©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.15.IPAS-047

Tracking the Guitarist's Fingers as Well as Recognizing Pressed Chords from a Video Sequence

Zhao WANG and Jun OHYA
Department of Modern Mechanical Engineering, Waseda University, Tokyo, Japan

Abstract

Towards the actualization of an autonomous guitar teaching system, this paper proposes the following two video analysis based methods: (1) pressed chord recognition and (2) fingertip tracking. For (1), an algorithm that can extract finger contours and chord changes is proposed so that the chords pressed by the guitar player are recognized. For (2), an algorithm that can track the fingertips by continuously monitoring the appearance and disappearance of the regions of fingertip candidates is proposed. Experimental results demonstrate that the proposed two modules are robust enough under complex contexts such as complicated background and different illumination conditions. Promising results were obtained for accurate tracking of fingertips and for accurate recognition of pressed chords.

1. Introduction

Learning how to play the guitar is very interesting, but difficult, because it involves many techniques: reading scores, pressing strings, sweeping, slipping and so forth. Going to a guitar teacher is one of the most useful and effective methods to learn, but it is very difficult to go very frequently such as every day. However, if a computer at home can autonomously teach how to play the guitar, it is very convenient, and the progress could be very fast. Such an automatic guitar teaching system should be able to teach many techniques such as the ones mentioned above. Among these techniques, this paper focuses on two fundamental but significant techniques: specifically, fingertip tracking and pressed chord recognition. Acquiring proper movements of fingers is very important for guitar skill progress; therefore, the teaching system should be able to evaluate how well the fingers move. For this, results of tracking the fingers could be useful. Pressed chord recognition is extremely helpful for novice guitar players; in addition, even for advanced level guitarists, it helps efficiently remember chord changes during early stage of practicing that tune.

Motokawa et al. [1] built a system called Online Guitar Tracking that supports a guitarist using augmented reality to detect the guitar so that the player can learn how to hold the strings of the guitar by overlapping the player's hand onto a manual model. Scarr et al. [2] proposed 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. Burns et al. [3] detected the positions of fingertips for the retrieval of the guitarist fingering without markers. By fixing a camera on the guitar neck, the guitar neck and the camera are relatively static. Kerdvibulvech et al. [4] proposed a novel approach to detect the position of the player's fingers in 3D. Stereo cameras are used to compute the 3D positions of fingers using the color marker attached on fingertips. It is obvious that some research works studied computer vision-based methods for supporting guitarists to improve their skills. However, there are also limitations in their works: (1) some of them [1][3][4] use inconvenient tool such as color markers, neck-fixed camera and ARTag; (2) some of them did not track the whole area of the guitar neck and fingers[1][2][3], which means it is very hard to recognize which area on the fretboard belongs to which fret and string. It is very hard to analyze the fingering without tracking the guitar neck and fingers. The evaluation of guitar fingering consists of several aspects: a. whether the player presses the right place of fretboard; b. whether the player uses the right finger; c. whether the hand of the player moves in an effective manner. In order to analyze fingering (at least 3 aspects above), instead of detecting fingers on certain frames, fingers should be tracked on every frame of the input image sequence.

Figure 1 outlines the two modules proposed by this paper. As can be seen in Fig. 1, the proposed two modules use the results of performing our guitar neck tracking method [7] for the input video sequence. The pressed chord recognition module extracts finger contours in the frames in which the guitarist changes the chords so that the pressed chords are recognized. The fingertip tracking module tracks the fingertips of the guitarist despite complex and volatile changes in hand gestures by continuously monitoring the appearance and disappearance of the regions of fingertips candidates frame by frame.

In this paper, Section 2 covers the chord recognition module; Section 3 explains the fingertip tracking module; Section 4 describes the experimental results, and analyzes the accuracy of each system. Section 5 concludes this paper.

