

Research Article

Guitarist Fingertip Tracking by Integrating a Bayesian Classifier into Particle Filters

Chutisant Kerdvibulvech and Hideo Saito

Department of Information and Computer Science, Keio University, 3-14-1 Hiyoshi, Kohoku-ku 223-8522, Japan

Correspondence should be addressed to Chutisant Kerdvibulvech, chutisant@ozawa.ics.keio.ac.jp

Received 3 October 2007; Revised 19 July 2008; Accepted 30 August 2008

Recommended by Adrian Cheok

We propose a vision-based method for tracking guitar fingerings made by guitar players. We present it as a new framework for tracking colored finger markers by integrating a Bayesian classifier into particle filters. This adds the useful abilities of automatic track initialization and recovery from tracking failures in a dynamic background. Furthermore, by using the online adaptation of color probabilities, this method is able to cope with illumination changes. Augmented Reality Tag (ARTag) is then utilized to calculate the projection matrix as an online process which allows the guitar to be moved while being played. Representative experimental results are also included. The method presented can be used to develop the application of human-computer interaction (HCI) to guitar playing by recognizing the chord being played by a guitarist in virtual spaces. The aforementioned application would assist guitar learners by allowing them to automatically identify if they are using the correct chords required by the musical piece.

Copyright © 2008 C. Kerdvibulvech and H. Saito. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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 “Virtual Air Guitar” (VAG) 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 track 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, 3]). We have different goal from most of these researches. In this paper, we propose a method for tracking the guitar fingering by using computer vision. Our research goal is to accurately determine and track the fingertip positions of a guitarist which are relative to guitar position in 3D space.

A challenge for tracking the fingers of a guitar player is that the guitar neck usually moves while the guitar is being played. It is therefore necessary to identify the guitar's position relative to the camera's position. Another important issue is recovery when the finger tracking fails. Our method for tracking the fingers of guitar player solves these problems.

At every frame, we first estimate the projection matrix of each camera by utilizing Augmented Reality Tag (ARTag) [4]. ARTag's marker is placed on the guitar neck. Therefore, the world coordinate system 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 the preprocessing 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 such as a chord tracker for the guitar learner. One of the example 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 is being played. Because guitar players can automatically identify whether their fingers are in the correct position, it would be an invaluable teaching aid for people learning to play the guitar.

2. Related Works

Related approaches for finger detection and tracking of guitarists will be described in this section. Cakmakci and Berard [9] detected the finger position by placing a small Augmented Reality Toolkit's (ARToolKit) [10] 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(s) in some angles, especially while the player was pressing their fingers on the strings. Therefore, it was 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 the retrieval of the 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 uses the circular Hough transform to detect fingertips. However, utilizing Hough transform to detect the fingertips when playing the guitar is neither accurate nor robust enough. This is because a fingertip shape does not appear as a circular shape in some angles. Also, in real-life performance, the lack of contrast between fingertips and background skin adds a complication.

In addition, these two methods [9, 11] use only one camera on the 2D image processing. The problem with using one camera is that it is very difficult to classify whether the fingers are pressing the strings or not. Therefore, stereo cameras are needed (3D image processing). However, when using these methods it is sometimes difficult to employ stereo cameras because all fingertips may not be perpendicularly captured by the two cameras simultaneously.

We therefore 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 current problem of color detection

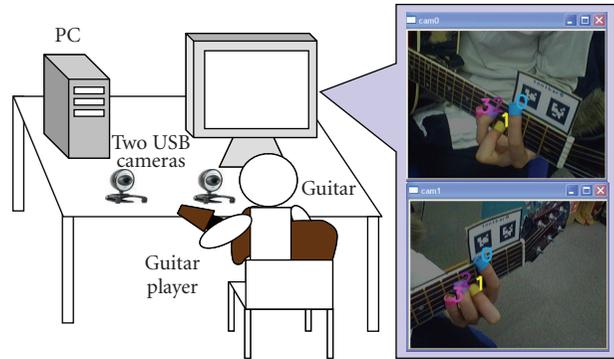


FIGURE 1: System configuration.

is the control of the lighting. Changing the levels of light and limited contrasts prevents correct registration, especially when there is a cluttered background. The survey of detecting faces in images [12] provides an interesting overview of color detection. A major decision toward 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, it is sometimes 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 [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 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 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 $4.5\text{ cm} \times 8\text{ cm}$ 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 fingertips of the left hand.

