Markerless Guitarist Fingertip Detection Using a Bayesian Classifier and a Template Matching For Supporting Guitarists

Chutisant Kerdvibulvech Hideo Saito

Department of Information and Computer Science, Keio University
3-14-1 Hiyoshi, Kohoku-ku 223-8522, Japan
{chutisant, saito}@ozawa.ics.keio.ac.jp

ABSTRACT

This paper proposes a vision-based application for recognizing the chord being played by a guitarist to help guitar learners to practice by themselves. In our previous work, color markers were used for detecting fingertips. In this paper, we present the preliminary results of markerless fingertip detection using a Bayesian classifier and a template matching algorithm. We segment the skin color of a guitar player's hand by using a Bayesian classifier. By using the online adaptation of color probabilities, this method is able to cope extremely well with illumination changes. We then use semicircle models for fitting curve to fingertip. Representative experimental results are also included. The method presented can be used to further develop guitar application to aid chord tracking for people learning to play the guitar.

Keywords: Guitarist Fingertip Detection, Bayesian Classifier, Adaptive Learning, Template Matching, Guitar Application

1. INTRODUCTION

Acoustic guitars are currently very popular and as a consequence, research about guitars is a very popular topic in the field of computer vision for musical applications.

Maki-Patola et al. [1] proposed a system called "Virtual Air Guitar" using computer vision. Their aim was to create a virtual air guitar which does not require a real guitar (e.g., by using only a pair of colored gloves), but produces music similar to a player using a real guitar. Liarokapis [2] proposed an augmented reality system for guitar learners. The aim of this work is to show the augmentation (e.g., the positions where the learner should place the fingers to play the correct chords) on an electric guitar as a guide for the player. Motokawa and Saito [3] built a system called "Online Guitar Tracking" that supports a guitarist using augmented reality. This is done by showing a virtual model of the fingers on a stringed guitar as a teaching aid for anyone learning how to play the guitar.

These systems do not aim to detect the fingering which a player is actually using (A pair of gloves are tracked in [1], and graphics information is overlaid on captured video in [2] and [3]). We have developed a different approach from most of these researches.

In our previous work [4] [5], we proposed a new guitar application for supporting guitar learners by employing computer vision aid. This was done by identifying whether the finger positions are correct and in accord with the finger positions required for the piece of music that is being played. In Fiala's work [6] an ARTag (Augmented Reality Tag) was used to detect the guitar position and to define world coordinate relative to guitar neck for dynamic camera calibration. Four colored fingertip markers were placed on the fingertips and used to calculate the 2D positions of fingertips. By utilizing a triangulation method on stereo cameras, the 3D positions of fingertip markers were recognized when a guitar string was pressed. The results of the guitar and fingertip markers detection were used to compute the guitar chord by applying PCA (Principal Component Analysis). An example of chord recognition system is represented in Fig. 1. The input images (above) are captured from two cameras. The output image (below) shows the detected 3D positions of four fingertip markers in the guitar coordinate system and the result of guitar chord recognition (E chord).

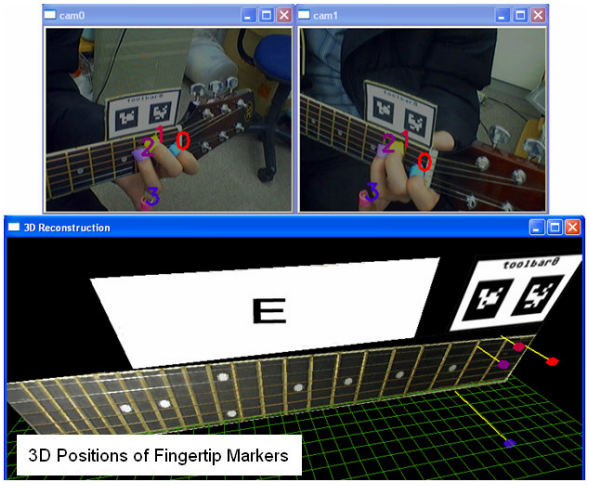

Figure 1: Our previous work for chord recognition by using colored fingertip markers

This can be developed into a guitar training application. An example user interface of this application is shown in Fig. 2. This application contains the lyrics,

guitar chord charts and voice information. Also, it recognizes if each successive chord used by the song, is being held correctly by the player. This would assist guitar learners because they would be able to automatically identify if they match the correct chords required by the musical piece. However, in this application, colored fingertip markers were required and this sometimes makes it unnatural for playing guitar in real life. We therefore overcame this problem by removing these fingertip markers. In this paper, we propose a markerless method for detecting the 2D positions of fingertips to further improve our guitar application.