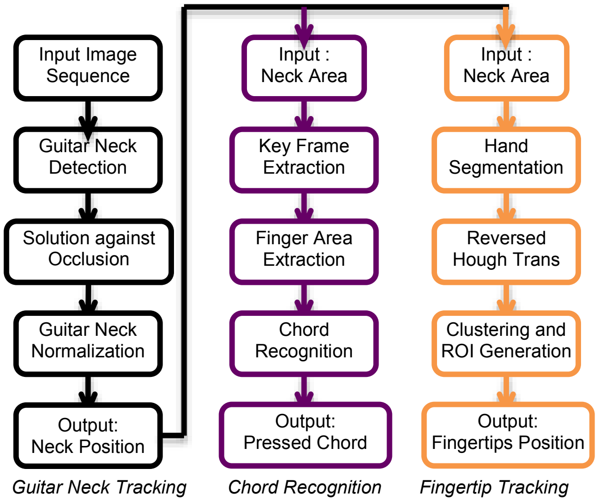


Figure 1 Outline of Proposed Research

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.15.IPAS-047

2. Pressed Chord Recognition Module

One of the first techniques novice guitar players learn may be how to press chords, as a chord is an aggregate of music pitches pressed simultaneously on fretboard. This module checks whether the guitarist presses the proper places on the fretboard of the guitar. As described Sec. 1, this module is useful also for advanced level guitarists.

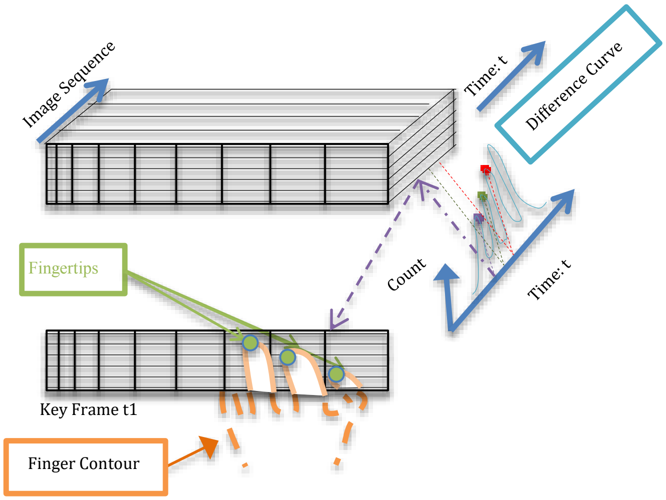


Figure 2 Conceptual Image of Guitar Chord Recognition

2.1 Key Frame Extraction

Instead of processing every frame the player played, this module focuses on the frames in which the guitar player changes his or her hand shape so as to change the current chord to the next chord. Such frames are defined as Key Frames. In other words, during the time guitar players play the guitar, they change their hand shapes to press the chord only in Key Frames. In frames that do not correspond to Key Frames, the player tends to keep his hand shape. Therefore, in general, in an image sequence, Key Frames could own more "different pixels" than non-Key Frames, where Different pixel is defined as a pixel that satisfies Eq. (1).

$$\left|\left[R_n(x,y) - R_{n+1}(x,y)\right] + \left[G_n(x,y) - G_{n+1}(x,y)\right] + \left[B_n(x,y) - B_{n+1}(x,y)\right]\right| > Threshold \qquad (1)$$

In Eq.(1), n and n+1 indicate consecutive frame ID numbers, respectively; (x,y) indicate the position of a pixel in the image; R,G,B indicate red, green, blue values of the pixel, respectively.

In every frame, every pixel is tested whether it is a Different Pixel, by comparing its RGB values with those of the pixel with the same x and y coordinates in the previous frame, using Eq.(1). Then, the count of Different Pixels in each frame forms a Difference Curve, as shown in Fig.3. Local peaks in the Difference Curve correspond to frames having more Different Pixels than the neighboring frames. In other words, the peaks correspond to "Key Frame", which are the time points at which chord changes take place.

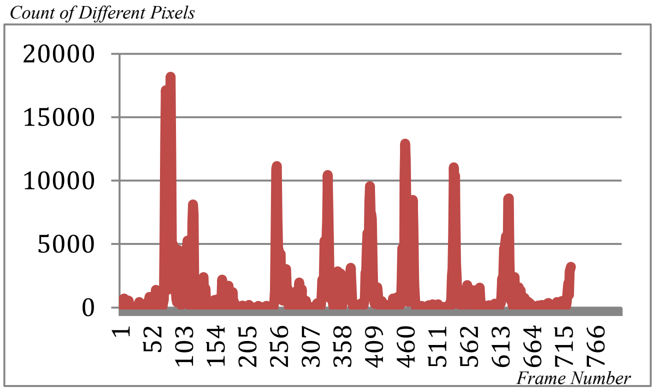


Figure 3 Difference Curve (Peaks indicate Key Frames)

2.2 Finger Area Detection

In key frames detected in Sec. 2.1, after some pre-processes (canny edge detection and dilation) are performed, finger contours are extracted 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 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), the 2D position of each fingertip could be detected in every Key Frame.

