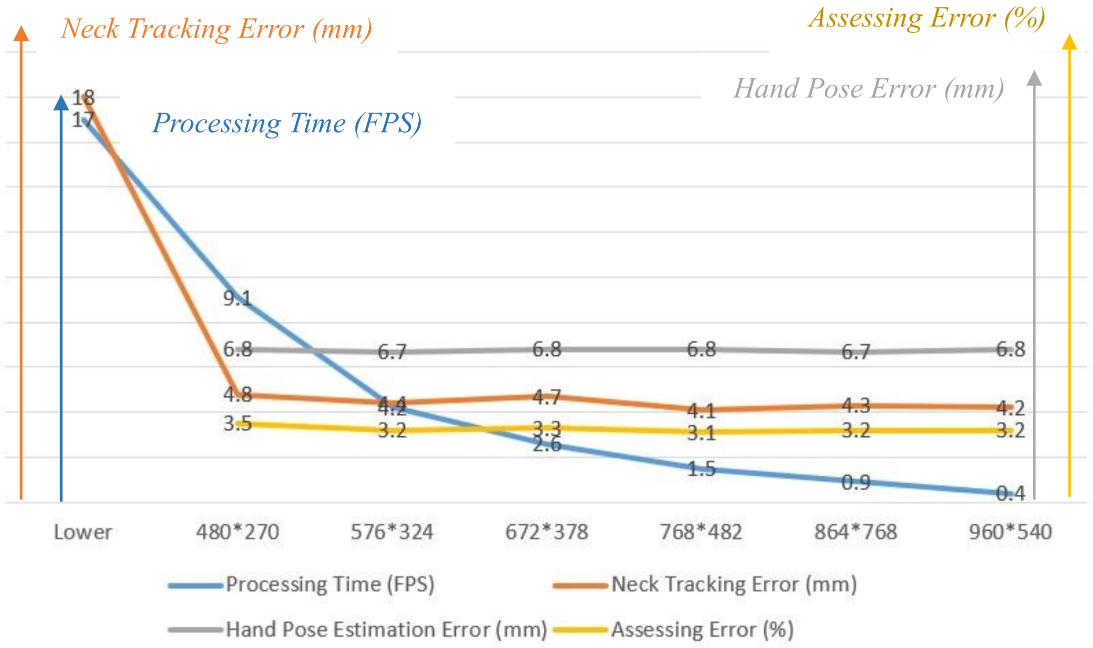


Fig 7.6 The Robustness against Resolution Changes

of mid-level players and the prediction of beginners respectively. The mean error of the system's prediction is only 0.063, and the variance is 54.3; on the other hand, the mean error of human mid-level players is 0.113, and the variance is 501.

7.5.5 Robustness Against Resolution, Intensity and Distance Changes

In this section, the experiments, which are based on the input videos of different resolution, intensity and distances between the camera and the player are conducted to test the robust of the system.

(1) Resolution

In the experiment, the different versions of the resolution of the input videos are set to 960*540 (original input), 864*768 (0.9*original size), 768*432 (0.8*original size), 672*378 (0.7* original size), 576*324 (0.6* original size), 480*270 (0.5* original size). The input videos with different resolution are shown in Fig 7.5.

In Fig 7.5, five different versions of the input resolution are shown: since the SIFT feature and the proposed Modified RANSAC method tracks guitar neck effectively in five different conditions of the resolution of the input videos, guitar neck can be projected into the middle of the new image sequences with the fixed resolution (1300* 300) as mentioned in Chapter 4, the hand pose can also be correctly segmented (the last line in Fig 7.5).

Figure 7.6 shows the mathematical result of the experiment: (1) for processing time, since the resolution changes, the processing time of each frame decreases significant (FPS increasing); (2) for neck tracking error, if the resolution is larger than 480*270, it is found that there is few changes in the neck tracking error (around 4.8 mm); if the resolution is lower than 480 *270, the guitar neck cannot be tracked accurately anymore; (3) for hand pose estimation error and assessing error, if the resolution is larger than 480*270, there is also few changes in the experimental result, and the reason is that, since the guitar neck can be projected into the new image sequence, the hand pose of each frame can be segmented correctly; if the resolution is lower than 480*270, since the guitar neck cannot be projected correctly, obviously the result of hand pose estimation and the assessing cannot be correct anymore.

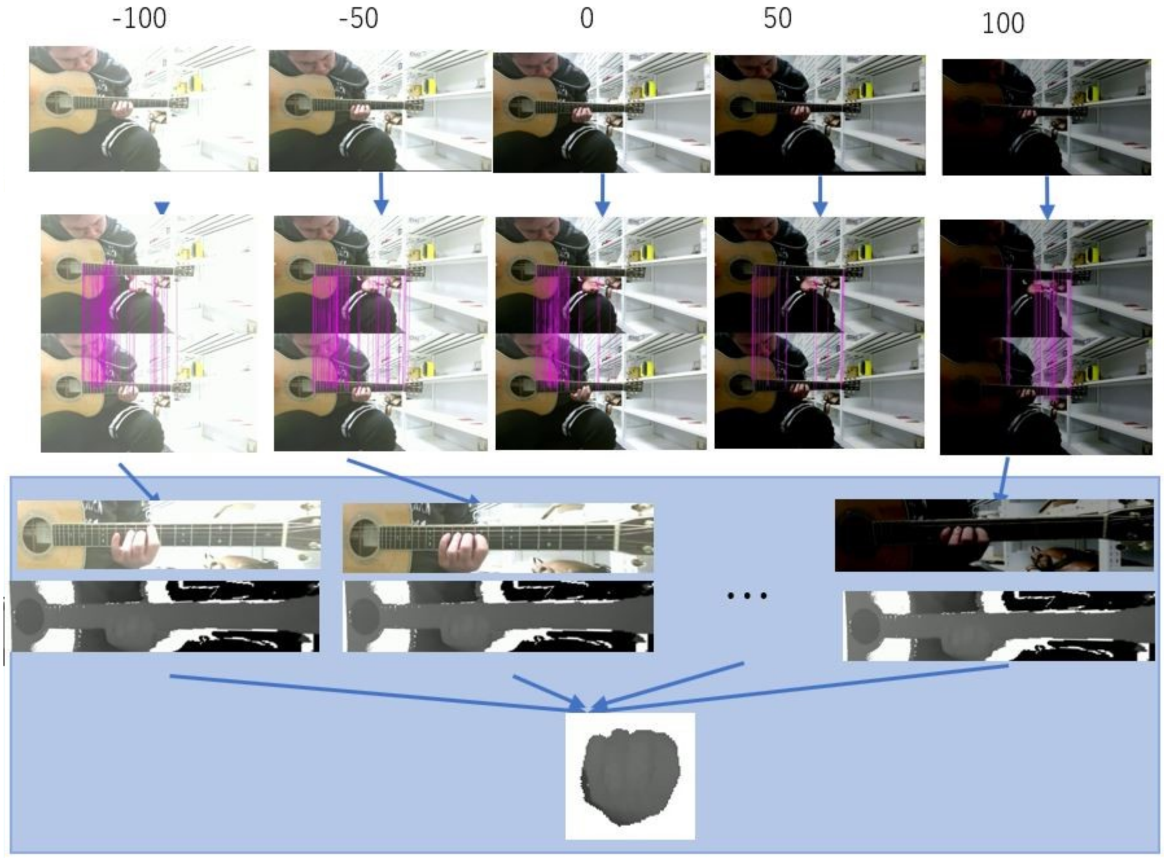


Fig 7.7 The Example of the Test for the Robustness against Pixel Intensity Changes

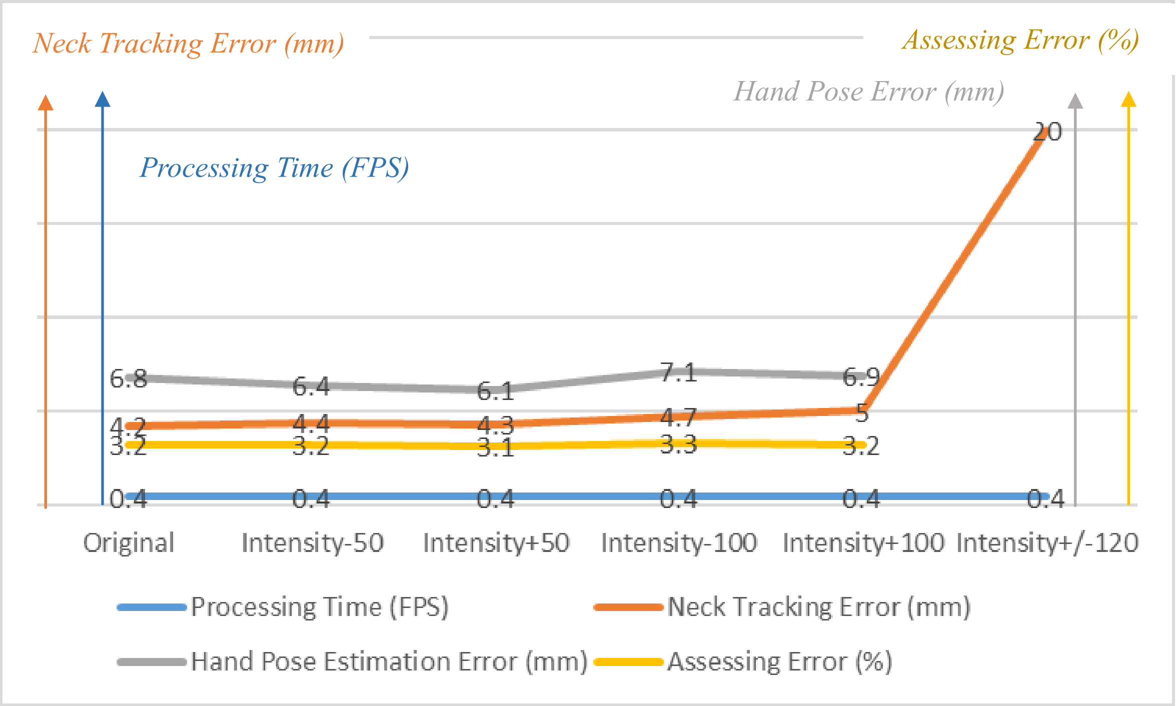


Fig 7.8 The Robustness against Pixel Intensity Changes

(2) Intensity

In the experiment, the different versions of the intensity of the input videos are set to -100, -50, 0 (original input), 50, 100. In order to test them fairly, the same video set is utilized in this experiment by only changing the pixel intensity by using the classic image processing algorithm:

$$y(p) = x\ (p) + i, i \in (-100, -50, 0, 50, 100) \tag{8.1}$$

where $x\ (p)$ is the original input pixel, while $y(p)$ is the output of changed intensity of the input video.