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

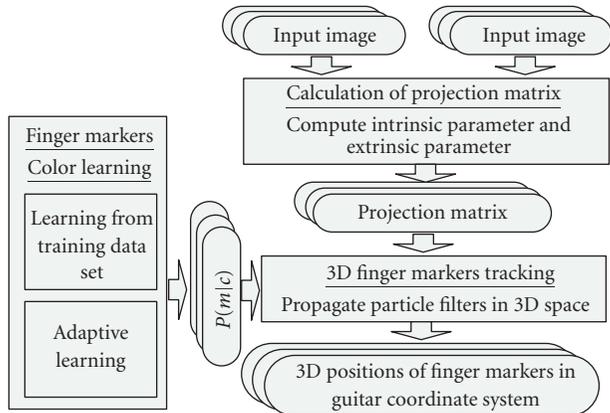


FIGURE 2: Method overview.

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.

4.1. Calculation of Projection Matrix

Detecting position of the 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, while the cameras are fixed, the guitar neck is not fixed to the ground, and therefore the projection matrix changes at every frame. Thus, we have to define the world coordinate system on the guitar neck as a guitar coordinate system (Figure 3). 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. As shown in (1), the matrices R and t in the camera calibration matrix describe the position and orientation of the camera with respect to world coordinate system. During online process, extrinsic parameters are then estimated in every frame by utilizing ARTag functions. Therefore, we can compute the projection matrix, P , by using

$$P = A[R, t]$$

$$= \begin{bmatrix} \alpha_u & -\alpha_u \cot \theta & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & \sin \theta & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{21} & R_{31} & t_x \\ R_{12} & R_{22} & R_{32} & t_y \\ R_{13} & R_{23} & R_{33} & t_z \end{bmatrix}, \quad (1)$$

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, and α_u and α_v represent the focal lengths.

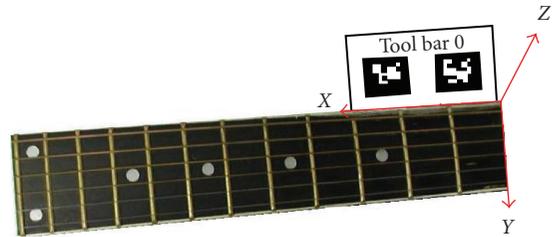


FIGURE 3: The world coordinate system on the top right corner of guitar neck.

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.

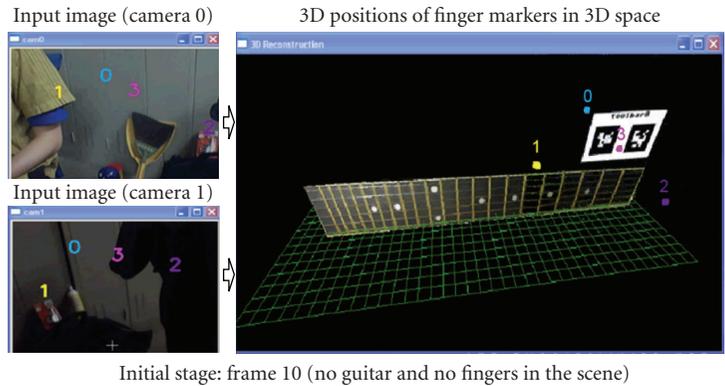
Therefore, this method will allow us to get accurate color probabilities of the finger markers from only a small set of manually prepared training images. This is because the additional marker regions do not need to be segmented manually. Also, because of the adaptive learning, it can be used robustly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

