Vision-Based Guitarist Fingering Tracking Using a Bayesian Classifier and Particle Filters

Chutisant Kerdvibulvech and Hideo Saito

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

Abstract. This paper presents a vision-based method for tracking guitar fingerings played by guitar players from stereo cameras. We propose a novel framework for colored finger markers tracking by integrating a Bayesian classifier into particle filters, with the advantages of performing automatic track initialization and recovering from tracking failures in a dynamic background. ARTag (Augmented Reality Tag) is utilized to calculate the projection matrix as an online process which allow guitar to be moved while playing. By using online adaptation of color probabilities, it is also able to cope with illumination changes.

Keywords: Guitarist Fingering Tracking, Augmented Reality Tag, Bayesian Classifier, Particle Filters.

1 Introduction

Due to the popularity of acoustic guitars, research about guitars is one of the most popular topics in the field of computer vision for musical applications.

Maki-Patola et al. [1] proposed a system called VAG (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 can produce music as similar as the player is playing the 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 their fingers to play the chord) on an electric guitar to guide 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 an aid to learning how to play the guitar.

In these systems, they do not aim to track the fingering which a player is playing (A pair of gloves are tracked in [1], and graphics information is overlaid on captured video in [2] [3]). We have different goal from most of these researches. In this paper, we propose a new method for tracking the guitar fingerings by using computer vision. Our research goal is to accurately determine and track the fingering positions of a guitarist which is relative to guitar position in 3D space.

A challenge for tracking fingers of a guitar player is naturally that the guitar neck often moves while the guitar is being played. It is then necessary to identify the guitar's position relative to the camera's position. Another important issue is recovery

D. Mery and L. Rueda (Eds.): PSIVT 2007, LNCS 4872, pp. 625–638, 2007.

when finger tracking fails. Our method for tracking fingers of guitar player can handle the mentioned problems. At every frame, we first estimate the projection matrix of each camera by utilizing ARTag (Augmented Reality Tag) [4]. ARTag's marker is placed on the guitar neck. Therefore the world coordinate is defined on the guitar neck as the guitar coordinate system so the system allows the players to move guitar while playing

We utilize a particle filter [5] to track the finger markers in 3D space. We propagate sample particles in 3D space, and project them onto the 2D image planes of both cameras to get the probability of each particle to be on finger markers based on color in both images. To determine the color probabilities being finger markers color, during preprocess we apply a Bayesian classifier that is bootstrapped with a small set of training data and refined through an offline iterative training procedure [6] [7]. Online adaptation of markers-color probabilities is then used to refine the classifier using additional training images. Hence, the classifier is able to deal with illumination changes, even when there is a dynamic background.

In this way, the 3D positions of finger markers can be obtained, so that we can recognize if the fingers of player are pressing the strings or not. As a result, our system can determine the complete positions of all fingers on the guitar fret. It can be used to develop instructive software to aid chord tracking or people learning the guitar. One of the possible applications [8] is to identify whether the finger positions are correct and in accord with the finger positions required for the piece of music that the players are playing. Therefore, guitar players can automatically identify whether their fingers are in the correct position.

2 Related Works

In this section, related approaches of finger detection and tracking of guitarists will be described. Cakmakci and Berard [9] detected the finger position by placing a small ARToolKit (Augmented Reality Toolkit) [10]'s marker on a fingertip of the player for tracking the forefinger position (only one fingertip). However, when we attempted to use the markers to all four fingertips, all markers were not exactly perpendicular when captured by the cameras view direction simultaneously in some angles (especially while the player was pressing their fingers on the strings). Therefore, it is quite difficult to accurately track the positions of four fingers concurrently by using the ARToolKit finger markers.

Burns and Wanderley [11] detected the positions of fingertips for retrieval of guitarist fingering without markers. They assumed that the fingertip shape can be approximated with a semicircular shape while the rest of the hand is roughly straight, and use the circular Hough transform to detect fingertips. However, utilizing Hough transform to detect the fingertips when playing the guitar is not accurate and robust enough. This is because a fingertip shape does not appear as a circular shape in some angles. Also, the lack of contrast between fingertips and background skin adds complication, which often the case in real-life performance.