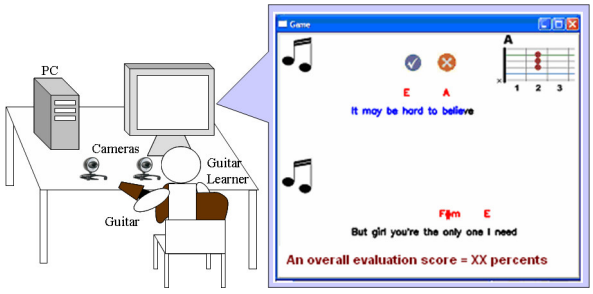

Figure 2: User interface of our guitar application

Our research goal is to accurately determine the 2D fingertip positions of a guitarist. Example input and output images are given in Fig. 3. A challenge for detecting the fingers of a guitar player is that, while playing the guitar, the fingers are not stretched out separately. So it is difficult to detect the fingertips correctly. Moreover, to detect the fingertips while playing the guitar, the background is usually dynamic and non-uniform (e.g., guitar neck and natural scene) which makes it more difficult to locate the fingertip positions. Also, in general classic guitar, colors of frets and strings are very similar to skin color, as a result it is challenging to segment the hand area from the background of guitar board correctly. Our method for detecting the fingertips of guitar player solves these problems.

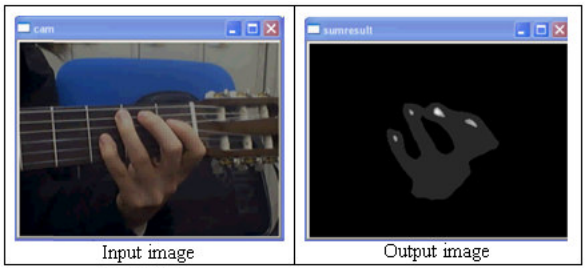

Figure 3: Sample detection without fingertip markers

At every frame, we firstly segment the hand from the background using a color detection algorithm. To determine the color probabilities of being skin color, during the pre-processing we apply a Bayesian classifier that is bootstrapped with a small set of training data and refined through an offline interactive training procedure [7] [8]. Online adaptation of skin probability is then used to refine the classifier using additional training images. Following this, we locate the fingertip positions by using a template matching algorithm. We crop the models of semicircle shape for a fit to the fingertip [9]. After superimposing the models on each candidate in the test image, we normalize the results and then find the proper threshold. As a final result, the system enables us to visually detect the fingertips even when the fingers are not fully stretched out or when the illumination changes.

2. RELATED WORKS

Related approaches for finger detection of guitarists will be described in this section. Cakmakci and Berard [10] detected the finger position by placing a small ARToolKit (Augmented Reality Toolkit) [11]'s marker on a fingertip of the player for tracking the forefinger position (only one fingertip). However, when we attempted to use the ARToolKit's markers to all four fingertips, some markers' planes were not simultaneously perpendicular to the optical axis of camera in some angles. Therefore, it was quite difficult to accurately detect the positions of four fingers concurrently by using the ARToolKit finger markers.

Burns and Wanderley [12] detected the positions of fingertips for the retrieval of the guitarist fingering without markers. They used the circular Hough transform to detect fingertips. However, from our experiments, utilizing Hough transform to detect the fingertips when playing the guitar is not robust enough. Also, they did not aim to deal with illumination changes during playing guitar in online process.