4.2.1. Learning from Training Data Set

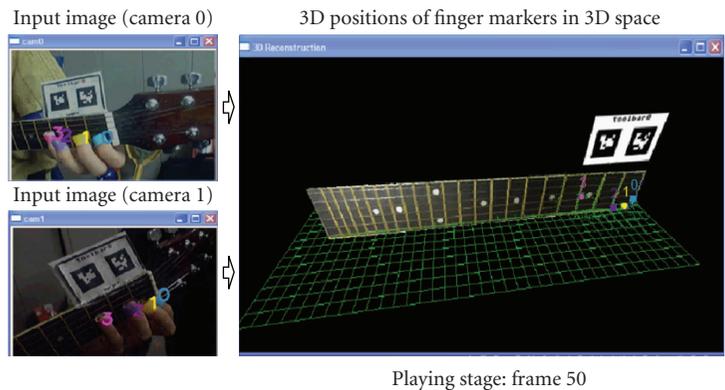
During an offline phase, a small set of training input images (20 images) is selected on which a human operator manually segments markers-colored regions. The color representation used in this process is YUV 4:2:2 [14]. However, the Y-component of this representation is not employed for two reasons. 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. 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 the following.

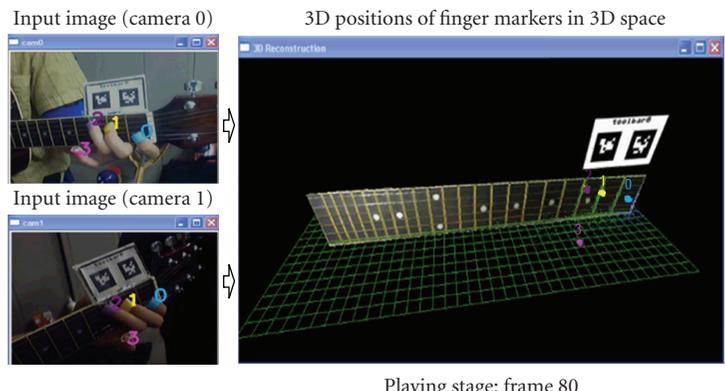
- (i) The prior probability $P(m)$ of having marker m color in an image: this is the ratio of the marker-colored pixels in the training set to the total number of pixels of whole training images.
- (ii) The prior probability $P(c)$ of the occurrence of each color in an image: this is computed as the ratio of the number of occurrences of each color c to the total number of image points in the training set.



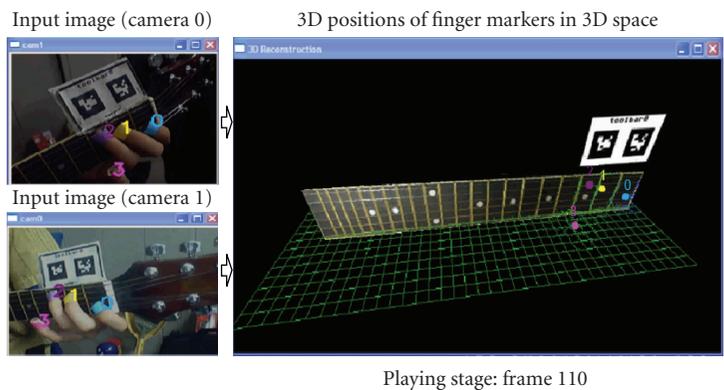
(a)



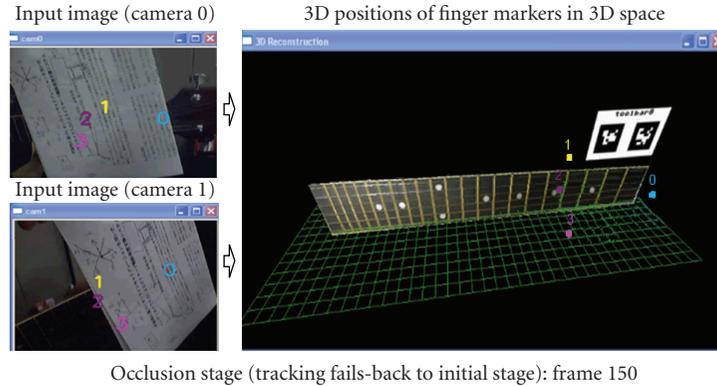
(b)