Figure 7.7 and 7.8 show examples and the mathematical results of the test for the robustness against pixel intensity changes. Based on the test, if the intensity of the pixels of the videos changes within +100 and -100 linearly in Eq. (8.1), there is also very few changes in the result of the processing time, the neck tracking error, the hand pose estimation and the assessing result; if the intensity of the pixels of the videos changes larger than +100 or smaller than -100, the proposed method cannot track the guitar neck accurately, and furthermore it also cannot estimate the hand pose of the guitarist and assess the playing of the guitarist anymore.

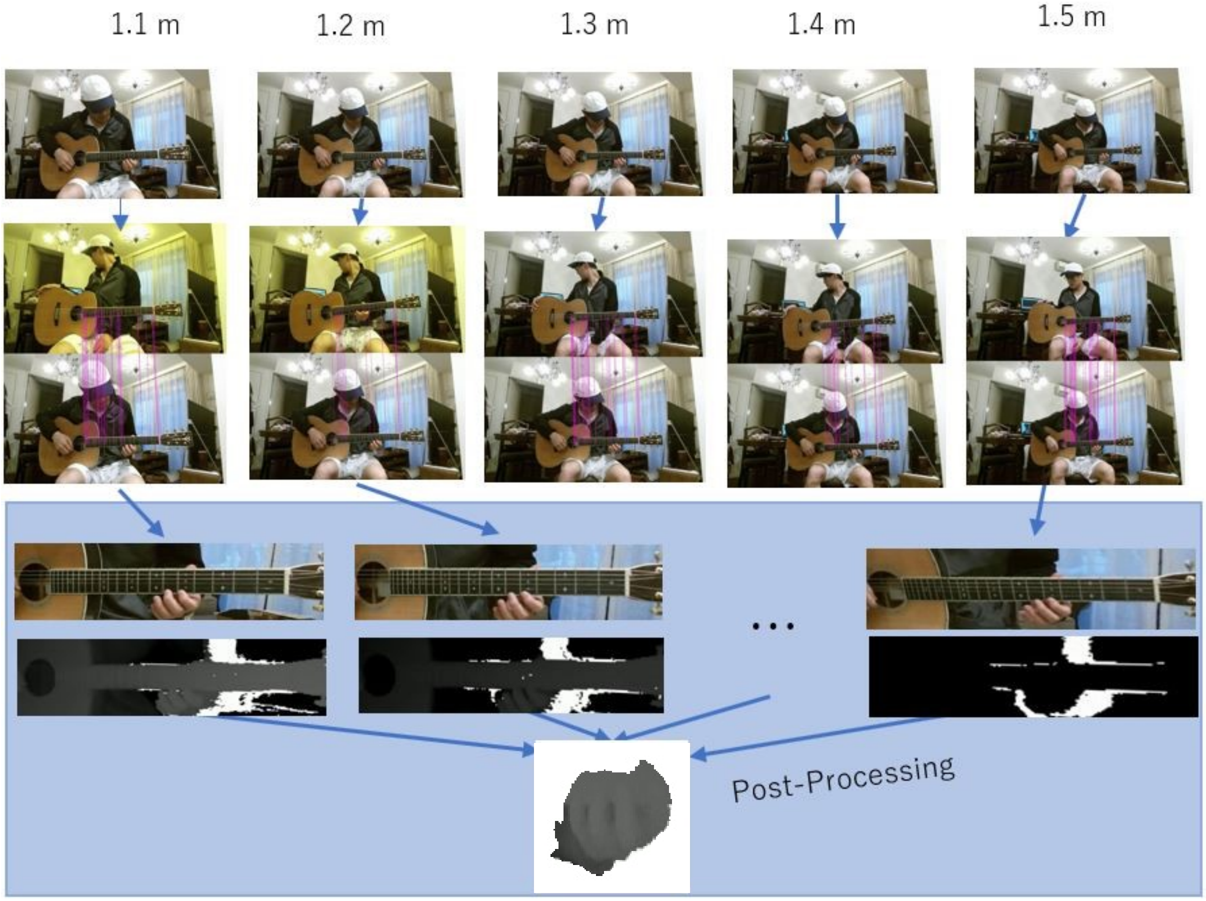


Fig 7.9 The Example of the Test for the Robustness against Distance Changes

.

(3) Distance

In the experiment, the different versions of the distance between the camera and the guitar players are set to 1.1-meter (original input), 1.2-meter, 1.3-meter, 1.4-meter, 1.5-meter and 1.6-meter.

Figure 7.9 and 7.10 show examples and the mathematical results of the test for the robustness against distance changes between the guitarist and the camera. Based on the test, if the distance of that changes within 1.1-meter and 1.6-meter, there is also very few changes in the result of the processing time, the neck tracking error, the hand pose estimation and the assessing result. The result is as same as the resolution change, because the distance becomes larger, the guitarist and the guitar neck also become smaller in the image scene. The only difference between the distance changes and the resolution changes is the processing time: in the test of distance changes, the resolution does not

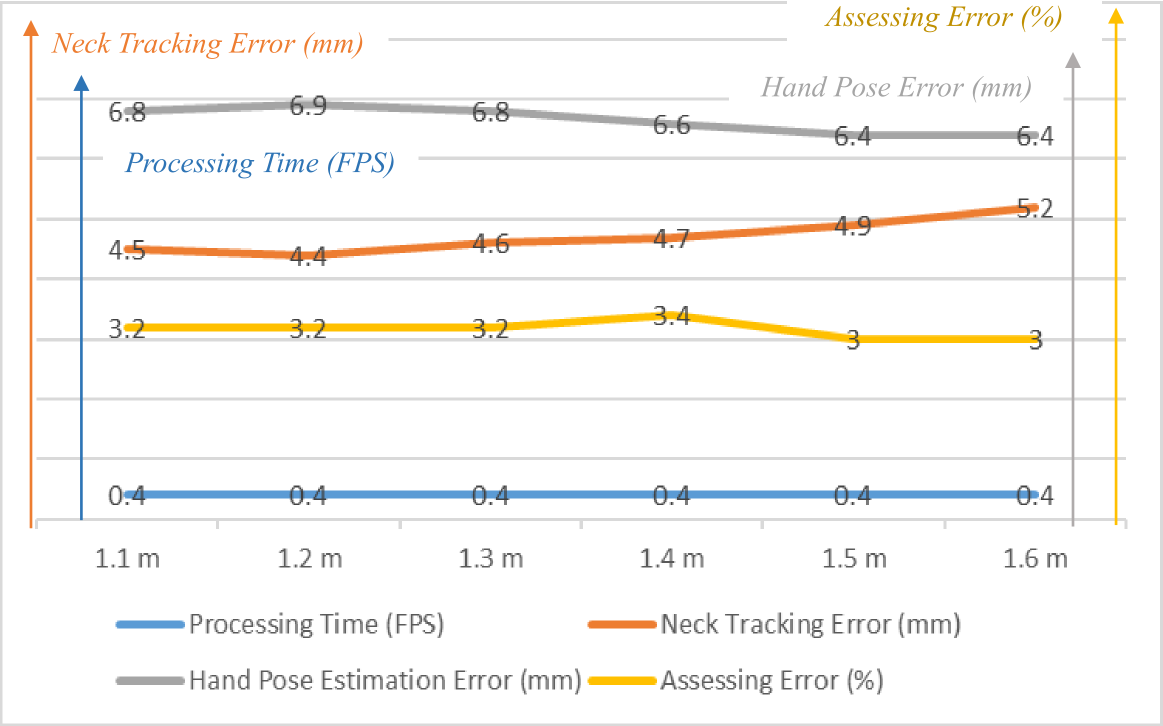


Fig 7.10 The Robustness against Distance Changes

change, therefore the processing time also does not change very much; on the other hand, in the test of the resolution changes, since the number of the processed pixels changes, the processing time also changes a lot.

7.6 Discussion

In this Chapter, the proposed guitar fingering assessing module applies a 3D-DCT and SVR based method to automatically assess how well guitarist perform in video by extracting the spatio-temporal hand pose features of guitarist and estimating a regression model that predicts the scores of guitar playings. The contribution is summarized as follows:

(1) It is the first research that solving the problems of automatically assessing the musician's performance and outputting the feedback to the musicians to help them to improve their skills. Specifically, by only inputting the video of the guitar play, it can automatically evaluate the guitarist's performance by outputting a score.

(2) Compared with the states-of-art [79], this module proposes a new 3D-based feature to calculate the effective representative form of joint-based movements. More specifically, the proposed 3D-

DCT feature not only calculates the feature of 3D placement of the hand pose during guitar playing, but more represents an effective form of the hand movement transition, which is one of the most difficult skills in guitar fingering.