We therefore overcome this problem by attempting to segment the skin color of hand robustly. However, a well-known current problem of skin color detection is the control of the lighting. Changing the levels of light and limited contrasts prevent correct registration, especially when there is a cluttered background. The survey of detecting faces in images [13] provides an interesting overview of color detection. A major decision has to be made when deriving a model of color. This relates to the selection of the color space to be employed. Once a suitable color space has been selected, one of the commonly used approaches for defining what constitutes color can be used on the coordinates of the selected space. However, by using the simple threshold, it is very difficult to accurately classify the color when the illumination changes.

Therefore, we use a Bayesian classifier by learning color probabilities from a small training image set and then adaptively learn the color probabilities from online input images (proposed recently in [7] [8]). Applying this method, the first attractive property is that it avoids the burden involved in the process of manually generating a lot of training data. From small amount of training data, it adapts the probability according to the current illumination and converges to a proper value. For this reason, the major advantage of using this method is its ability to cope with changing illumination because it can adaptively describe the distribution of the skin color.

3. METHOD

Fig. 4 shows the schematic of the implementation. After capturing the images, we firstly apply a Bayesian classifier and an on-line adaptation of color probabilities for hand segmentation. As the next step, we apply a matching algorithm and then find the proper threshold to visually detect the positions of fingertips.

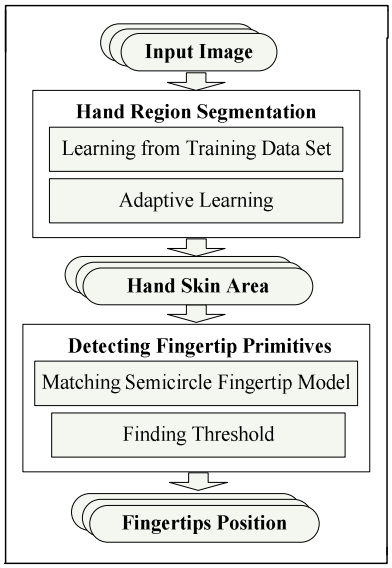

Figure 4: Method overview

3.1 Hand Region Segmentation

This section will explain the method we used for segmenting the hand region. We calculate the color probabilities being skin color which will be then used to segment the hand region. The learning process is composed of two phases. In the first phase, the color probability is learned from a small number of training images during an offline pre-process. In the second phase, we gradually update the probability from the additional training data images automatically and adaptively. The adapting process can be disabled as soon as the achieved training is deemed sufficient.

Therefore, this method allows us to get accurate color probability of the skin from only a small set of manually prepared training images. This is because the additional skin region does not need to be segmented manually. Also, because of the adaptive learning, it can be used robustly with changing illumination during the online operation.

3.1.1 Learning from Training Data Set

During an offline phase, a small set of training input images (20 images) is selected on which a human operator manually segments skin regions. The color representation used in this process is YUV 4:2:2 [14]. However, the Y-component of this representation is not employed for two reasons. Firstly, the Y-component corresponds to the illumination of an image pixel. By omitting this component, the developed classifier becomes less sensitive to illumination changes. Secondly, compared to a 3D color representation (YUV), a 2D color representation (UV) is lower in dimensions and, therefore, less demanding in terms of memory storage and processing costs.

Assuming that image pixels with coordinates *(x,y)* have color values $c = c(x,y)$, training data are used to calculate:

(i) The prior probability $P(s)$ of having skin s color in an image. This is the ratio of the skin-colored pixels in the training set to the total number of pixels of whole training images.

(ii) The prior probability $P(c)$ of the occurrence of each color in an image. This is computed as the ratio of the number of occurrences of each color c to the total number of image points in the training set.

(iii) The conditional probability $P(c|s)$ of a skin being color c. This is defined as the ratio of the number of occurrences of a color c within the skin-colored areas to the number of skin-colored image points in the training set.