In addition, these two methods [9] [11] used only one camera on 2D image processing. The constraint of using one camera is that it is very difficult to classify whether fingers are pressing the strings or not. Therefore, stereo cameras are needed

(3D image processing). At the same time, these methods are sometimes difficult to use with stereo cameras because all fingertips may not be perpendicularly captured by two cameras simultaneously.

We propose a method to overcome this problem by utilizing four colored markers placed on the four fingertips to determine the positions of the fingertips. However, a well-known problem of color detection nowadays is the control of lighting. Changing levels of light and limited contrasts prevent correct registration, especially in the case of a cluttered background. A survey [12] provides an interesting overview of color detection. A major decision towards deriving a model of color 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 is to employ bounds on the coordinates of the selected space. However, by using the simple threshold, when changing illumination, it is sometimes difficult to accurately classify the color.

Therefore, we use a Bayesian classifier by learning color probabilities from small training image set and then learn the color probabilities from online input images adaptively (proposed recently in [6] [7]). Applying this method, the first attractive property is that it can avoid the burden involved in the process of manually generating a lot of training data. From small number of training data, it then adapts the probability according to current illumination and converges to a proper value. For this reason, the main attractive property of this method is its ability to cope with changing illumination because it can adaptively describe distribution of markers color.

3 System Configuration

The system configuration is shown in Figure 1. We use two USB cameras and a display connected to the PC for the guitar players. The two cameras capture the position of the left hand (assuming the guitarist is right-handed) and the guitar neck to obtain 3D information. We attach a 4.5cm x 8cm ARTag fiducial marker onto the top right corner of guitar neck to compute the position of the guitar (i.e., the poses of cameras relative to guitar position). The colored markers (with different color) are attached to the fingers of the left hand.

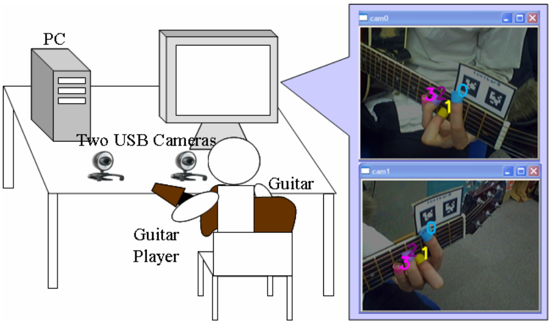


Fig. 1. System configuration

4 Method

Figure 2 shows the schematic of the implementation. After capturing the images, we calculate the projection matrix in each frame by utilizing ARTag. We then utilize a Bayesian classifier to determine the color probabilities of the finger markers. Finally, we apply the particle filters to track the 3D positions of the finger markers.

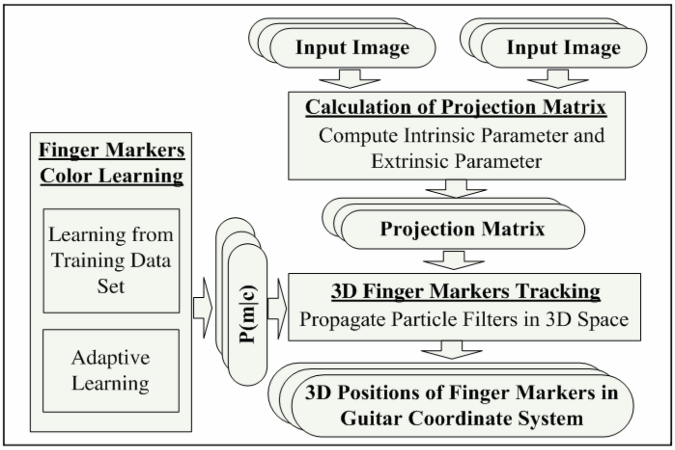

Input Image
Input Image
Calculation of Projection Matrix
Compute Intrinsic Parameter and Extrinsic Parameter
Finger Markers Color Learning
Learning from Training Data Set
Adaptive Learning
Projection Matrix
P(m|c)
3D Finger Markers Tracking
Propagate Particle Filters in 3D Space
3D Positions of Finger Markers in Guitar Coordinate System