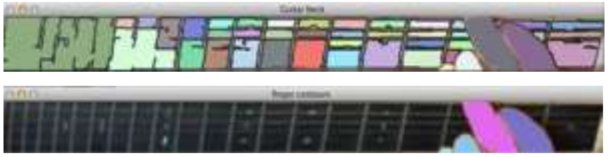


Figure 4 Result of Finger Area Detection

2.3 Chord Recognition

The guitar neck tracking process shown in Fig. 1 constantly projects the neck area to a new image frame by frame with resolution (1000*80), no matter how the player waves the guitar during his/her play [7]. An example of the projection result is shown in Fig.5. The historical technique to calculate the positions of each fret is called the rule of 18, and it involves successively dividing the scale length minus the offset to the previous fret by 18 [8]. It means if the guitar

Figure 5 Guitar Neck Tracking Result

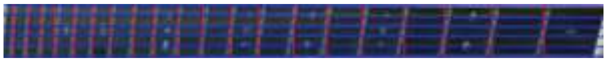


Figure 6 Detection Result of Fret (Red Line) and Strings (Blue Line)

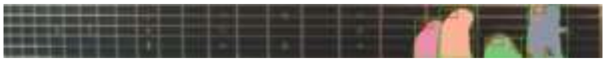


Figure 7 Pressed Chord Recognition Result

neck's length is given (From Fret 1 to Fret 20), the position of each fret can be easily calculated. Meanwhile, six strings are placed in a parallel manner at a uniform interval (between adjacent strings). An example of the result of detecting every fret and every string is shown in Fig.6.

The process explained in Section 2.2 can extract fingertips by evaluating the four criteria. Figure 7 shows an example of finger and fingertip extraction. Based on the 2D positions of the normalized guitar neck, the detected frets and strings as well as the 2D position of each fingertip, we can recognize the chord pressed by the fingers. For instance, in Fig.7, four fingertips press Fret 1 String 6, Fret 2 String 2, Fret 3 String 5 and Fret 3 String 4, respectively; thus, we can recognize that the player is pressing Bm Chord.

3. Fingertips Tracking System

In this tracking module for guitarist fingertips, after inputting the projection result of tracking guitar neck (the result of Guitar Neck Tracking), first, a machine learning-based Bayesian Pixel Classifier is used to segment the hand area on the test data.

Then, the probability map of fingertips is generated on segmentation results by counting the voting numbers of the Reversed Hough Transform. Furthermore, a clustering algorithm, which is a geometry analysis for buffer images (10 adjacent frames), is applied to removal noises. Finally, a K-means-based categorization algorithm is utilized to discriminate 4 fingers (index finger, middle finger, ring finger, little finger).

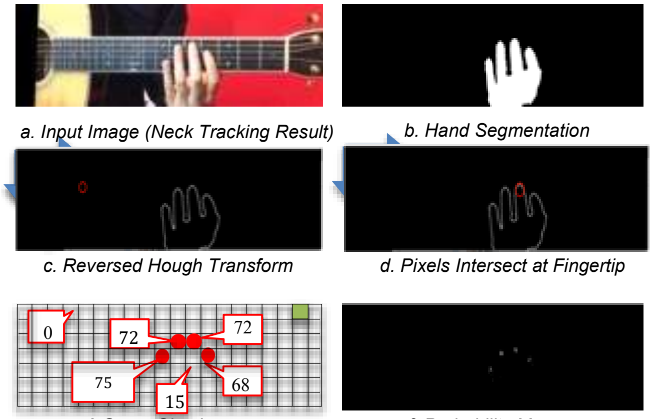
In this Fingertips Tracking Module, the algorithm of tracking guitar neck) is still applied in this system, and the tracking result of the guitar neck is still the input to this Fingertips Tracking System. The only difference is, in this system, the guitar neck is projected to a new image with the resolution 1300*300, not as same as the projection result in Sec. 2. The Guitar neck is still in the middle of the projection result.

3.1 Reversed Hough Transform

In order to cope with hand segmentation problem effectively under different illuminations and backgrounds, we use a Bayesian Classifier to separate skin-color pixels from non-skin-color pixels. During the off-line learning process, hand areas are manually extracted from the training data set so that only the hand areas in their original colors are obtained, where non-hand areas are obtained in black color. Using the training data set and manual extraction of the hand area, a-prior probabilities P(s) and P(c) and a conditional probability P(c|s) are obtained, where s indicates being skin pixel, c indicates the occurrence of each color in training data set. During the on-line testing process, every pixel in a testing image is calculated the a-posteriori probability to be the hand-color pixel.