By employing Bayes' rule, the probability $P(s|c)$ of a color c being a skin color can be computed by using

$$P(s \mid c) = \frac{P(c \mid s)P(s)}{P(c)} \qquad (1)$$

This equation determines the probability of a certain image pixel being skin-colored using a lookup table indexed with the pixel's color. The resultant probability map thresholds are then set to be $Threshold_{\max}$ and $Threshold_{\min}$, where all pixels with probability $\mathrm{P(s \mid c)} > Threshold_{\max}$ are considered as being skin-colored—these pixels constitute seeds of potential skin-colored blobs—and image pixels with probabilities $\mathrm{P(s \mid c)} > Threshold_{\min}$ where $Threshold_{\min} < Threshold_{\max}$ are the neighbors of skin-colored image pixels being recursively added to each color blob. The rationale behind this region growing operation is that an image pixel with relatively low probability of being skin-colored should be considered as a neighbor of an image pixel with high probability of being skin-colored. Indicative values for $Threshold_{\max}$ and $Threshold_{\min}$ are 0.5 and 0.15, respectively. A standard connected component labelling algorithm (i.e., depth-first search) is then responsible for assigning different labels to the image pixels of different blobs. Size filtering on the derived connected components is also performed to eliminate small isolated blobs that are attributed to noise and do not correspond to interesting skin-colored regions. Each of the remaining connected components corresponds to a skin-colored blob.

3.1.2 Adaptive Learning

The success of the skin-color detection depends crucially on whether or not the illumination conditions during the online operation of the detector are similar to

those during the acquisition of the training data set. Despite the fact that using the UV color representation model has certain illumination independent characteristics, the skin-color detector may produce poor results if the illumination conditions during online operation are considerably different to those used in the training set. Thus, a means for adapting the representation of skin-colored image pixels according to the recent history of detected colored pixels is required. To solve this problem, skin color detection maintains two sets of prior probabilities. The first set consists of *P(s)*, *P(c)*, *P(c|s)* that have been computed offline from the training set. The second is made up of $P_W(s)$, $P_W(c)$, $P_W(c \mid s)$ corresponding to *P(s)*, *P(c)*, *P(c|s)* that the system gathers during the *W* most recent frames respectively. Obviously, the second set better reflects the "recent" appearance of skin-colored objects and is therefore better adapted to the current illumination conditions. Skin color detection is then performed based on the following weighted moving average formula:

$$P_A(s \mid c) = \gamma P(s \mid c) + (1-\gamma) P_W(s \mid c) \qquad (2)$$

where γ is a sensitivity parameter that controls the influence of the training set in the detection process, $P_A(s \mid c)$ represents the adapted probability of a color *c* being a skin color, $P(s \mid c)$ and $P_W(s \mid c)$ are both given by Equation (2) but involve prior probabilities that have been computed from the whole training set [for $P(s \mid c)$] and from the detection results in the last *W* frames [for $P_W(s \mid c)$]. In our implementation, we set $\gamma = 0.8$ and $W = 5$.

Thus, the finger skin-color probability can be determined adaptively. By using online adaptation of finger skin-color probabilities, the classifier is easily able to cope with considerable illumination changes, and also it is able to segment the hand even in the case of a dynamic background.

3.2 Detecting Fingertip Primitives

In this section, we describe the method we used to detect the positions of the fingertips. We assume that the fingertip shape can be approximated with a semicircular shape while the rest of the hand is roughly straight.

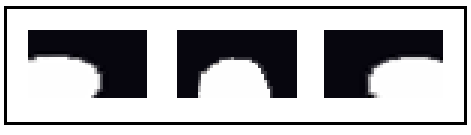

Figure 5: Semicircle fingertip models

After segmenting the hand region, we use the semicircle models (Fig. 5) for a fit to the curved fingertip. We use 3 models for fingertips to cope with different finger orientations. We match semicircle templates against the results of hand segmentation. However, the accuracy of the fingertip detection depends on this matching function. Therefore, we have tried different matching functions, but our experimental results have revealed that the best result is given by using

$$R(x,y) = \frac{\sum_{x',y'} [T(x',y') - I(x+x', y+y')]^2}{\sqrt{[\sum_{x',y'} (T(x',y')^2) \times \sum_{x',y'} I(x+x', y+y')^2]}} \qquad (3)$$

where $T(x, y)$ is a searched template at coordinates *(x,y)* and $I(x, y)$ is a image where the search is running. Following this, we summarize results of the fingertip models using

$$R_{sum}(x,y) = \sum_{i=1}^{N} R_i(x,y) \qquad (4)$$

where *N* is a number of fingertip models.