Fig. 2. Method overview

4.1 Calculation of Projection Matrix

Detecting positions of fingers in captured images is the main point of our research, and the positions in images can give 3D positions based on stereo configuration of this system. Thus, it is necessary to calculate projection matrix (because it will be then used for projecting 3D particles to the image planes of both cameras in particle filtering step in section 4.3). However, because the guitar neck is not fixed to the ground while the cameras are fixed, the projection matrix changed at every frame. Thus, we have to define the world coordinate on the guitar neck as a guitar coordinate system. In the camera calibration process [13], the relation by projection matrix is generally employed as the method of describing the relation between the 3D space and the images. The important camera properties, namely the intrinsic parameters that must be measured, include the center point of the camera image, the lens distortion and the camera focal length. We first estimate intrinsic parameters during the offline step. During online process, extrinsic parameters are then estimated every frame by utilizing ARTag functions. Therefore we can compute the projection matrix, *P*, by using

$$P = A[R,t] = \begin{bmatrix} \alpha_u & -\alpha_u \cot\theta & u_0 \\ 0 & \alpha_v / \sin\theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{21} & R_{31} & t_x \\ R_{12} & R_{22} & R_{32} & t_y \\ R_{13} & R_{23} & R_{33} & t_z \end{bmatrix} \tag{1}$$

where A is the intrinsic matrix, $[R,t]$ is the extrinsic matrix, u_0 and v_0 are the center point of the camera image, θ is the lens distortion, α_u and α_v represent the focal lengths.

4.2 Finger Markers Color Learning

This section will explain the method we used for calculating the color probabilities being finger markers color which will be then used in the particle filtering step (section 4.3).

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 preprocess. 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 will allow us to get accurate color probabilities being finger markers from only a small set of manually prepared training images because the additional marker regions do not need to be segmented manually. Also, due to adaptive learning, it can be used robustly with changing illumination during the online operation.

4.2.1 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 markers-colored 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. Second, 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(m)* of having marker *m* color in an image. This is the ratio of the marker-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*|*m)* of a marker being color *c*. This is defined as the ratio of the number of occurrences of a color c within the marker-colored areas to the number of marker-colored image points in the training set.

By employing Bayes' rule, the probability *P(m*|*c)* of a color *c* being a marker color can be computed by using

$$P(m \mid c) = \frac{P(c \mid m)P(m)}{P(c)} \tag{2}$$

This equation determines the probability of a certain image pixel being marker-colored using a lookup table indexed with the pixel's color. The resultant probability map thresholds are then set to be $T_{\max}$ and $T_{\min}$, where all pixels with probability $\mathrm{P(m \mid c)} > T_{\max}$ are considered as being marker-colored—these pixels constitute seeds of potential marker-colored blobs—and image pixels with probabilities $\mathrm{P(m \mid c)} > T_{\min}$ where $T_{\min} < T_{\max}$ are the neighbors of marker-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 marker-colored should be considered as a neighbor of an image pixel with high probability of being marker-colored. Indicative values for the thresholds $T_{\max}$ and $T_{\min}$ are 0.5 and 0.15, respectively. A standard connected component labeling 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 marker-colored regions. Each of the remaining connected components corresponds to a marker-colored blob.

4.2.2 Adaptive Learning

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

$$P_A(m \mid c) = \gamma P(m \mid c) + (1-\gamma) P_W(m \mid c) \tag{3}$$

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

Thus, the finger markers-color probabilities can be determined adaptively. By using online adaptation of finger markers-color probabilities, the classifier is able to cope with considerable illumination changes and also a dynamic background (e.g., moving guitar neck).