In general, fingertips have semi-circle shape; therefore, fingertips' contours could intersect with circles more frequently than non-fingertip contours (Fig.8b). After segmenting the hand area (Fig.8b), to test the probability that a pixel in an image is the center of a set of circles, Reversed Hough Transform, which is proposed by the authors, is performed (Fig.8c). As Fig.8c shows, every pixel in the hand segmentation result is scanned (red arrows in Fig.8c), and at every pixel a set of circles (radius varies from 15 to 25 pixels) are yielded to test how many pixels of the circle intersect with the hand contours. For example, if the set of circles is moved to the position such as the one shown in Fig.8d, the circle intersects with the hand contours at many pixels.

Figure 8e shows how to save the count of intersections. The counts of intersected pixels are saved in an image that has the same size with the Hand Segmentation result. In Fig.8e, each accumulated cell of our Reversed Hough Transform for instance, the green cell in Fig.8e indicates every single pixel of the hand segmentation result.



a. Input Image (Neck Tracking Result)
b. Hand Segmentation
c. Reversed Hough Transform
d. Pixels Intersect at Fingertip
e. A Same Size Image Recording the count of the Intersection
f. Probability Map (Brighter Pixels Indicate High Probability)

Figure 8 Reversed Hough Transform

The red points indicate the positions of the four fingertips. At the position of every rectangle in Fig.8e, it records the maximal count of intersections when a set of circles is yielded at this pixel, which corresponds to the center of the set of circles. From Fig.8e, it is clear that the pixels around fingertips own higher counts than other pixels, because, as mentioned earlier, fingertips have semi-circle shape, and fingertips contours have more intersection pixels with circles.

Then, using the largest count in every frame, each pixel count is normalized by:

$$\text{Count}_{\text{normalized}} = \text{Count} \times \frac{255}{\text{Max}} \qquad (2)$$

where, Max indicates the largest count in this frame.

By using Eq.(2), the probability map shown in Fig.7f is generated, where the bright and dark pixels indicate high and low probabilities, respectively.

3.2 Clustering and ROI Generation

The probability map cannot always yield good results for accurate location of fingertips. Sometimes noises also own higher probability, as shown in Fig.9. In order to remove noises from the probability map, we carry out a clustering algorithm [6] by focusing on Image Buffer [5] (10 previous frames) in every single frame. Compared with real fingertips, "life span" of the noise is shorter, because the noise is mainly caused by curved joints of the player's hand (as a curved joint generates circle-like shape while our reversed Hough Transform aims at detecting the fingertips with circle-like shapes). Figure 9 visualizes the clustering algorithm. Bright areas in the left side indicate the pixels that have higher probabilities to be fingertips, while bright areas could contain noises also. In the consecutive frames (right side of Fig.9), the same color circles are grouped into a cluster, because in time series, (from Frame t to Frame t-2), these grouped circles have very short Euclid distance in the buffer. On the other hand, if contours are contaminated by noise such as the red circle in the left side, no circles can be grouped with it even though it has a high probability.

DOI: 10.2352/ISSN.2470-1173.2016.15.IPAS-047

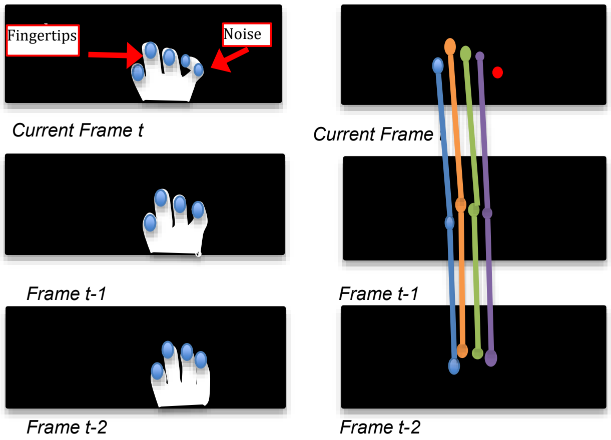


Figure 9 Clustering

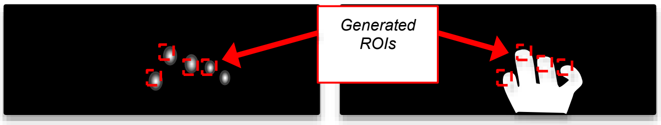


Figure 10 ROIs Generation（Only at the position of Fingertips）

That is, the noise is not grouped over multiple frames; i.e. the "life span" of a noise is short. The noise can be detected and removed, by checking whether the "life span" is short.[5]