Then we normalize results of each model. For each pixel, if a result scores more than a given threshold, then the corresponding pixel will be represented as a fingertip. On the other hand, if the result of each pixel is less than the threshold, this pixel will be given as a non-fingertip. As a result, we can successfully locate the positions of fingertips of guitarist.

4. RESULTS

Representative results from our experiments are shown in this section. Fig. 6 provides some results of the experiments. The 'cam' window (top-left window) depicts the input images which are captured from USB camera with resolution 320x240. This USB camera captures the scene that a guitar player uses our system. The 'detect hand output' window (top-right window) represents the hand segmentation results after applying the online adaptation of color probabilities. The 'sumresultbefore' window (bottom-left window) depicts the matching results when using the semicircle fingertip models. Finally, after thresholding, the 'sumresult' window (bottom-right window) shows the output images of fingertip detection.

Colors in the 'sumresultbefore' window and the 'sumresult' window are represented whether each pixel is fingertip or not. The white areas are represented the fingertip regions, while the dark areas are shown the non-fingertip regions.

In our experiments, we used various kinds of background to test our system. We tested our system while the fingers of the hand are not obviously stretched out which is difficult to locate correctly. However, it can be seen that the system can successfully segment the hand region and, at the same time, the fingertip positions can be located. Moreover, in the case that the illumination changes during the online process, the system is also able to locate the fingertip positions correctly. The reader is encouraged to observe the illumination difference between each input image.