4.3 3D Finger Markers Tracking

Particle filtering [5] is a useful tool to track objects in clutter, with the advantages of performing automatic track initialization and recovering from tracking failures. In this paper, we apply particle filters to compute and track the 3D position of finger markers in the guitar coordinate system (The 3D information is used to help for determining whether fingers are pressing a guitar string or not). The finger markers can then be automatically tracked initially, and the tracking can be recovered from failures. We use the color probability of each pixel which obtained from the section 4.2 as the observation model

The particle filtering (system) uniformly distributes particles all over the area in 3D space, and then projects the particles from 3D space onto the 2D image planes of the two cameras to obtain the probability of each particle to be finger markers. As new information arrives, these particles are continuously re-allocated to update the position estimate. Furthermore, when the overall probability of particles to be finger markers is lower than the threshold we set, the new sample particles will be uniformly distributed all over the area in 3D space. Then the particles will converge to the areas of finger markers. For this reason, the system is able to recover the tracking. (The calculation is based on the following analysis.)

Given that the process at each time-step is an iteration of factored sampling, the output of an iteration will be a weighted, time-stamped sample-set, denoted by $\{s_t^{(n)}, n = 1,..., N\}$ with weights $\pi_t^{(n)}$, representing approximately the probability-density function $p(X_t)$ at time t: where *N* is the size of sample sets, $s_t^{(n)}$ is defined as the position of the n^{th} particle at time *t*, X_t represents the position in 3D of finger marker at time *t*, $p(X_t)$ is the probability that a finger marker is at 3D position *X* = (x,y,z) ^{T} at time *t*. The number of particles used is 900 particles. The iterative process can be divided into three main stages: (i) Selection stage; (ii) Predictive state; (iii) Measurement stage.

In the first stage (the selection stage), a sample $s'^{(n)}_t$ is chosen from the sample-set $\{s_{t-1}^{(n)}, \pi_{t-1}^{(n)}, c_{t-1}^{(n)}\}$ with probabilities $\pi_{t-1}^{(j)}$, where $c_{t-1}^{(n)}$ is the cumulative weight. This is done by generating a uniformly distributed random number $r \in$ *[0, 1]*. We find the smallest *j* for which $c_{t-1}^{(j)} \geq r$ using binary search, and then $s'^{(n)}_t$ can be set as follows: $s'^{(n)}_t = s_{t-1}^{(j)}$.

Each element chosen from the new set is now subjected to the second stage (the predictive step). We propagate each sample from the set s'_{t-1} by a propagation function, $g(s'^{(n)}_t)$, using

$$s_t^{(n)} = g(s'^{(n)}_t) + noise \tag{4}$$

where noise is given as a Gaussian distribution with its mean = $(0,0,0)^T$. The accuracy of the particle filter depends on this propagation function. We have tried different propagation functions (e.g., constant velocity motion model and acceleration motion model), but our experimental results have revealed that using only noise information gives the best result. A possible reason is that the motions of finger markers are usually quite fast and constantly changing directions while playing the guitar. Therefore the calculated velocities or accelerations in previous frame do not give accurate prediction of the next frame. In this way, we use only the noise information by defining $g(x) = x$ in Equation (4).

In the last stage (the measurement stage), we project these sample particles from 3D space to two 2D image planes of cameras using the projection matrix results from Equation (1). We then determine the probability whether the particle is on finger marker. In this way, we generate weights from the probability-density function $p(X_t)$ to obtain the sample-set representation $\{(s_t^{(n)}, \pi_t^{(n)})\}$ of the state-density for time t using

$$\pi_t^{(n)} = p(X_t = s_t^{(n)}) = P_A(m \mid c)_{Camera0} P_A(m \mid c)_{Camera1} \tag{5}$$

where $p(X_t = s_t^{(n)})$ is the probability that a finger marker is at position $s_t^{(n)}$.

We assign the weights to be the product of $P_A(m \mid c)$ of two cameras which can be obtained by Equation (3) from the finger markers color learning step (the adapted probability $P_A(m \mid c)_{Camera0}$ and $P_A(m \mid c)_{Camera1}$ represent a color c being a marker color in camera 0 and camera 1, respectively). Following this, we normalize the total weights using the condition

$$\Sigma_n \pi_t^{(n)} = 1 \tag{6}$$

Next, we update the cumulative probability, which can be calculated from normalized weights using

$$c_t^{(0)} = 0,\ c_t^{(n)} = c_t^{(n-1)} + \pi_t^{(n)}{}_{Total} \qquad (n = 1,...,N) \tag{7}$$

where $\pi_t^{(n)}{}_{Total}$ is the total weight.