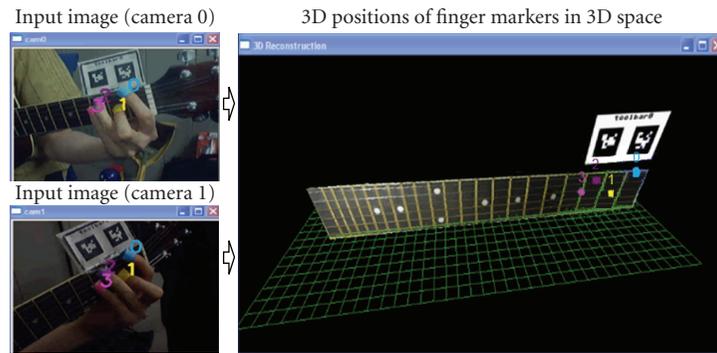
(c)



(d)



(e)



(f)

FIGURE 4: Representative snapshots from the online tracking experiment.

- (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

$$P(m | c) = \frac{P(c | m)P(m)}{P(c)}. \quad (2)$$

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 threshold T_{\max} and threshold T_{\min} , where all pixels with probability $P(m | c) > T_{\max}$ are considered as being marker colored—these pixels constitute seeds of potential marker-colored blobs—and image pixels with probabilities $P(m | c) > T_{\min}$ 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. The values for T_{\max} and T_{\min} should be determined by test experiments (we use 0.5

and 0.15, resp., in the experiment in this paper). A standard connected component labelling algorithm 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 using the UV color representation model has certain illumination independent characteristics, the marker-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 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. The second

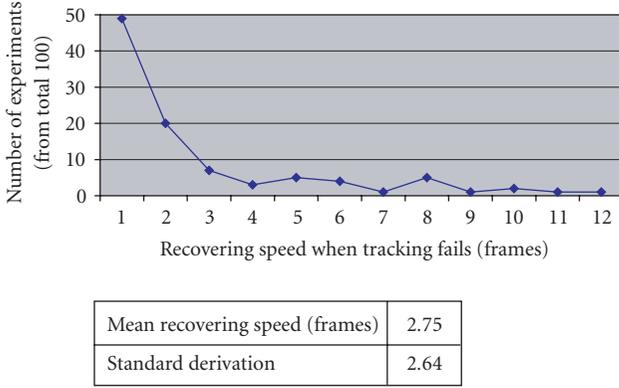


FIGURE 5: Speed used for recovering from tracking failures.

is made up of $P_W(m)$, $P_W(c)$, $P_W(c | m)$ corresponding to $P(m)$, $P(c)$, $P(c | m)$ that the system gathers 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:

$$P_A(m | c) = \gamma P(m | c) + (1 - \gamma) P_W(m | c), \quad (3)$$

where γ is a sensitivity parameter that controls the influence of the training set in the detection process, $P_A(m | c)$ represents the adapted probability of a color c being a marker color, and $P(m | c)$ and $P_W(m | c)$ are both given by (2) but involve prior probabilities that have been computed from the whole training set [for $P(m | c)$] and from the detection results in the last W frames [for $P_W(m | c)$]. 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 easily 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 a 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 determine whether fingers are pressing a guitar string or not). The finger markers can then be automatically tracked, and the tracking can be recovered from the failures.

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 reallocated 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 $\{s_t^{(n)}, n = 1, \dots, N\}$ 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 , and $p(X_t)$ is the probability that a finger marker is at 3D position $X = (x, y, z)^T$ at time t .

The number of particles we used, N , is 900 particles in each marker. The dimensions of the 3D space we used in the particle filter are 20 cm, 15 cm, and 7 cm for x -axis, y -axis, and z -axis respectively, that is, $x \in [-3, 17]$, $y \in [-7, 8]$, and $z \in [-2, 5]$. Each axis and the origin of the world coordinate space are depicted in Figure 3. This size of 3D space is determined by measuring from the size of actual guitar we used. The size should cover the area of guitar neck in which the fingers of a player will hold a chord. If the size of 3D space is too small, the finger markers cannot be successfully tracked whenever the markers leave the scope of the defined 3D space. However, if the size is too large, the accuracy of tracking will decline. For this reason, the suitable size of 3D space should be determined.

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 $s_t^{(n)}$ 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 $s_t^{(n)}$ can be set as follows: $s_t^{(n)} = s_{t-1}^{(j)}$.