(3) The proposed 3D-DCT can be easily generalized into other human actions by only labeling the training data; on the other hand, the traditional action assessing-based related works [89-91] define evaluation function for every single action, which requires much more human labor than the proposed method.

.

The 3D-DCT and SVR-based guitar fingering assessing achieves the above contribution is because of the following reasons:

(1) The 3D-DCT feature changes the hand movement from time domain to frequency domain, therefore it can represent and address the "change" of hand pose during the guitar playing better;

(2) Compared with other features of frequency domain such as DFT, its representation tends to have more of its energy concentrated in a small number of coefficients, and it only uses the real value part of cosine function, therefore when fitting the DCT into SVR regression model, it is more accurate as the imaginary part of other frequency domain has to be deleted.

The limitation tests described in Section 7.5 show not only the limitation of the guitar fingering assessing result but also the results of other modules of the whole system, which includes guitar neck tracking module, hand pose estimation when the pixel intensity, the distance between the guitarist and the camera, the resolution changes. The limitation result is shown as follows:

(1) In the limitation test against resolution changes: if the resolution is lower than 480 *270, the guitar neck cannot be tracked accurately anymore; the hand pose of each frame cannot be segmented correctly, and hand pose estimation and the assessing cannot be correct anymore. If the resolution is larger than 480 *270, not matter how resolution changes, the guitar neck tracking, the hand pose estimation and the assessing can be correctly tracked or evaluated.

(2) In the limitation test against intensity changes: if the intensity of the pixels of the videos changes within +100 and -100 linearly, the neck tracking error, the hand pose estimation and the guitar fingering assessing can be correctly tracked or evaluated; if the intensity of the pixels of the videos changes larger than +100 or smaller than -100, the proposed system cannot track the guitar neck accurately, and furthermore it also cannot estimate the hand pose of the guitarist and assess the playing of the guitarist anymore.

In the limitation test against distance changes: if the distance between the guitarist and the camera is within 1.6 meters, the results of the guitar neck tracking, the hand pose estimation and the fingering assessing show few changes; in other words, the accuracy does not change much when that distance is within 1.6 meters. On the other hand, if the distance between the guitarist and the camera is over 1.6 meters or smaller than 0.9 meters, the test result of the distance change is not accurately anymore because (a) if the distance is smaller than 0.9 meters, the Kinect camera cannot capture the depth information; (b) if the distance is larger than 1.6 meters, the guitar neck area becomes very small in images, therefore, it cannot be tracked anymore, and also the hand pose estimation and the fingering assessing cannot be conducted.

7.7 Conclusion

In this chapter, after 3D position of the 16 joints of guitarist's hand are estimated in Hand Pose Estimation Module, first the spatio-temporal hand pose is formulated as a two-dimensional matrix for each video of guitar playing: the horizontal axis of the matrix is the 3D coordinates of 16 joints in one frame lined in one row, while the vertical axis is the frame number. Then, the 3D-DCT (Discrete Cosing Transform) feature of the 16 joints' movement of one guitar playing video is proposed by calculating the inner product between the spatio-temporal matrix of hand pose and the DCT matrix. Finally, a supervised regression SVR model is training by using the 3D-DCT features to predict how well guitarist plays in the video by outputting the general score (full mark is 100 points) of the video.

Experimental results show (a) high mean-rank correlation (0.68) for the proposed 3D-DCT feature; (b) better prediction than mid-level players, that fulfills the expectation before conducting the research: the feedback of the system should be better than human mid-level players; (c) statistical effectiveness of *T-Test* result between the proposed method and other related methods: the p value of the T-test between 3D-DCT and 3D-STIP is **0.029** at statistical significant at an alpha level of 0.05; the p value of the T-test between 3D-DCT and 3D-DFT is **0.0095** at statistical significant at an alpha level of 0.05; the p value of the T-test between 3D-DCT and AQA [15] is **0.01598** at statistical significant at an alpha level of 0.05.

Furthermore, the limitation experiments based on the input videos of different resolution, intensity and distances between the camera and the player are conducted to test the robust of the system: (1) the limitation of the resolution of the input video is 480*270: if the resolution is lower than 480*270, the system cannot work correctly to assess the fingering of the guitarist; (2) limitation of the intensity change of the input video is [-50, 50]: if the intensity of the pixels linearly change over the range of [-50, 50], the system cannot work correctly to assess the fingering of the guitarist; (3) limitation of the distance change of the input video is [1.1 m, 1.6 m]: if the distance between the camera and the user is larger than 1.6 m, the system cannot work correctly to assess the fingering of the guitarist.

Chapter 8 Conclusion and Future Work

8.1 Conclusion

8.1.1 Contribution

Towards the actualization of an autonomous guitar fingering teaching system, this thesis has proposed three modules: guitar neck tracking, hand pose estimation and guitar fingering assessing modules. Each module's contribution is described in the following.

(1) Guitar Neck Tracking Module

For the guitar neck tracking, after inputting the video of guitar playing, SIFT feature points are detected on every frame as SIFT Feature is invariant to rotation, illumination and scale changes in images; then a KD-tree searching based algorithm is utilized to match the SIFT features between the first frame and any other frame of input videos; furthermore, a modified version of RANSAC (Random Sample Consensus) to overcome the occluded SIFT issue: SIFT feature is overlapped and occluded by fingers of guitar players during guitar playing. The proposed modified RANSAC filters out and eliminates the mis-matched feature points due to the occluded SIFT issue, and then recover all the mis-matched features; finally, to suppress the effect of the guitar neck motion, the tracked guitar neck area at each frame is rectified so that the centroid of the guitar neck area is constantly centered at each frame, and the neck's long and short sides are parallel to the horizontal and vertical axes at each frame, respectively. Owing to this rectification, how the guitar player shakes or swings the guitar neck while playing does not affect the subsequence process.

Experiments using 50 videos of guitar playing with nearly 300 frames of the color images (also 300 frames of depth) of different guitar plays under different conditions show promising results of the proposed method: the total mean tracking error is only 4.2 mm and variance is 1.5 mm.

The contribution of the proposed guitar neck tracking module is summarized as follows:

(a) Despite occlusions caused by the guitarist's hand during guitar playing, the guitar neck can be tracked much more accurately than related works [9, 10, 17]. The mean tracking error is only 4.2 *mm,* which is only 4/10 of the distance between adjacent strings on the guitar fretboard. Furthermore, the variance value 1.5 indicates that the proposed method is also stable and robust enough to further the subsequence research effectively.

(b) To deal with the case in which guitarists move the guitars during playing, the rectification method for the guitar neck area at each frame is proposed so that the neck is centered, and the neck's long and short sides are parallel to the horizontal and vertical axes at each frame. This method makes the subsequence process of this thesis easy to conduct, because once the guitar neck is tracked and rectified to the centered position, it is easy to recognize or assess the fingering of guitarist by only analyzing the hand pose information of the guitarist.

(c) Limitation test shows the proposed method tracks the guitar neck effectively and robustly even under the situation that guitarist shakes or swing the guitar neck aggressively on purpose. The method is robust to the 3-dimensional rotations and translation movement of guitar.

(2) Hand Pose Estimation Module

Two algorithms for hand pose estimating are proposed: a ROI (Region Of Interests) associated particle filter based fingertip tracking algorithm and a CNN (Convolutional Neural Network) based hand joint estimation algorithm. Two algorithms share the same input that is the result of guitar neck tracking module, and work in a parallel manner in the hand pose tracking module.

For the ROI associated particle filter-based fingertip tracking algorithm, first a CNN-based hand segmentation net is used to discriminate the hand area from the background. Then, the template matching and reversed Hough Transform are performed to the hand areas so that the count map for fingertip candidates is generated using the segmentation result, where the results of the template matching and reversed Hough Transform are used as weighted features to extract the fingertip candidates. Furthermore, a temporal grouping is applied to remove noise and group the same four fingertips (index finger, middle finger, ring finger, little finger) on the successive count maps. Then, an ROI association algorithm is utilized to associate the four fingertips with their individual trajectories on the frame-by-frame count maps. Here, for this ROI association algorithm, three patterns for tracking fingertips movement during the whole process are defined: the active pattern, adding pattern, vanishing pattern. All the tracked trajectories of fingertip candidates are fitted into one of these three patterns in order to solve the problem such as self-occlusion, joint-finger etc. Finally, the particle filter is utilized to track the fingertips by distributing particles within the associated ROIs of fingertips at every two adjacent frames of the video.

Experimental results show the proposed method outperforms the current state-of-art tracking algorithm with high accuracy: the mean error 6.5, 3.3, 4.9, 6.0 pixels for fore finger, middle finger ring finger and little finger respectively.

The contribution of the proposed ROI association particle filter-based fingertip tracking algorithm is summarized as follows:

(a) The proposed method tracks four fingertips in a very high accuracy. Compared with related works [10, 12, 83], four fingertips are tracked with a low tracking error of 5.2 pixel on average. The proposed method even outperforms the deep learning-based states-of-art [21] in the tracking accuracy of fingertips.

(b) The ROI association-based idea makes the proposed method robust to the joint-finger, self-occlusion, frame-out problems, which are the most difficult issues in fingertip tracking and guitar playing situation.

(c) Compared with other deep learning-based tracking algorithm, the proposed tracking method does not need off-line training process, which causes huge amount of human resources to label the training data; therefore, it is very efficient in processing time and usage of manual labor.