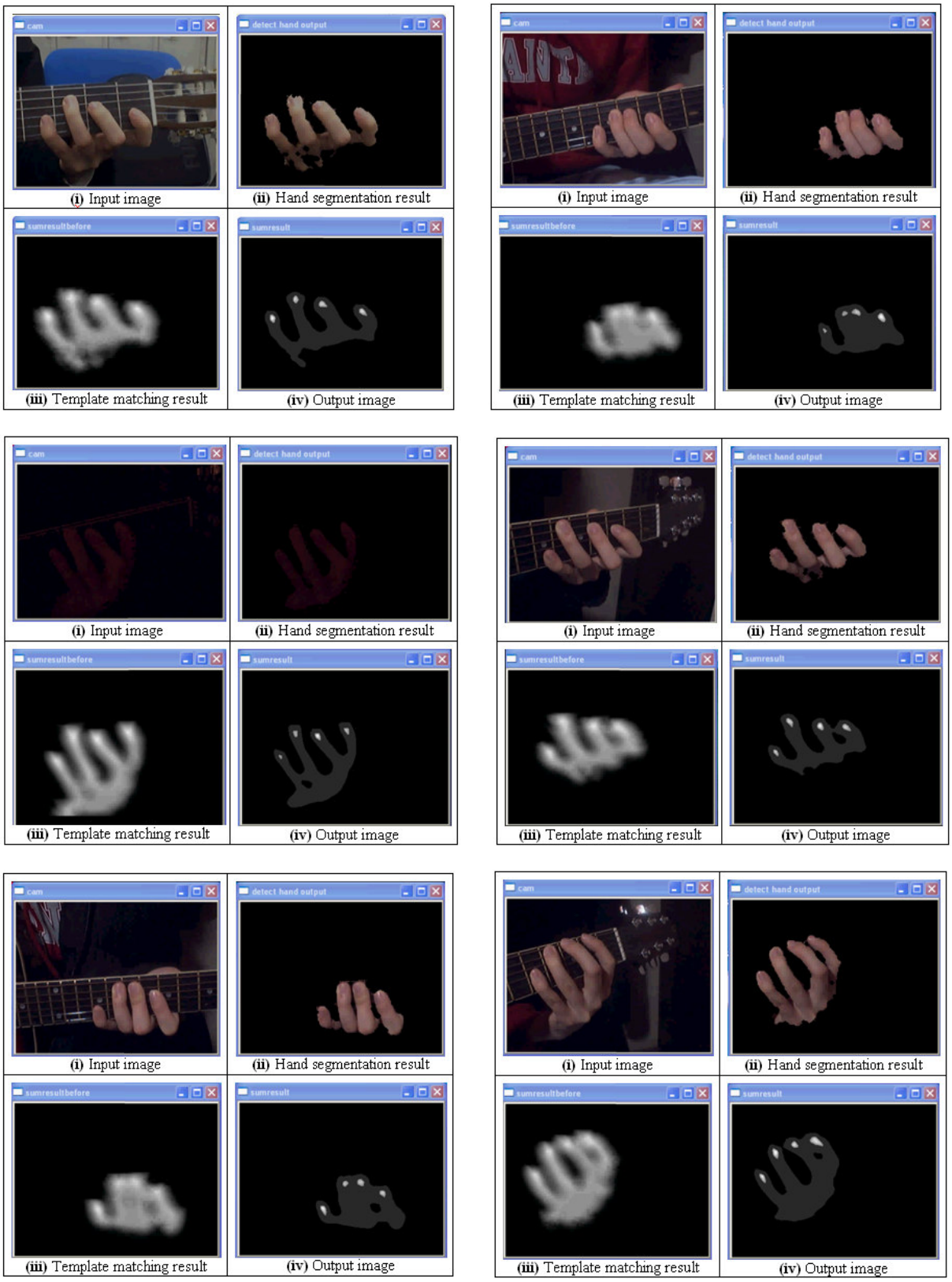

Figure 6: Detections in several poses with various backgrounds and different illumination conditions