Once the N samples have been constructed, we estimate moments of the tracked position at time-step t as using

$$\varepsilon[f(X_t)] = \Sigma_{n=1}^{N} \pi_t^{(n)} s_t^{(n)} \tag{8}$$

where $\varepsilon[f(X_t)]$ represents the centroid of each finger marker. The four finger markers can then be tracked in 3D space, enabling us to perform automatic track initialization and track recovering even in dynamic background. The positions of four finger markers in the guitar coordinate system can be obtained.

5 Results

In this section, representative results from our experiment are shown. Figure 3 provides a few representative snapshots of the experiment. The reported experiment is based on a sequence that has been acquired. Two USB cameras with resolution 320x240 have been used.

The camera 0 and camera 1 windows depict the input images which are captured from two cameras. These cameras capture the player's fingers in the left hand positioning and the guitar neck from two different views. For visualization purposes, the 2D tracked result of each finger marker is also shown in camera 0 and camera 1 windows. The four colored numbers depict four 2D tracking results from the finger markers (forefinger [number0 - light blue], middle finger [number1 - yellow], ring finger [number2 - violet] and little finger [number3 - pink]).

The 3D reconstruction window, which is drawn using OpenGL, represents both the tracked 3D positions of the four finger markers in guitar coordinate system. In this 3D space, we show the virtual guitar board to make it clearly understand that this is the guitar coordinate system. The four-color 3D small cubes show each 3D tracked result of the finger markers (these four 3D cubes correspond to the 2D four colored numbers in the camera 0 and the camera 1 windows).

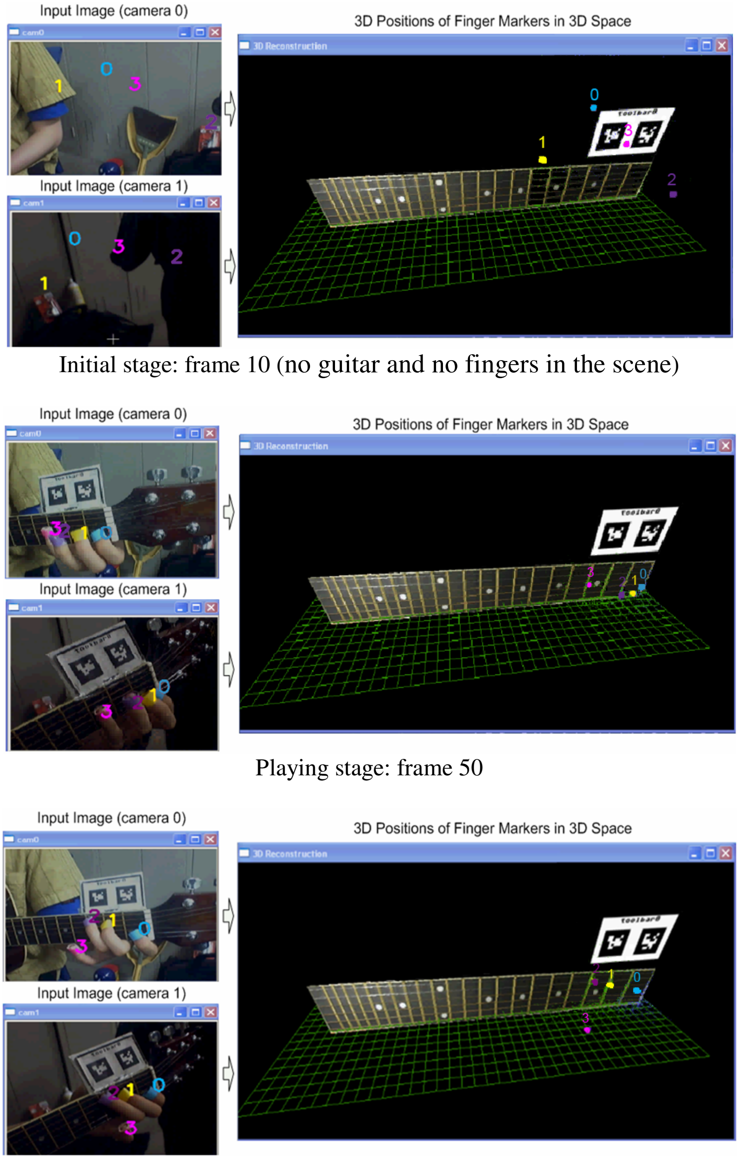
In the initial stage (frame 10), when the experiment starts, there are no guitar and no fingers in the scene. The tracker attempts to find the color which is similar to the markers-colored region. For example, because the color of player's shirt (light yellow) is similar to a middle finger marker's color (yellow), the 2D tracking result of middle finger marker (number1) in the camera 0 window detects wrongly as if the player's shirt is the middle finger marker.