For the CNN (Convolutional Neural Network) based hand joint estimation algorithm, first three convolutional layers and two max-pooling layers output 512 channels of feature maps; then two fully-connected layers with 1024 notes respectively are connected after convolutional process; furthermore, instead of directly estimating the 3D position of each joint, a lower parameter space of a fully-connected layer with only 24 notes is utilized; finally, a fully-connected layer with 3*J (J is the number of the joints, in the case, J=16) notes output the 3D position of hand pose.

Experiments are conducted using videos of guitar plays under different conditions. For the hand region segmentation, the proposed method outperforms the related works in terms of segmentation accuracy (98%) and training efficiency (only 420 training images). For the CNN (Convolutional Neural Network) based hand joint estimation algorithm, it shows a competitive accuracy (mean error of 6.1 pixels for 16 joints) with state-of-arts but outperform them in time efficiency for both training and testing (only 4 *hours* for training and 0.19 *ms* for testing).

The contribution of the proposed CNN-based hand pose estimation algorithm is summarized as follows:

(a) Because the FCN hand segmentation achieves 98% accuracy for correctly segmenting the hand area in the first step, compared with other deep learning-based methods [13, 15, 19], the proposed method can cope with the situation while the users are holding an object, such as guitar neck in their hand; while the states-of-arts [13, 15, 19] can only estimate the hand pose when users hold nothing in their hand.

(b) The proposed CNN structure adopts a low dimensional embedding of hand parameters. More specifically, the proposed method implements a fully-connected layer with only 24 notes in the middle of all fully-connected layer, which makes the network achieves early same accuracy with states-of-art [21], but highly outperforms it in training and test efficiency.

(c) The proposed CNN based method is also robust to the joint-finger, self-occlusion, frame-out problems, which are the most difficult issues in fingertip tracking and guitar playing situation.

(d) As there are 13 peoples' data in the training dataset, the proposed method is not sensitive to the gender, the personal difference, such as the bone length of the hand; while in the related works, such as hand model-based method [11], it highly depends on the parameters of hand model, which makes difficult to estimate the hand pose for different people.

(3) Fingering Assessing Module

For the guitar fingering assessing, after 3D position of the 16 joints of guitarist's hand are estimated in (2), first the spatio-temporal hand pose is formulated as a two-dimensional matrix for each video of guitar playing: the horizontal axis of the matrix is the 3D coordinates of 16 joints in one frame lined in one row, while the vertical axis is the frame number. Then, the 3D-DCT (Discrete Cosing Transform) feature of the 16 joints' movement of one guitar playing video is proposed by calculating the inner product between the spatio-temporal matrix of hand pose and the DCT matrix. Finally, a supervised regression SVR model is training by using the 3D-DCT features to predict how well guitarist plays in the video by outputting the general score (full mark is 100 points) of the video.

Experimental results show (a) high rank correlation (0.68) for the proposed 3D-DCT, (b) better prediction than mid-level players, that fulfills the expectation before conducting the research: the feedback of the system should be better than human mid-level players.

The contribution of the proposed fingering assessing module is summarized as follows:

(a) It is the first research that solving the problems of assessing the musician's performance and outputting the feedback to the musicians to help them to improve their skills.

(b) Compared with the states-of-art [79], this module proposes a new 3D-based feature to calculate the effective representative form of joint-based movements. More specifically, the proposed 3D-DCT feature not only calculates the feature of 3D placement of the hand pose during guitar playing, but more represents an effective form of the hand movement transition, which is one of the most difficult skills in guitar fingering.

(c) The proposed 3D-DCT can be easily generalized into other human actions by only labeling the training data; on the other hand, the traditional action assessing-based related works [89-91] define evaluation function for every single action, which requires much more human labor than the proposed method.

8.1.2 Limitation

(1) Guitar Neck Tracking Module:

As described in Section 4.6.5, this thesis tests the limitation of the input video against the rotation and translation movement by aggressively shaking the guitar on purpose. Based on the result of the experimental test, the limitation of guitar neck tracking is shown as follows:

(a) it is very robust to the translation movement of guitar neck: as long as guitar neck area is in the image scene, no matter how it moves, guitar neck can be tracked;

(b) it is robust to the rotation movement using the right hand of guitarist as the pivot of the rotation; averagely, the limitation angle of the rotation is 20 degrees: if the angle of the guitar necks of SIFT-matched two frames is larger than 20 degrees, guitar neck cannot be tracked anymore;

(c) it is robust to the upward rotation using the center line of guitar neck as the pivot, the limitation angle of the rotation is averagely 8 degree, but it is not robust to the downward rotation using the center line of guitar neck as the pivot at all. it is because when the guitar neck is rotated downward, the intensity of pixels changes a lot since the lighting source is always upon the guitarist, therefore the SIFT features changes a lot and they cannot be matched anymore;

(d) it is very robust over the rotation movement using the horizontal center of the guitar neck as the pivot of the rotation: the limitation angle of the rotation is averagely 15 degrees.

(2) Hand Pose Estimation Module

For the ROI associated particle filter-based fingertip tracking algorithm, the limitation is shown as follows:

(a) The ROI associated particle filter-based fingertip tracking algorithm only tracks fingertips in

2D space without depth information. As mentioned in Section 8.1.1, the proposed method tracks the fingertips of guitarists more accurately than CNN based algorithm; however, because it lacks the depth information of fingertips, by using the 2D fingertip tracking result, it is not accurate for fingering assessing compared with 3D CNN based method.

(b) The ROI associated particle filter-based fingertip tracking algorithm only tracks four fingertips of guitarist's hand. However, assessing the fingering of the guitarist should not only focus on the movement of fingertips, but also more joints that can fully represent the movement of the hand of the guitarist.

For the CNN (Convolutional Neural Network) based hand pose estimation algorithm, the limitation is shown as follows:

Although the proposed method achieves the same accuracy as the large-scale training-based states-of-art [21] by only training on a small dataset, the accuracy is still not as good as the expectation of the authors, especially, in the accuracy of estimating top four joint of each finger (four fingertips) when occlusion happens.

Other limitations of CNN based hand pose estimation is listed with (3) Guitar Fingering Assessing Module, as the experiments tested in Section 7.5.5 show the limitation of the whole system from Module 1 to Module 3.

(3) Guitar Fingering Assessing Module

In Section 7.5.5, this thesis also tests the limitation of the robustness against resolution, intensity and distance changes respectively, and the limitation is shown as follows:

(a) In the limitation test against resolution changes: for processing time, since the resolution changes, the processing time of each frame decreases a lot (FPS increasing); for neck tracking error, if the resolution is larger than 480*270, it is found that there is few changes in the neck tracking error (around 4.8 mm); if the resolution is lower than 480 *270, the guitar neck cannot be tracked accurately anymore; for hand pose estimation error and assessing error, if the resolution is larger than 480*270, there is also few changes in the experimental result, and the reason is that, since the guitar neck can be projected into the new image sequence, the hand pose of each frame can be segmented correctly; if the resolution is lower than 480*270, since the guitar neck cannot be projected correctly, obviously the result of hand pose estimation and the assessing cannot be correct anymore.

(b) In the limitation test against intensity changes: if the intensity of the pixels of the videos changes within +100 and -100 linearly in Formula 8.1, there is also very few changes in the result of the processing time, the neck tracking error, the hand pose estimation and the assessing result; if the intensity of the pixels of the videos changes larger than +100 or smaller than -100, the proposed method cannot track the guitar neck accurately, and furthermore it also cannot estimate the hand pose of the guitarist and assess the playing of the guitarist anymore.

(c) In the limitation test against distance (between camera and guitarist) changes: if the distance of that changes within 1.1-meter and 1.6-meter, there is also very few changes in the result of the processing time, the neck tracking error, the hand pose estimation and the assessing result. The result is as same as the resolution change, because the distance becomes larger, the guitarist and the guitar neck also become smaller in the image scene. The only difference between the distance changes and the resolution changes is the processing time: in the test of distance changes, the resolution does not change, therefore the processing time also does not change very much; on the other hand, in the test of the resolution changes, since the number of the processed pixels change, the processing time changes significantly.

Besides, although the proposed method outperforms the human middle level player in assessing the guitar fingering, the mean error of the fingering assessing is still not as good as professional guitar players. This is due to the number of training data is not as much as the previous knowledge of the professional expert: considering the professional guitar player in the experimental tests are all experts of guitarists, all of who have been playing the guitar over 5 years; while in proposed method, only 100 videos are used to train the SVR regression model.

8.1.3 Discussion

(1) In Guitar Neck Tracking Module, the proposed Modified RANSAC-based system tracks guitar neck accurately and robustly, and it solves the problems of tracking the guitar neck in the circumstance of the rotation, translation of guitar neck in 3D spaces and the occlusion by the hand of the guitarist. First, as SIFT feature is utilized in the module, it can accurately detect image feature within the guitar fretboard to track the whole fretboard. However, during the guitar plays, since the performer's fingers frequently overlap the fretboard, the feature points cannot always be matched accurately. Therefore, by using the proposed

Table 8.1. Comparison between 2D Fingertip tracking and 3D Hand Pose Estimation

	2D Fingertip Tracking	*3D Hand Pose Estimation*
Accuracy	*Higher Accuracy for Fingertips,* *Only Track Fingertips (4 Joints)*	*Lower Accuracy for Fingertips* *16 Joints Tracking Result*
Time Efficiency	*Low Efficiency of Processing* *Not Need Offline -Train*	*0.19 ms of Processing Time for Each Frame* *Huge Number of Training Data and Time*
Future Prospects	*Requires Huge Number of Parameter, Hard to Tune*	*Requires Huge Number of Data, Human Label*

Modified RANSAC algorithm that filters out the tracking error of the feature points due to the overlapping issue mentioned before, perspective transformation matrix is obtained between the correctly matched feature points detected at the first and other frames. Furthermore, as the module projects the guitar neck images to a new image sequence with fixed resolution, it can accurately track guitar neck despite of changing the resolution, pixel intensity and distance between the camera and the guitarist, Consequently, the guitar neck is tracked correctly based on the perspective transformation matrix.