Then, only at the positions of clustered circles, we generate ROIs [5], as shown in Fig.10. Actually the generated ROI contains consecutive clustered fingertips over time-series. Once ROIs are generated, we assume that among its previous 10 frames, we find consecutive fingertips, which also indicates that by comparing the contours (fingertips) in ROIs in the current frame and previous frame, we associate the ROIs in the current frame with the ROIs in the previous frame. By applying the clustering algorithm and the ROI's association frame by frame, fingertips are tracked.

4. Experimental Results and Discussion

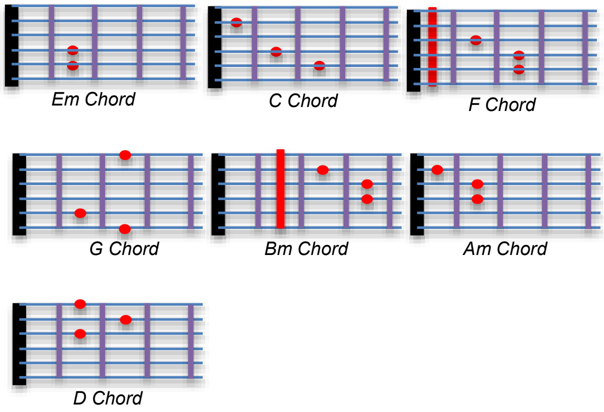


Figure 11 Seven Chords for Evaluating Our System

The system we used for testing was a Macbook Pro released in 2013 with a 2.7 GHz Intel Core i7 processor and DDR3 16GB memory. The camera was Iphone 5S front camera with resolution 1136*640. The videos were taken under the different illuminations with front-facing scenarios. All the algorithms were implemented in Xcode 6.4 with C++ and OpenCV 2.4.3 library.

4.1 Pressed Chord Recognition

We collected 30 videos using six subjects to test the validity of the guitar chord recognition system, where each video includes seven chords (C, Em, F, G, Bm, Am, D) played as Fig.11 shows. In the test data, there are total 210 chords (30 videos times 7 Chord per video). Examples of recognition results are shown in Fig.11. The confusion matrix is shown in Table 1. Table 1 shows that the accuracy of the guitar chord recognition rate is 93.8%. The average processing time for each frame is 0.018 seconds.

Table 1 Confusion Matrix of Chord Recognition Result

	C	Em	F	G	Bm	Am	D	O
C	28							2
Em		30						
F			27					3
G				24				6
Bm					28			2
Am						30		
D							29	1

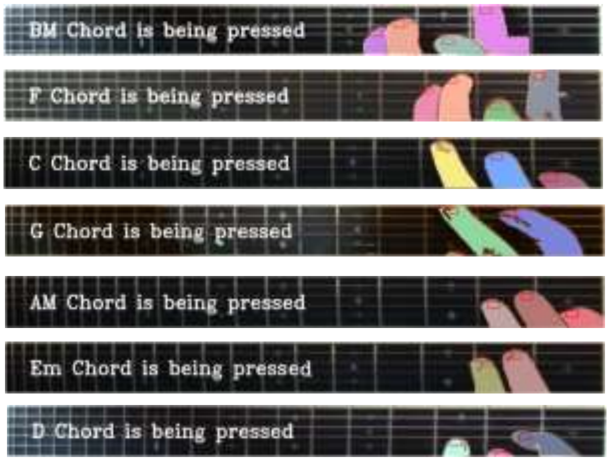
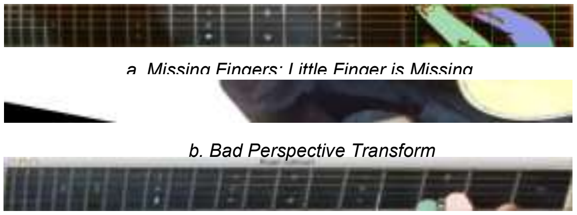


Figure 12 The Examples of Chord Recognition Result



a. Missing Fingers: Little Finger is Missing

b. Bad Perspective Transform

c. False Key Frame Extraction

Figure 13 Reasons for False Recognition

DOI: 10.2352/ISSN.2470-1173.2016.15.IPAS-047

The false recognition results are caused by the following three reasons: (1) missing fingers; (2) bad perspective transform; (3) false key frame extraction result as shown in Fig.13 respectively.