However, later during the playing stage (frame 50), the left hand of a player and the guitar enter the fields of cameras' views. The player is playing the guitar, and then the system can closely determine the accurate 3D fingering positions which correspond to the 2D colored numbers in the camera 0 and the camera 1 windows. In this way, this implies that the system can perform automatic track initialization because of using particle filtering.

Next, the player changes to hold to the next fingering positions in frame 80. The system can continue to correctly track and recognize the 3D fingering positions which correspond nearly to the positions of 2D colored numbers in the camera 0 and the camera 1 windows. Following this, the player moves the guitar position (from the old position in frame 80) to the new position in frame 110, but still holding the same fingering positions on the guitar fret. It can be observed that the detected 3D positions of the four finger markers from different guitar positions (i.e., but the same input fingering on the guitar fret) are almost the same positions. This is because ARTag marker is used to track the guitar position.

Later on, in the occlusion stage (frame 150), the finger markers are totally occluded by the white paper. Therefore, the system is again back to find the similar colors of each marker (backing to the initial stage again).

However, following this in the recovering stage (frame 180), the occlusion of white paper is moved out, and then the cameras are capturing the fingers and guitar neck again. It can be seen that the tracker can return to track the correct fingerings (backing to the playing stage again). In other words, the system is able to recover from tracking failure due to using particle filtering.