(2) In Hand Pose Estimation Module, hand pose of guitarist is estimated by tracking the fingertips in 2D space and estimating 16 joints of hand in 3D space in Chapter 4 and Chapter 5 respectively. Section 4.4 shows the total mean error of 2D fingertip tracking is 6.5, 3.3, 4.9, 6.0 pixels for fore finger, middle finger ring finger and little finger, while Table 5.1 shows the mean estimation error of 3D hand pose (16 joints) is 6.1 *mm (nearly* 4.3 pixels) and the mean estimation error of 3D fingertips (4 joints) is over 10 *mm* (13.2 *mm* 12.1 *mm*, 11.3 *mm*, 9.2 *mm* for fore fingertip, middle fingertips, ring fingertip and little fingertip respectively). However, 2D fingertip tracking and 3D hand pose estimation show respective merits and demerits from not only the accuracy but also other aspects. Table 8.1 shows the comparison result between 2D fingertip tracking and 3D hand pose estimation.

Furthermore, the CNN net structure needs to be constructed “deeper” when a better estimation result is needed by adding more training data to the existing dataset. Current network shows the tendency of underfitting when a larger dataset is used on the proposed network.

In Fingering Assessing Module, as the proposed 3D spatio-temporal DCT feature calculates a more accurate representative form for the transition of finger movement in guitar fingering assessing, the module accurately outputs the score based on the performance of the guitarist by fitting the proposed feature to a linear SVR regression model. The spatio-temporal DCT feature, which is better than other representative forms of fingering movement, shows two merits: a. the DCT feature changes the hand movement from time domain to frequency domain, therefore it can represent and address the “change” of hand pose during the guitar playing better; b. compared with other features of frequency domain such as DFT, its representation tends to have more of its energy concentrated in a small number of coefficients, and it only uses the real value part of cosine function, therefore when fitting the DCT into SVR regression model, it is more accurate as the imaginary part of other frequency domain has to be deleted.

8.1.4 Significant Digit

In this thesis, the significant digit of all the experiments including guitar neck tracking, hand pose estimation (both 2D and 3D) is accurate to 1 decimal after digit point with the unit of ***mm*** (millimeter). As described in the experiment of each module of the guitar fingering assessing system, for each algorithm, the accuracy is calculated by averaging the error distance between experimental result and ground truth of each frame, for example, the tracking accuracy of guitar neck is calculated by counting and averaging the tracking error of each four corners of guitar neck in the dataset, which includes over 100 videos with nearly

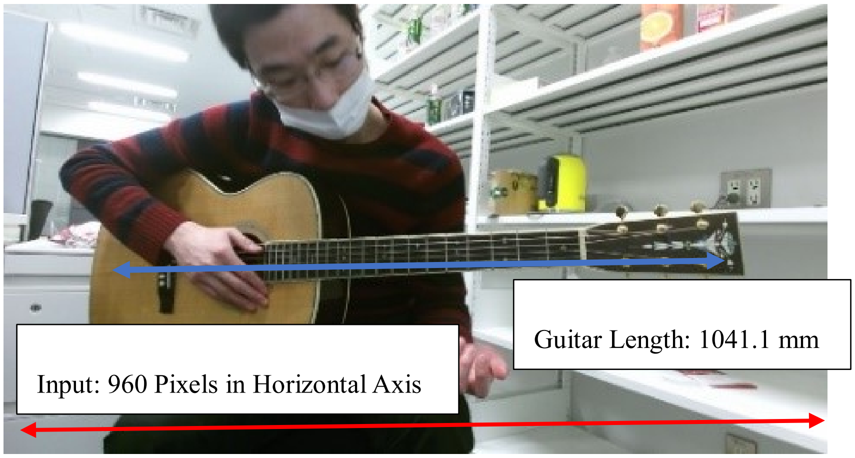


Fig 8.1 The Original Input of The Guitar Fingering Assessing System

7000 frames, therefore, the accuracy (error distance) is averaged and divided into infinite unrepeated decimals.

The reason that the accuracy of guitar neck tracking module and hand pose estimation module are set to 1 decimal after digit point is based on the reasons shown as below:

(1) Input Accuracy

As described in Chapter 4, the original input of the guitar fingering assessing system is captured by Kinect, therefore, the input resolution of video is 960 pixels * 540 Pixels; on the other hand, the standard 41-inch guitar has nearly 1041.1 mm in length, and considering the remaining blank spaces on the two sides of guitar neck in the input frame shown in Fig 8.1 and the motion of the guitar player such as rotation of guitar neck, the distance between user of the system and the camera and etc., it is believed that the 960 pixels in the horizontal axis of image space could be viewed and represented the real distance of nearly 1800 *mm* in real world. As the experimental results of guitar neck tracking and hand pose estimation are both calculated based on pixel processing results, the tracking accuracy of guitar neck and hand pose could be at least set to 0.1 *mm* (1 decimal after digit point).

(2) Possibility of Comparison with State-of-arts

As described before, the accuracy of the system is calculated by averaging the error distance between experimental result and ground truth of each frame in a huge amount of dataset with nearly 7000 frames, in order to compare the accuracy between the proposed methods and other state-of-arts, setting the accuracy to 1 *mm* is obviously not a good idea, as it would be very hard to compare the accuracy. The accuracy with 1 or 2 decimals after digit point is easier to do the self-comparison or compare with other state-of-arts.

(3) Possibility of Measurement

In the physical measurement, the accuracy of a normal vernier calipers is 0.1 *mm*, which means in physical measurement of tracking accuracy in real world, the most accurate way to measure the tracking error of guitar neck and hand pose could be set to 1 decimal after digit point with the unit of *mm*.

Based on the reasons shown above, setting the tracking accuracy to 0.1 *mm* (1 decimal after digit point) is the best way to measure the accuracy in guitar neck tracking module and hand pose estimation module.

8.2 Future Work

In the evaluation part of each module in this system, not only the accuracy but also the limitations are discussed: for example, the limitation of rotation angle of guitar neck in guitar tracking, only score that assesses the general performance of guitarist without any improvement advices for guitarists and etc. In the future work, the work that needs to be added into the system are shown as follows:

(1) Right Hand Fingering Assessing

Guitar playing is the art performance required motion of both hands. In this thesis, both the ROI associated particle filter based 2D fingertip tracking algorithm and 3D finger pose estimation algorithm can be applied to the left fingering: in 2D fingertip tracking algorithm, the features of fingertips are the template matching and reverse Hough Transform, which both require fingertips can be captured by camera; while guitarist normally curves right fingertips to perform music all the time, the fingertips cannot be captured by camera if the camera is set up at the front of guitarist (as only-one-camera is one of the most preferable feature of the system, so the camera needs to be placed at the front of guitarist in order to capture more information of guitar neck and guitarist). The feature of the right hand of guitarist needed to be discussed in future work of 2D fingertip tracking. On the other hand, in 3D finger pose estimation algorithm, by annotating more images of right hand of guitarist may easily predict the accurate 3D position information of the joint of the guitarist right hand, however, as the same "fingertips of right hand cannot be capture by camera" issues exists, the experiment of CNN based right hand finger pose estimation needs to be conducted as well.

Considering the current trend in image processing and the robustness, the author of this thesis highly recommends CNN based algorithm for both 2D fingertip tracking and 3D finger pose estimation. The reasons are listed as follow:

a. CNN based algorithm does not require big amount of Hyper-parameter. The Hyper-parameter indicates the parameters in an algorithm that need to be set by programmer. The traditional algorithm of image processing requires a lot of hyper-parameters need to be set, such as the voting threshold for Hough Transform, the distance threshold for ROI associated, and etc. this issue is unavoidable in traditional image processing method. However, the parameters in CNN based method is not that many.

b. The robustness of CNN based method is good as it requires much more training data than traditional method. By training on a dataset with a large number of data not only generates more accurate result, more importantly it would generate more robust algorithm as a large number dataset normally tackles with more situations. The demerit is that, it requires more human effort to collect data and label data. However, this is a big-data world right now, and a lot of dataset can be freely downloaded and accessed.

(2) Wrist and Arm Assessing

In guitar playing or other instrument playing such as piano, the movement of wrist and arm directly influences the movement of finger as human being is constructed with many joint. the player may feel uncomfortable or relax due to the bad movement of the wrist or arm that they may easily ignore the influences of wrist and arm.

(3) Audio-Assessing

The audio-based fingering assessing for guitarist needs to be added into the system in order to conduct a more accurate and robust algorithm. As mentioned in Chapter 1, in the fingering assessing of C Scale performance, some assessing rules designed by professional player such as "*Scale is performed with a good sound without muted pr buzzing notes"* or "*Scale is performed without breaks or stops*" cannot be assessed by only processing the image or video signal. Furthermore, if the audio signal is added to this system, the author believes it would become more accurate and practical to the users of the system.

(4) Guitarist-Body-Assessing

The motion of guitarist's body is another important aspect for assessing fingering of guitarist. Classical guitar requires guitarist holding the guitar neck at 45 angles with ground during the playing; left leg needs to be elevated, and right leg needs to support guitar in order to perform the beautiful and elegant music. These are the very basic rule for beginners of guitar that need to be assessed during the whole playing process if guitar beginners in order to support them to generate a better playing habit.