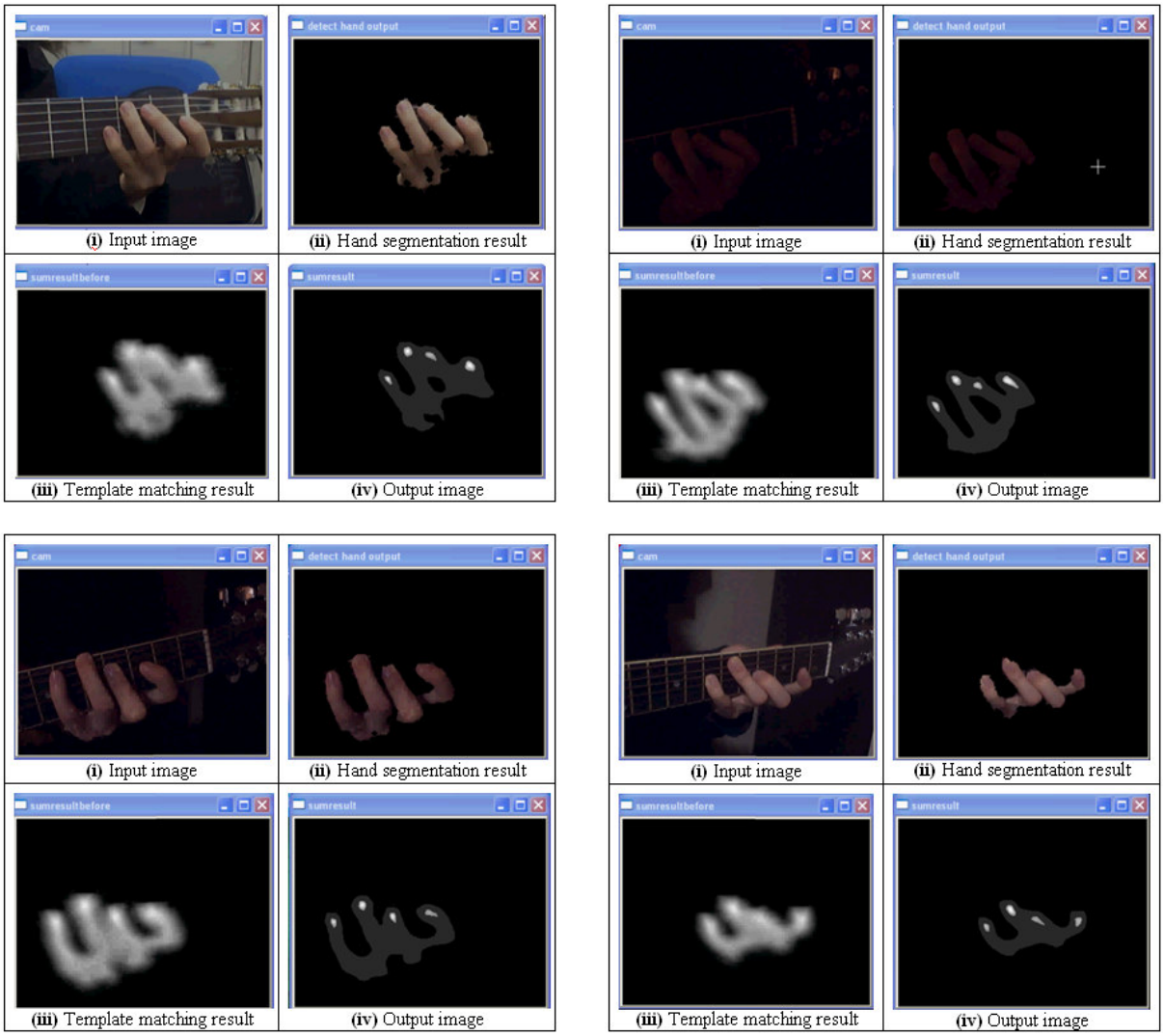

Figure 6 (Cont.): Detections in several poses with various backgrounds and different illumination conditions

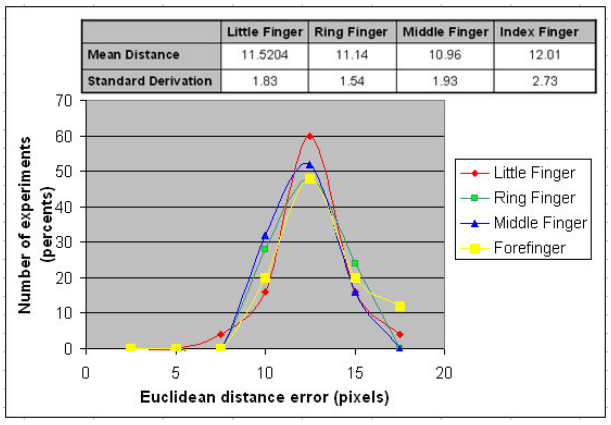

Figure 7: Accuracy of fingertip detection results

We also show quantitative evaluation of our system. We evaluate accuracy by using 25 samples data sets for testing. Fig. 7 shows the accuracy of our experimental results when detecting fingertip positions. All errors are measured in pixels (320x240 image size). With respect to the manually measured ground truth positions, the mean distance error and standard derivation error in each axis are shown in the table in Fig. 7.

5. CONCLUSIONS

In this paper, we have developed a system that locates the positions of the fingertips of a guitar player accurately. We segment the skin colored hand region of guitar player by using on-line adaptation of color probabilities and a Bayesian classifier which can deal with considerable illumination changes and a dynamic background.

Matching algorithm is then used to detect fingertips based on their primitives. This implementation can be used to further develop our guitar application without any fingertip markers (a chord tracker for the guitar learner [4] [5]).

Although we believe that we can successfully produce a system output, the current system has the limitation about the finger self-occlusion. This is because, when finger self-occlusion happens, the fingertip shape does not totally appear as the semicircular shape. For this reason, the assumption that the fingertip shape is a semicircular shape is sometimes difficult to use during occlusion (e.g., in Fig. 8). In the future, we intend to make technical improvements to further refine the system to address this problem. We hope to make the system robust even in an occlusion case.