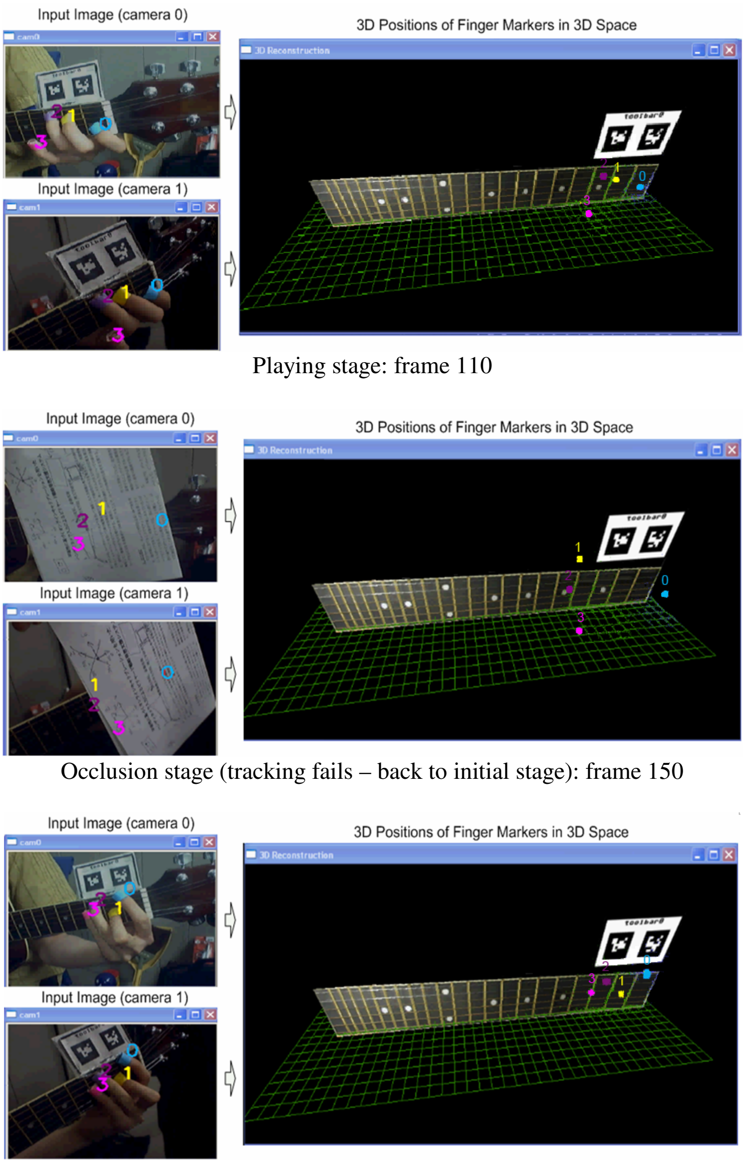


Initial stage: frame 10 (no guitar and no fingers in the scene)

Playing stage: frame 50

Playing stage: frame 80

Fig. 3. Representative snapshots from the online tracking experiment



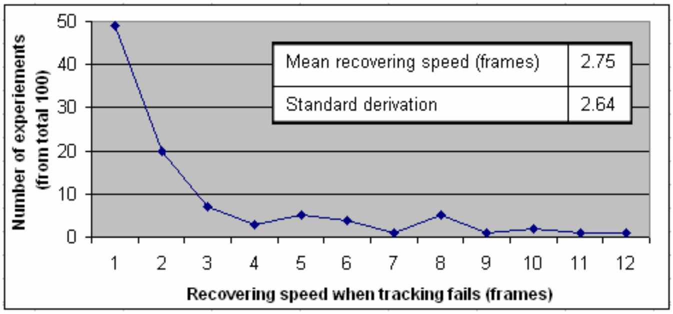
Playing stage: frame 110

Occlusion stage (tracking fails – back to initial stage): frame 150

Recovering stage (playing stage again): frame 180

Fig. 3. (*continued*)

The reader is also encouraged to observe illumination difference between camera 0 and camera 1 windows. Our experimental room composes of two main light sources which are located oppositely. We turned on the first light source of the room which is located near to use for capturing images in camera 0, while we turned off the second light source (opposite to the first source) of the room which is located near for capturing images in camera 1. Hence, the lighting used to test in each camera is

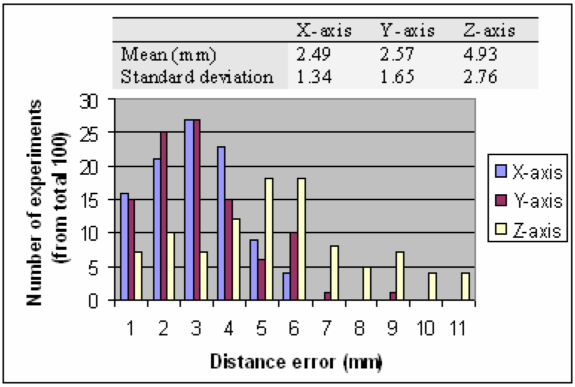


Mean recovering speed (frames)	2.75
Standard derivation	2.64

Fig. 4. Speed used for recovering from tracking failures

different. However, it can be observed that the 2D tracked result of finger markers can be still determined without effects of different light sources in both camera 0 and camera 1 windows in each representative frame. This is because a Bayesian classifier and online adaptation of color probabilities are utilized to deal with this.

We also evaluate the recovering speed whenever tracking the finger markers fails. Figure 4 shows the speeds used for recovering from lost tracks. In this graph, the recovering speeds are counted from initial frame where certainty of tracking is lower than threshold. At the initial frame, the particles will be uniformly distributed all over the 3D space as described in the section 4.3. Before normalized weights in particle filtering step, we determine the certainty of tracking from the sum of the weight probability of each distributed particle to be marker. Therefore, if the sum of weight probability is lower than the threshold, we assume that tracker is failing. On the other hand, if the sum of weight probability is higher than threshold, we imply that tracking has been recovered. Thus, the last counted frame will be decided at this frame (the particles have been already converged to the areas of finger markers). The mean recovering speed and the standard derivation are also shown in the table in Figure 4, in frames (the speed of fingering tracking is approximately 6 fps). We believe this recovering speed is fast enough for recovering of tracking in real-life guitar performance.