Furthermore, the coordination of the finger movement (both left hand and right hand), the combination of visual signal and audio signal, the coordination between body and hand are needed to be conducted in the future work.

8.3 Marketing Prospect

As described in Chapter 1, guitar is one of the most popular instruments in the world. Currently there are 50 million guitar players worldwide, and 51% of younger adults aged 18 to 34 were more likely to know how to play an instrument than older adults, and within all the younger adults who want to learning an instrument, 69% would like to learn to play guitar. Given the data before, any business concerning guitar playing is a market of at least 200 million people consisting of guitar players and the people who want to learn guitar.

It is very hard to predict the profit of an application of guitaring teaching, as it involves too many aspects, such as the cost of software development, advertisement and etc. However, in the generation of thriving and prosperous of machine learning, more and more people tend to use more effective, more efficient and lower cost substitute production of human labor such as self-driving, medical application and etc. Suppose only 10000 people deciding to buy the automatic guitar teaching system, and the price for each license is only 100 dollars, the total income would be 1,000,000 dollars.

Acknowledgements

At this place I want to thank all the people who have helped me during my doctoral program in Department of Modern Mechanical Engineering, Graduate School of Creative Science and Engineering, Waseda University, Tokyo, Japan, under the supervision of Prof. Jun Ohya.

Firstly, I would like to express my grate acknowledgement to my supervisor, Prof. Ohya, for giving me the opportunity to complete my research. During the five and a half years, he gave me valuable guidance, kind and constant help and encouragement.

I also would like to convey my sincere gratitude to my doctoral thesis review panel which included Prof. Hiroyasu Iwata and Prof. Atsuo Takanishi Graduate School of Creative Science and Engineering, and Prof. Takashi Kawai, GITS, Waseda University, Tokyo, Japan for their kind acceptance to be reviewers and their helpful comments.

In addition, I would like to give my thanks to all the members in Ohya Lab, Waseda University, especially, Ms. Dan Luo, Dr. Ye Li, Mr. Tatsuya Ogusu, Ms. Jing Yan, Mr. Yan Li, for their help and technical discussions.

Reference

[1] B. Heany, J. Li, H. Park, “Ibanez Market Strategy” Alena Noson, https://billydigital.files.wordpress.com/2015/07/ibanez-market-strategy-final.pdf, June,2017.

[2] https://orcinternational.com/solution/omnibus-surveys/

[3] “About American Academy of Guitar Mastery”, American Academy of Guitar Mastery 001, LLC, https://americanacademyofguitarmastery.com/about/ , 2018

[4] Douglas, W. “MUSIC TEACHING DEVICE AND METHOD”, United States Patent, Patent No.: US 7,030,307 B2. Jun. 10, 2002

[5] Daniele, R. and Vincenzo, L. "Guitar fingering for music performance." strings. Vol. 40. No. 45. 2005.

[6] Radisavljevic, A. and Peter F. D. "Path Difference Learning for Guitar Fingering Problem." ICMC. Vol. 28. 2004.

[7] Sayegh,S.I. "Fingering for String Instruments with the Optimum Path Paradigm" Computer Music Journal, vol.13, No. 3, Fall 1989, pp. 76-83. 1989

[8] Y. Motokawa and H. Saito, “Support system for guitar playing using augmented reality display,” in Proceedings of the 2006 Fifth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)- Volume 00. IEEE Computer Society, pp. 243–244, 2006.

[9] Joseph Scarr and Richard Green, “Retrieval of Guitarist Fingering Information using Computer Vision,” Image and Vision Computing New Zealand (IVCNZ), 2010 25th International Conference, ISSN：2151-2191 , pp. 1–7, 2010.

[10] A. Burns, “Visual Methods for the Retrieval of Guitarist Fingering”, Proceeding of the 2006 conference on New interfaces for musical expression ISBN:2-84426-314-3, pp.196-199, 2006.

[11] Chutisant Kerdvibulvech and Hideo Saito, “Real-Time Guitar Chord Estimation by Stereo Cameras For Supporting Guitarists”. In Proceeding of 10th International Workshop on Advanced Image Technology 2007 (IWAIT), pp.147-152, 2007.

[12] Chutisant Kerdvibulvech and Hideo Saito, “Guitarist Fingertip Tracking by Integrating a Bayesian Classifier into Particle Filters”. International Journal of Advances in Human-Computer Interaction (AHCI), pp121-131, 2008.

[13] Oberweger M, Wohlhart P, Lepetit V. “Hands deep in deep learning for hand pose estimation[J]”. arXiv preprint arXiv:1502.06807, 2015.

[14] Tang, D, Jin Chang, H, Tejani, A, & Kim, T. K. “Latent regression forest: Structured estimation of 3d articulated hand posture”. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pp. 3786-3793, 2014.

[15] Sinha, Ayan, Chiho Choi, and Karthik Ramani. "DeepHand: Robust Hand Pose Estimation by Completing a Matrix Imputed with Deep Features-Supplementary Material." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pp. 124-131, 2015

[16] Wan, Chengde, Probst T “Crossing Nets: Dual Generative Models with a Shared Latent Space for Hand Pose Estimation." arXiv preprint arXiv:1702.03431,2017.

[17] C.Keskin, ,F.Kırac ̧,Y.E.Kara,andL.Akarun. “Hand Pose Estimation and Hand Shape Classification Using Multi-Layered Randomized Decision Forests”. In European Conference on Computer Vision, 2012

[18] Simon, T., Joo, H., Matthews, I., & Sheikh, Y. Hand Keypoint Detection in Single Images using Multiview Bootstrapping. arXiv preprint arXiv:1704.07809, 2017

[19] Gattupalli, Srujana, Dylan Ebert, Michalis Papakostas, Fillia Makedon, and Vassilis Athitsos. "Cognilearn: A deep learning-based interface for cognitive behavior assessment." In Proceedings of the 22nd International Conference on Intelligent User Interfaces, pp. 577-587. ACM, 2017

[20] Qian, C., Sun, X., Wei, Y., Tang, X., & Sun, J. “Realtime and robust hand tracking from depth”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pp. 1106-1113. 2014.

[21] Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Kim, D., ... & Freedman, D. “Accurate, robust, and flexible real-time hand tracking”. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems pp. 3633-3642, 2015.

[22] Togootogtokh, E., Shih, T.K., Kumara, W.G.C.W., Wu, S.J., Sun, S.W. and Chang, H.H., 2017. 3D finger tracking and recognition image processing for real-time music playing with depth sensors. Multimedia Tools and Applications, pp.1-16 2017.

[23] She, Yingying, et al. "A real-time hand gesture recognition approach based on motion features of feature points." Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on. IEEE, 2014.

[24] G.Bebis, D.Egbert, M.Shah “Review of Computer vision Education”, IEEE Transactions on Education, volume 46, No.1, pages 2-21, 2003.

[25] Ke, Yan, Derek Hoiem, and Rahul Sukthankar. "Computer vision for music identification." Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005.

[26] Hollis, Mason, "Music Feature Matching Using Computer Vision Algorithms". Computer Science and Computer Engineering Undergraduate Honors Theses. University of Arkansas, Fayetteville 47, 2017.

[27] Raposo, F., de Matos, D. M., Ribeiro, R., Tang, S., & Yu, Y.. Towards Deep Modeling of Music Semantics using EEG Regularizers. arXiv preprint arXiv:1712.05197. 2017.

[28] P. Suteparuk, "Detection of piano keys pressed in video," Dept. of Comput. Sci., Stanford Univ., Stanford, CA, USA, Tech. Rep., Apr. 2014 [Online]. Available: http://web.stanford.edu/class/ee368/ Project_Spring_1314/, 2014

[29] Akbari, Mohammad, and Howard Cheng. "Real-time piano music transcription based on computer vision." IEEE Transactions on Multimedia 17.12: 2113-2121.2015.

[30] Manitsaris, S., Tsagaris, A., Dimitropoulos, K., & Manitsaris, A. . Finger musical gesture recognition in 3D space without any tangible instrument for performing arts. International Journal of Arts and Technology, 8(1), 11-29. 2015.

[31] A.,Mohammad, J., Liang, and H.,Cheng. "A real-time system for online learning-based visual transcription of piano music." Multimedia Tools and Applications (2018): 1-23, 2018.

[32] AM., Burns, "Computer Vision Methods for Guitarist Left-Hand Fingering Recognition" Input Deviees and Music Interaction Lab Schulich School of Music McGill University Montréal, Québec, Canada, 2007.

[33] V., Paul, J.,Michael. "Robust real-time face detection". International journal of computer vision, 2004, 57.2: 137-154. 2004.

[34] Togootogtokh, E., Shih, T. K., Kumara, W. G. C. W., Wu, S. J., Sun, S. W., & Chang, H. H.. "3D finger tracking and recognition image processing for real-time music playing with depth sensors". Multimedia Tools and Applications, 1-16. 2017.

[35] Burns, A. M., Bel, S., & Traube, C.. Learning to play the guitar at the age of interactive and collaborative Web technologies. 2017.

[36] Chordify. [Online]. Available: https://chordify.net

[37] Capo. [Online]. Available: http:// supermegaultragroovy.com/products/capo/

[38] Guitarmaster. [Online]. Available: http://www. guitarmaster.co.uk

[39] Zinemanas, P., Arias, P., Haro, G., & Gómez, E. “Visual music transcription of clarinet video recordings trained with audio-based labelled data”. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 463-470. 2017.

[40] Kelz, R., & Widmer, G. “An experimental analysis of the entanglement problem in neural-network-based music transcription systems”. arXiv preprint arXiv:1702.00025, 2017.