4.2 Fingertip Tracking

In this experiment, we invited six subjects and took four videos for each subject. The subjects who learned the guitar before played whatever they can (scales, half-scale, chord pressing, finger style). On the other hand, the novices played simple things that almost every beginner would practice (half-scale practice, the chords easy to be pressed) according to oral instruction. The duration of each video lasts approximately 20 seconds to 30 seconds

Totally, 24 videos (3704 frames) were used to test the validity of the fingertips tracking. Actually, the fingertips tracking is the most challenging part in this paper, because during guitar playing, sometimes the shape of hands changes so fast (Fig.15.a), sometimes two fingers jointed together (Fig.15.b), sometimes due to occlusion fingertips cannot be seen (Fig.15.d) and so forth. Another difficult problem is that the system has to track four fingers at the same time while four fingers share same feature: they have almost same shape and color. Figure 14 shows the trajectories of tracking the four fingertips for an image sequence with 214 consecutive frames. The ground truth is manually observed by human eye, while the tracking result is obtained by the fingertips tracking. Table 2 shows the mean error of each finger in all 24 videos (total 3704 frames). With the mean error 6.16, 5.65, 7.4, 10.2 pixels for little finger, ring finger, middle finger and fore finger respectively, it is obvious that the system is robust enough for tracking the fingertip of the guitarist during guitar playing. The average processing time for each frame is 0.45 seconds.

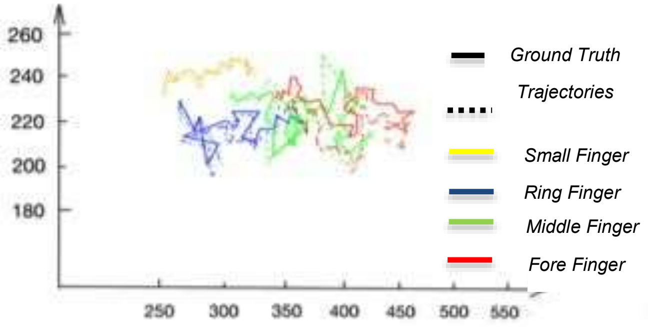
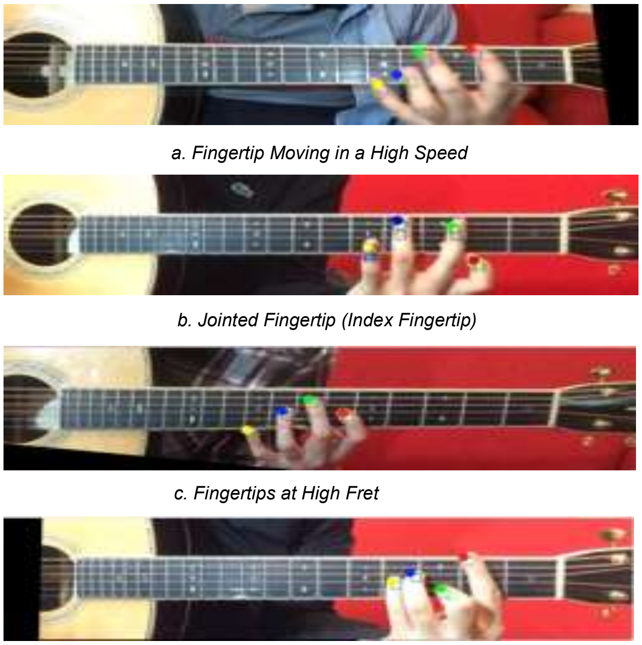


Figure 14 Trajectories of Four Fingertips for an image Sequence

Table 2 Mean Error of Fingertips Tracking for All The Test

	Little Finger	Ring Finger	Middle Finger	Fore Finger
Mean Error (Pixel)	6.15	5.65	7.4	10.2



a. Fingertip Moving in a High Speed

b. Jointed Fingertip (Index Fingertip)

c. Fingertips at High Fret

d. Curved Middle Finger (Invisible Middle Fingertip)

Figure 15 The Examples of Fingertips Tracking
The Colorful Point Indicates Tracking Result of Individual Fingertip
Yellow: Small Finger Blue: Ring Finger
Green: Mid Finger Red: Index Finger

5. Conclusion

Towards the actualization of an autonomous guitar teaching system, this paper has proposed the following two video analysis based methods: (1) pressed chord recognition and (2) fingertip tracking.