	X-axis	Y-axis	Z-axis
Mean (mm)	2.49	2.57	4.93
Standard deviation	1.34	1.65	2.76

Fig. 5. Accuracy of 3D finger detection results

Then, we evaluate accuracy of our system by using 100 samples data sets for testing. Figure 5 shows the accuracy of our experimental results when detecting fingering positions. All errors are measured in millimetre. 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 Figure 5.

Finally, we will note about a limitation of the proposed system. The constraint of our system is that, although a background we used can be cluttered, the background should not be composed of large objects which are the same color as the colors of finger markers. For instance, if the players wear their clothes which are very similar color to the markers' colors, the system cannot sometimes determine the output correctly.

6 Conclusions

In this paper, we have developed a system that measures and tracks the positions of the fingertips of a guitar player accurately in the guitar's coordinate system. A framework for colored finger markers tracking has been proposed based on a Bayesian classifier and particle filters in 3D space. ARTag has also been utilized to calculate the projection matrix.

Although we believe that we can successfully produce a system output, the current system has the limitation about the background color and the markers' colors. Because four finger markers composed of four different colors, it is sometimes not convenient for users to select their background. As future work, we intend to make technical improvements to further refine the problem of the finger markers by removing these markers which may result in even greater user friendliness.

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.

References

1. Maki-Patola, T., Laitinen, J., Kanerva, A., Takala, T.: Experiments with Virtual Reality Instruments. In: Fifth International Conference on New Interfaces for Musical Expression, Vancouver, Canada, pp. 11–16 (2005)
2. Liarokapis, F.: Augmented Reality Scenarios for Guitar Learning. In: Third International Conference on Eurographics UK Theory and Practice of Computer Graphics, Canterbury, UK, pp. 163–170 (2005)
3. Motokawa, Y., Saito, H.: Support System for Guitar Playing using Augmented Reality Display. In: Fifth IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR 2006, pp. 243–244. IEEE Computer Society Press, Los Alamitos (2006)
4. Fiala, M.: Artag, a Fiducial Marker System Using Digital Techniques. In: IEEE International Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 1, pp. 590–596. IEEE Computer Society Press, Los Alamitos (2005)

5. Isard, M., Blake, A.: Condensation - conditional density propagation for visual tracking. International Journal on Computer Vision, IJCV 1998 29(1), 5–28 (1998)
6. 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, Advanced Robotic Systems International (2005)
7. Argyros, A.A., Lourakis, M.I.A.: Tracking Multiple Colored Blobs with a Moving Camera. In: IEEE International Conference on Computer Vision and Pattern Recognition, CVPR 2005, San Diego, CA, vol. 2(2), p. 1178 (2005)
8. Kerdvibulvech, C., Saito, H.: Real-Time Guitar Chord Estimation by Stereo Cameras for Supporting Guitarists. In: Tenth International Workshop on Advanced Image Technology, IWAIT 2007, Bangkok, Thailand, pp. 256–261 (2007)
9. Cakmakci, O., Berard, F.: An Augmented Reality Based Learning Assistant for Electric Bass Guitar. In: Tenth International Conference on Human-Computer Interaction, HCI 2003, Rome, Italy (2003)
10. Kato, H., Billinghurst, M.: Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System. In: Second IEEE and ACM International Workshop on Augmented Reality, pp. 85–94. IEEE Computer Society Press, Los Alamitos (1999)
11. Burns, A.M., Wanderley, M.M.: Visual Methods for the Retrieval of Guitarist Fingering. In: Sixth International Conference on New Interfaces for Musical Expression, Paris, France, pp. 196–199 (2006)
12. Yang, M.H., Kriegman, D.J., Ahuja, N.: Detecting Faces in Images: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 34–58 (2002)
13. Forsyth, D.A., Ponce, J.: Computer Vision: A Modern Approach. Prentice Hall, Upper Saddle River, NJ (2003)
14. Jack, K.: Video Demystified. Elsevier Science, UK (2004)