[41] Valero-Mas, Jose J., Emmanouil Benetos, and José M. Iñesta. "Assessing the Relevance of Onset Information for Note Tracking in Piano Music Transcription." Audio Engineering Society Conference: 2017 AES International Conference on Semantic Audio. Audio Engineering Society, 2017.

[42] Dalmazzo, David, and Rafael Ramirez. "Air violin: a machine learning approach to fingering gesture recognition." Proceedings of the 1st ACM SIGCHI International Workshop on Multimodal Interaction for Education. ACM, 2017.

[43] Ultimate guitar. [Online]. Available: https://www. ultimate-guitar.com

[44]] D. Fober, J. Bresson, P. Couprie, and Y. Geslin, “Les nouveaux espaces de la notation musicale,” in Actes des Journees d’Informatique Musicale JIM2015 ´ , Montreal, Quebec, Canada, 7-9, 2015.

[45] B. Ong, A. Khan, K. Ng, P. Bellini, N. Mitolo, and N. Paolo, “Cooperative multimedia environnements for technology-enhanced music playing and learning with 3d posture and gesture supports,” in Proceedings of the International Computer Music Conference (ICMC 2006), Tulane University, USA, 6-11 November 2006, pp. 135–138,2006.

[46] Kwon, T., Jeong, D., & Nam, J. . Audio-to-score alignment of piano music using RNN-based automatic music transcription. arXiv preprint arXiv:1711.04480,2017

[47] Cano, P., Batle, E., Kalker, T., & Haitsma, J. (2002, December). “A review of algorithms for audio fingerprinting”. In 2002 IEEE Workshop on Multimedia Signal Processing, pp. 169-173,2002.

[48] J. Haitsma and T. Kalker. “A highly robust audio fingerprinting system”. In Proceedings of International Conference on Music Information Retrieval, pp. 63-70, 2002

[49] Shazam Entertainment. http://www.shazam.com/

[50] Yang, G., Chen, X., & Yang, D. (2014, July). Efficient music identification by utilizing space-saving audio fingerprinting system. In Multimedia and Expo (ICME), 2014 IEEE International Conference on (pp. 1-6). IEEE,2014.

[51] Zhang, X., Zhu, B., Li, L., Li, W., Li, X., Wang, W., ... & Zhang, W. SIFT-based local spectrogram image descriptor: a novel feature for robust music identification. EURASIP Journal on Audio, Speech, and Music Processing, 2015 (1):6, 2015

[52] S., Siddharth, Emmanouil Benetos, and Simon Dixon. "An end-to-end neural network for polyphonic piano music transcription." IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP) 24.5 (2016): 927-939,2016.

[53] Schindler, A., & Rauber, A. An audio-visual approach to music genre classification through affective color features. In European Conference on Information Retrieval (pp. 61-67). Springer, Cham,2015.

[54] Rader, C.M, "Discrete Fourier transforms when the number of data samples is prime", Proceedings of the IEEE (Volume:56 , Issue: 6), ISSN : 0018-9219, pp. 1107 – 1108, 1968.

[55] OpenCV Tutorials: http://docs.opencv.org/doc/tutorials/core/discrete_fourier_transform/discrete_fourier_transform.html

[56] Lowe, D.G. Object recognition from local scale-invariant features. In International Conference on Computer Vision, Corfu, Greece, pp. 1150-1157,1999.

[57] Sunil Arya and David Mount, Approximate Nearest Neighbor Queries in Fixed Dimensions, In Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms, Austin, United States, pp. 271-280, 1993.

[58] Li, Minjie, Liqiang Wang, and Ying Hao. "Image matching based on SIFT features and kd-tree." Computer Engineering and Technology (ICCET), 2010 2nd International Conference on. Vol. 4. IEEE , 2010.

[59] HORN, Berthold K. SCHUNCK, Brian G. "Determining optical flow", In: 1981 Technical Symposium East. doi:10.1117/12.965761:International Society for Optics and Photonics, pp. 319-331, Washington, D.C., USA, April 21, 1981.

[60] Andrew Burton and John Radford, "Thinking in Perspective: Critical Essays in the Study of Thought Processes". Routledge. ISBN 0-416-85840-6,1978.

[61] David H. Warren and Edward R. Strelow. "Electronic Spatial Sensing for the Blind: Contributions from Perception", Springer. ISBN 90-247-2689-1, Springer, 1985.

[62] Lucas, B. D., Kanade, T. "An iterative image registration technique with an application to stereo vision". In IJCAI Vol. 81, pp. 674-679, August 1981.

[63] LUCAS, Bruce David, "Generalized image matching by the method of differences", Doctoral Dissertation, generalized Image Matching by the Method of differences, pp.213-234, Camegie Mellon University, Pittsburgh, PA,USA, 1985.

[64] O. Cakmakci and F. Berard. "An Augmented Reality Based Learning Assistant for Electric Bass Guitar,"

In Proceeding of 10th International Conference on Human-Computer Interaction, 2003

[65] M. Fiala, “ARTag, a fiducial marker system using digital techniques,” in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05), vol. 2, pp. 590–596, San Diego, Calif, USA, 2005.

[66] M. Isard and A. Blake, “CONDENSATION—conditional density propagation for visual tracking,” International Journal of Computer Vision, vol. 29, no. 1, pp. 5–28, 1998

[67] Kerdvibulvech, Chutisant, and Hideo Saito. "Markerless guitarist fingertip detection using a bayesian classifier and a template matching for supporting guitarists." Proceedings of the 10th ACM/IEEE Virtual Reality International Conference, VRIC ‘08, Laval, France. 2008.

[68] Manitsaris, S., Tsagaris, A., Dimitropoulos, K., & Manitsaris, A.. Finger musical gesture recognition in 3D space without any tangible instrument for performing arts. International Journal of Arts and Technology, 8(1), 11-29, 2015.

[69] Aggarwal, A., Kumar, R., Sahay, T., & Chandra, M.. GuiTones-I: An audio-visual database of monophonic guitar tones. In Region 10 Conference (TENCON), 2016 IEEE (pp. 497-500). IEEE, 2016.

[70] Perez-Carrillo, A., Arcos, J. L., & Wanderley, M. Estimation of guitar fingering and plucking controls based on multimodal analysis of motion, audio and musical score. In International Symposium on Computer Music Multidisciplinary Research (pp. 71-87). Springer, Cham, 2015.

[71] Traube, C., & Smith, J. O. Extracting the fingering and the plucking points on a guitar string from a recording. In Applications of Signal Processing to Audio and Acoustics, 2001 IEEE Workshop on the (pp. 7-10). IEEE, 2001.

[72] Yilmaz, Alper, Omar Javed, and Mubarak Shah. "Object tracking: A survey." Acm computing surveys (CSUR) 38.4 (2006): 13, 2006.

[73] Padmavathi S. and Divya, S., “Survey on Tracking Algorithms”, International Journal of Engineering Research & Technology (IJERT), vol. 3, no. 2, pp. 830 - 834, 2014.

[74] Martin, Manu. "Hand Segmentation from RGB Images in Uncontrolled Indoor Scenarios Using Randomized Decision Forests." 2017.

[75] Vodopivec, Tadej, Vincent Lepetit, and Peter Peer. "Fine hand segmentation using convolutional neural networks." arXiv preprint arXiv:1608.07454, 2016.

[76] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” ACM Transactions on Intelligent Systems and Technology, vol. 2, pp. 27:1–27:27, 2011

[77] http://fourier.eng.hmc.edu/e161/lectures/fourier/node11.html

[78] http://www.robots.ox.ac.uk/~sjrob/Teaching/SP/l7.pdf

[79] Pirsiavash, Hamed, Carl Vondrick, and Antonio Torralba. "Assessing the quality of actions." European Conference on Computer Vision. Springer, Cham, 2014.

[80] Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines. ACM Transactions on Intelligent

Systems and Technology (TIST), 2011.

[81] Drucker, H., Burges, C.J., Kaufman, L., Smola, A., Vapnik, V.: Support vector regression machines. NIPS, 1997.

[82] Zhao, WANG, Jun OHYA. “An Accurate and Robust Algorithm for Tracking Guitar Neck in 3D Based on Modified RANSAC Homography”In Electronic Imaging 2018, IRIACV-204, Burlingame, USA, 2018.

[83] Zhao, WANG., and Jun OHYA. “Detecting and Tracking the Guitar Neck Towards the Actualization of a Guitar Teaching-aid System”. International conference on advanced mechatronics: toward evolutionary fusion of IT and mechatronics ICAM2015, No. 6, pp. 187-188. Tokyo, Japan, 2015.

[84] Zhao, WANG. “Research on Detecting and Tracking the Guitar Neck and Guitarist’s Fingers from a Video Sequence” Master Thesis of Waseda University, 2015.

[85] Martin A. Fischler & Robert C. Bolles . "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography" (PDF). Comm. ACM. 24 (6): 381–395, 1981.

[86] Y. Zhou, G. Jiang, and Y. Lin. A novel finger and hand pose estimation technique for real-time hand gesture recognition. Pattern Recognition, 49:102–114, 2016.

[87] Jaroensri, Ronnachai, Amy Zhao, Guha Balakrishnan, Derek Lo, Jeremy D. Schmahmann, Frédo Durand, and John Guttag. "A Video-Based Method for Automatically Rating Ataxia." In Machine Learning for Healthcare Conference, pp. 204-216. 2017.

[88] Poppe, Ronald. "A survey on vision-based human action recognition." Image and vision computing 28, no. 6: 976-990. 2017

[89] Gordon, A.S.: Automated video assessment of human performance. In: AI-ED. (1995) 2, 1995.

[90] Jug, M., Perˇs, J., Deˇzman, B., Kovaˇciˇc, S.: Trajectory based assessment of coordinated human activity. Springer (2003) 3, 2003.