For (1), an algorithm that can extract finger contours and chord changes is proposed so that the chords pressed by the guitar player are recognized. First, the tracking algorithm of the guitar neck is applied to the input image sequence. Then, finger areas are extracted by evaluating several criteria. Finally, based on the position information of the guitar neck and the located fingertips, the chord, which is pressed by the player of guitar, is recognized.

For (2), an algorithm that can track the fingertips by continuously monitoring the appearance and disappearance of the regions of fingertips candidates is proposed. First, it uses the same guitar neck tracking algorithm as guitar chord recognition. Then Bayesian pixel classifier is applied to pixels in order to obtain the hand segmentation result. The probability map of the fingertip is generated from the segmentation results obtained by the reversed Hough Transform we propose. A clustering algorithm is applied to consecutive multiple frames so as to remove noises in the probability map. Finally, a ROIs generation algorithm is used to discriminate the fingertips from the noise, and by monitoring the association of ROIs in consecutive frames, the fingertips can be tracked.

Experimental results for (1) and (2) are summarized as follows.

DOI: 10.2352/ISSN.2470-1173.2016.15.IPAS-047

(1) As a result of processing 30 videos that contain 210 chords, it turns out that a recognition rate of 93.8% is achieved.

(2) 24 videos that contain six subjects' guitar play are processed. Mean errors of 6.16, 5.65, 7.4, 10.2 pixels for little finger, ring finger, middle finger and fore finger, respectively, are achieved. It is clear that this module is robust enough for tracking the fingertips of guitarists.

However, the system also has some limitations: (1) as mentioned before, both the chord recognition module and the fingertip tracking module could not work on real-time; (2) the angle of the camera could only be placed at front-oriented to prevent tracking failure; (3) once the guitar neck is moved out of the frame, tracking failure happens.

The future work includes: a. the algorithm optimization for real-time processing of the fingertip tracking module; b. the fingering evaluation module.

References

[1] 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'06)-Volume 00. IEEE Computer Society, pp. 243–244, 2006.

[2] Joseph Scarr, 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, Queenstown New Zealand, 8-9 Nov. 2010.

[3] 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, Paris, France, 2006.

[4] 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'07), Bangkok, Thailand, pages 147-152, January 2007. Oral (Full Paper)

[5] 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), Hindawi Publishing Corporation, ISSN 1687-5893, 10 pages, 2008.

[6] W. Ng, J. Li, S. Godsill, and J. Vermaak. "A hybrid approach for online joint detection and tracking for multiple targets," In Proceeding of IEEE Aerospace Conferences, pages 2126-2141, 2005.

[7] Zhao WANG, Jun OHYA, "Detecting and Tracking the Guitar Neck Towards the Actualization of a Guitar Teaching-aid System", JSME Robotics and Mechatronics Division, the 6th International Conference on Advanced Mechatronics ICAM 2015, 1P2-05, Tokyo, Japan.

[8] Calculating Fret Positions: http://liutaiomottola.com/formulae/fret.htm

Author Biography

WANG Zhao is now a current PhD candidate in the department of MME (Modern Mechanical Engineering) in Waseda University. He got the Bachelor Degree in Sun Yet-sun University in China (2010), and Master Degree in Waseda University (2015). Now he is mainly working on tracking algorithm of Computer Vision and Machine Learning.

Dr. Jun Ohya is a professor at the Department of Modern Mechanical Engineering, Waseda University, Japan. He earned his B.S., M.S., and Ph.D. degrees in Precision Machinery Engineering from the University of Tokyo in 1977, 1979, and 1988, respectively. In 1979, he entered NTT, Japan. From 1988, he stayed at the Univ. of Maryland, USA, for one year. In 1992, he transferred to ATR, Kyoto, Japan. In 2000, he joined Waseda University as a professor. In 2005, he was a guest professor at the Univ. of Karlsruhe, Germany. Dr. Ohya is a member of IEEE, IEICE, the Information Processing Society of Japan, ect. His research fields include image processing, computer vision, virtual reality, multimedia, pattern recognition.