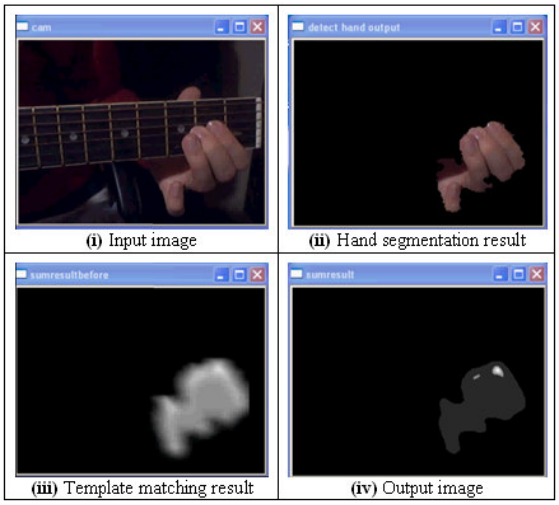

Figure 8: Example of wrong detection (little finger and ring finger) due to finger self-occlusion

Acknowledgments

This work is supported in part by a Grant-in-Aid for the Global Center of Excellence for High-Level Global Cooperation for Leading-Edge Platform on Access Spaces from the Ministry of Education, Culture, Sport, Science, and Technology in Japan. The work presented in this paper is partially supported by CREST, JST (Research Area: Foundation of technology supporting the creation of digital media contents).

References

1. Maki-Patola, T., Laitinen, J., Kanerva A., Takala T.: Experiments with Virtual Reality Instruments. *International Conference on New Interfaces for Musical Expression,* (2005) pages 11-16, Vancouver, Canada
2. Liarokapis, F.: Augmented Reality Scenarios for Guitar Learning. *International Conference on Eurographics UK Theory and Practice of Computer Graphics,* (2005) pages 163-170, Canterbury, UK
3. Motokawa, Y., Saito H.: Support System for Guitar Playing using Augmented Reality Display. *IEEE and ACM International Symposium on Mixed and Augmented Reality,* ISMAR '06, (2006) pages 243-244, Santa Barbara, CA
4. Kerdvibulvech, C., Saito H.: Real-Time Guitar Chord Estimation by Stereo Cameras for Supporting Guitarists. *10th International Workshop on Advanced Image Technology*, IWAIT '07, (2007) pages 256-261, Bangkok, Thailand
5. Kerdvibulvech, C., Saito H.: Vision-Based Guitarist Fingering Tracking Using a Bayesian Classifier and Particle Filters. *IEEE Pacific-Rim Symposium on Image and Video Technology*, PSIVT '07, *Lecture Note in Computer Science of Springer-Verlag (Springer LNCS) 4872*, Advances in Image and Video Technology, (2007) pages 625-638, Santiago, Chile
6. Fiala, M.: Artag, a Fiducial Marker System Using Digital Techniques. *IEEE International Conference on Computer Vision and Pattern Recognition*, CVPR '05, (2005) pages 590-596, San Diego, CA
7. Argyros, A. A., Lourakis, M. I. A..: Tracking Skin-colored Objects in Real-time. Invited Contribution to the "Cutting Edge Robotics Book", ISBN 3-86611-038-3, pages 784, *International Journal of Advanced Robotic Systems* (2005)
8. Argyros, A. A., Lourakis, M. I. A.: Tracking Multiple Colored Blobs with a Moving Camera. *IEEE International Conference on Computer Vision and Pattern Recognition*, CVPR '05, Vol. 2, No. 2, (2005) pages 1178, San Diego, CA
9. Baris Caglar, M., Lobo, N.: Open Hand Detection in a Cluttered Single Image using Finger Primitives. *IEEE International Conference on Computer Vision and Pattern Recognition Workshop*, CVPR '06 Workshop, (2006) pages 148, New York, NY
10. Cakmakci, O., Berard, F.: An Augmented Reality Based Learning Assistant for Electric Bass Guitar. *International Conference on Human-Computer Interaction*, HCI '03, (2003), Crete, Greece
11. Kato, H., Billinghurst M.: Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System. *IEEE and ACM International Workshop on Augmented Reality*, (1999) pages 85-94, San Francisco, CA
12. Burns, A. M., Wanderley M. M.: Visual Methods for the Retrieval of Guitarist Fingering. *International Conference on New Interfaces for Musical Expression*, (2006) pages 196-199, Paris, France
13. Yang M. H., Kriegman D. J., Ahuja, N.: Detecting Faces in Images: A Survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions* on **24** (2002) pages 34-58
14. Jack, K.: Video Demystified. *Elsevier Science 4th Edition* (2004)