[91] Perˇse, M., Kristan, M., Perˇs, J., Kovacic, S.: Automatic Evaluation of Organized Basketball Activity using Bayesian Networks. Citeseer, 2007.

[92] Prokopowicz, P., Cooper, P. The Dynamic Retina: Contrast and Motion Detection for Active Vision (Technical Report No. 38). The Institute for the Learning Sciences, Northwestern University, 1993.

[93] Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatiotemporal features for action recognition with independent subspace analysis. In: CVPR., 2011.

[94] Z. Al-Taira, R. Rahmat, M. Saripan, and P. Sulaiman. Skin Segmentation Using YUV and RGB Color Spaces. Journal of Information Processing Systems, 10(2):283–299, 2014.

[95] M. Kawulok, J. Nalepa, and J. Kawulok. Skin Detection and Segmentation in Color Images, 2015.

[96] Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation." arXiv preprint arXiv:1511.00561, 2015.

[97] Long, J., Shelhamer, E., & Darrell, T. “Fully convolutional networks for semantic segmentation”. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR) pp. 3431-3440, 2015.

[98] Mazur, T. R., Fischer-Valuck, B. W., Wang, Y., Yang, D., Mutic, S., & Li, H. H. (2016). SIFT-based dense pixel tracking on 0.35 T cine-MR images acquired during image-guided radiation therapy with application to gating optimization. Medical physics, 43(1), 279-293, 2016.

[99] Shi, Jianbo. "Good features to track." In Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on, pp. 593-600. IEEE, 1994.

[100] Ha, Seok-Wun, and Yong-Ho Moon. "Multiple object tracking using SIFT features and location matching." International Journal of Smart Home 5, no. 4 (2011): 17-26, 2011.

[101] Bay, H., Tuytelaars, T. and Gool, L. V., ―SURF: Speeded Up Robust Features‖, Proc. Of the ninth European Conference on Computer Vision, 2006.

[102] Greg Welch, Gary Bishop,” An introduction to the Kalman Filter”, In University of North Carolina at Chapel Hill, Department of Computer Science. Tech. Rep. 95-041, 2006

[103] Parekh, Himani S., Darshak G. Thakore, and Udesang K. Jaliya. "A survey on object detection and tracking methods." International Journal of Innovative Research in Computer and Communication Engineering 2, no. 2 (2014): 2970-2979, 2014.

[104] J.Joshan Athanesious, P.Suresh, “Systematic Survey on Object Tracking Methods in Video”,International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 242-247, 2012.

[105] Joshan Athanesious J, Suresh P.” Implementation and Comparison of Kernel and Silhouette Based Object Tracking”, International Journal of Advanced Research in Computer Engineering & Technology, pp 1298-130, 2013.

[106] Saravanakumar, S.; Vadivel, A.; Saneem Ahmed, C.G., "Multiple human object tracking using background subtraction and shadow removal techniques," Signal and Image Processing (ICSIP), 2010 International Conference on , vol., no., pp.79,84, 15-17, 2010.

[107] Wang, Lijun, et al. "Visual tracking with fully convolutional networks." Proceedings of the IEEE International Conference on Computer Vision, 2015.

[108] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[109] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.

[110] Lenet, B. J., Komorowski, R., Wu, X. Y., Huang, J., Grad, H., Lawrence, H. P., & Friedman, S.. Antimicrobial substantivity of bovine root dentin exposed to different chlorhexidine delivery

vehicles. Journal of endodontics, 26(11), 652-655, 2000.

[111] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” CoRR, vol. abs/1502.03167, 2015.

[112] Zhou, Xingyi, et al. "Model-based deep hand pose estimation." arXiv preprint arXiv:1606.06854 2016.

[113] Guo, Hengkai, Guijin Wang, Xinghao Chen, Cairong Zhang, Fei Qiao, and Huazhong Yang. "Region Ensemble Network: Improving Convolutional Network for Hand Pose Estimation." arXiv preprint arXiv:1702.02447, 2017.

[114] “Student's *t*-test”, https://en.wikipedia.org/wiki/Student%27s_t-test

[115] Zhao, WANG, Jun OHYA. “A Fingertips Tracking Algorithm for Guitarist Based on CNN-Segmentation and Extended Particle Filter”, Journal of Imaging Science and Technology ,(Under Submitting)

[116] Zhao, WANG, Jun OHYA. “Tracking the Guitarist’s Fingers as Well as Recognizing Pressed Chords from a Video Sequence”. In Electronic Imaging 2016, 2016(15), pp.1-6 San Francisco, USA, 2016.

[117] Zhao, WANG, Jun OHYA. “Fingertips Tracking Algorithm for Guitarist Based on Temporal Grouping and Pattern Analysis”. In Asian Conference on Computer Vision (ACCV), pp. 212-226. Taipei, Taiwan, 2016.

[118] Zhao, WANG, Jun OHYA. “A 3D guitar fingering assessing system based on CNN-hand pose estimation and SVR-assessment”, In Electronic Imaging 2018, IRIACV-204, Burlingame, USA, 2018.

[119] Andrieu, C., Doucet, A., and Holenstein, R. “Particle Markov chain Monte Carlo methods”. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72(3):269–342, 2010.

[120] Bacon, Tony. “The Ultimate Guitar Book”. Alfred A. Knopf. ISBN 0375700900, 1997

[121] https://en.wikipedia.org/wiki/Guitar

[122] Christopher Perez, “Evaluating Guitar Performance” 2016: https://nafme.org/evaluating-guitar-performance-using-comprehensive-assessment-students/

[123] Shigeki Sugano, “kenban gakki enso robotto ni kansuru kenkyu”, Doctor Thesis of Waseda University, 32689-795, 1989

[124] Radisavljevic, Aleksander, and Peter F. Driessen. "Path Difference Learning for Guitar Fingering Problem." In ICMC, vol. 28, 2004.

[125] Yonebayashi, Yuichiro, Hirokazu Kameoka, and Shigeki Sagayama. "Automatic Decision of Piano Fingering Based on a Hidden Markov Models." In IJCAI, pp. 2915-2921, 2007.

Publications

Academic Paper (International Conferences)

○1. Zhao, WANG, Jun OHYA. “Tracking the Guitarist’s Fingers as Well as Recognizing Pressed Chords from a Video Sequence”. In Electronic Imaging 2016, 2016(15), pp.1-6 San Francisco, USA.

○2. Zhao, WANG, Jun OHYA. “Fingertips Tracking Algorithm for Guitarist Based on Temporal Grouping and Pattern Analysis”. In Asian Conference on Computer Vision (ACCV), pp. 212-226. Taipei, Taiwan, Oct, 2016

○3. Zhao, WANG, Jun OHYA. “A 3D guitar fingering assessing system based on CNN-hand pose estimation and SVR-assessment”, In Electronic Imaging 2018, IRIACV-204, Burlingame, USA

○4. Zhao, WANG, Jun OHYA. “An Accurate and Robust Algorithm for Tracking Guitar Neck in 3D Based on Modified RANSAC Homography”In Electronic Imaging 2018, IRIACV-204, Burlingame, USA

○5. Zhao, WANG., and Jun OHYA. “Detecting and Tracking the Guitar Neck Towards the Actualization of a Guitar Teaching-aid System”. International conference on advanced mechatronics: toward evolutionary fusion of IT and mechatronics ICAM2015, No. 6, pp. 187-188. Tokyo, Japan, 2015

6. Gao, S., Tatematsu, N., Ohya, J., & Wang, Z. Estimating Clean-up Robots’ Mechanical Operations of Objects Using a SLAM Based Method. In The… international conference on advanced mechatronics: toward evolutionary fusion of IT and mechatronics: ICAM: abstracts Vol. 2015, No. 6, pp. 249-250. Tokyo, Japan,2015

7. Keishi Nishikawa, Zhao Wang, Jun Ohya, Takashi Matsuzawa, Kenji Hashimoto, and Atsuo Takanishi ,“Automatic, Accurate Estimation of the position and pose of a Ladder in 3D Point Cloud”, The 5th IIEEJ International Workshop on Image Electronics and Visual Computing, 5-2C, Da Nang, Vietnam March, 2017

Academic Paper (Domestic Conferences)

8. Zhao WANG, Ye LI, Jing YAN, Jun OHYA, “Study of Detecting the Frets and Strings on the Neck of the Guitar from RGBD Images towards the Actualization of an Autonomous Guitar Teaching System”, Section D7,Media Computing Conference 2014

9. Zhao WANG, Jun OHYA, “Study of a Vision Based Method for Checking the Position of Each Finger of Guitar Players - Towards the Actualization of an Autonomous Guitar Chord Teaching System -”, Section D-11-7, IEICE Society Conference 2015.

10. Zhao WANG, Jun OHYA, “A Method for Tracking Guitar Neck and Fingertips: Necking Tracking Robust against Occlusions Based on Geometry Analysis and Fingertips Tracking Based on Temporal Probability Map-”, FIT Information and Science Technology Forum 2015.

11. 本田 浩暉, 王 釗, 大谷 淳, 透視変換を用いたギター演奏時のネックの動画像における追跡法の検討, 画像電子学会第 280 回研究会, 03/2017

12. 前田 尚俊, 王 釗, 大谷 淳, Support Vector Regression に基づく 3 次元動画像処理による人物の動作評価法の検討, 画像電子学会第 280 回研究会, 03/2017

13. Zelin Zhang, Zhao Wang, Jun Ohya. “Hand Pose Estimation from Single Depth Images with 3D Convolutional Neural Network”, in IEICE PRMU2017-140, vol. 117, no. 391, pp. 271-276, 01/2018.