Each element chosen from the new set is now subjected to the second stage (the predictive step). We propagate each sample from the set $s_{t-1}^{(n)}$ by a propagation function, $g(s_t^{(n)})$, using

$$s_t^{(n)} = g(s_{t-1}^{(n)}) + \text{noise}, \quad (4)$$

where noise is given as a Gaussian distribution with its mean = $(0, 0, 0)^T$. The value for the variance of the Gaussian distribution of noise should be determined by test experiments (we use the variance of Gaussian distribution = 12 cm^2 in the experiment in this paper). Specifically, the particles should be distributed covering the area that the finger markers will move in the consecutive frame. Therefore, if the variance value is too low, the tracker will easily fail

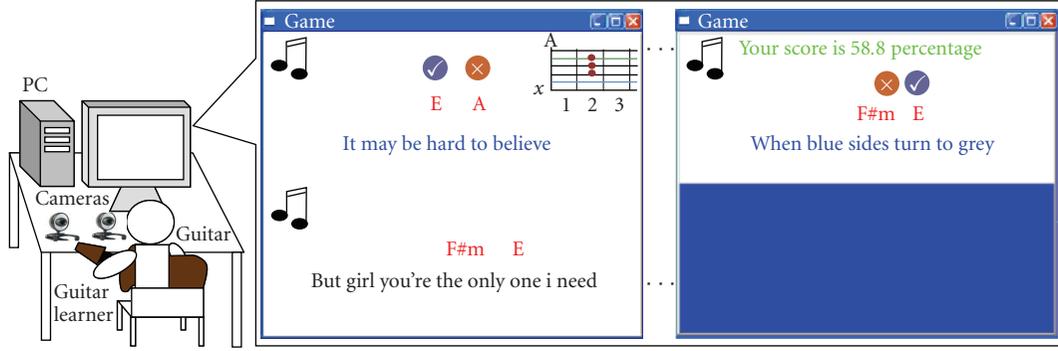


FIGURE 6: User interface of our guitar application.

to track and need to recover too frequently. On the other hand, if the variance is too high, the accuracy of tracking will decrease. Therefore, a suitable value of variance should be determined by test experiments.

Also, 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 (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 (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

$$\begin{aligned} \pi_t^{(n)} &= p(X_t = s_t^{(n)}) \\ &= P_A(m | c)_{\text{Camera}0} \times P_A(m | c)_{\text{Camera}1}, \end{aligned} \quad (5)$$

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 $P_A(m | c)$ of two cameras which can be obtained by (3) from the finger markers color learning step (the adapted probabilities $P_A(m | c)_{\text{Camera}0}$ and $P_A(m | c)_{\text{Camera}1}$ represent a color c being a marker color in camera 0 and camera 1, resp.). Following this, we normalize the total weights using the condition

$$\sum_n \pi_t^{(n)} = 1. \quad (6)$$

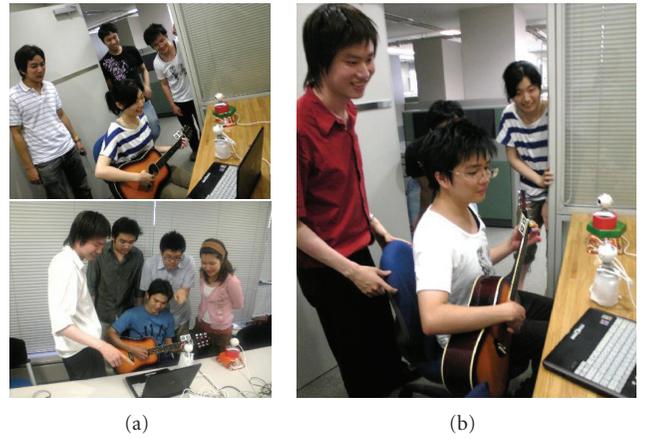


FIGURE 7: The system environment while users were testing our system.

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

$$c_t^{(0)} = 0, \quad c_t^{(n)} = c_t^{(n-1)} + \pi_t^{(n)} \pi_{\text{Total}}^{(n)} \quad (n = 1, \dots, N), \quad (7)$$

where $\pi_{\text{Total}}^{(n)}$ is the total weight.

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

$$\varepsilon[f(X_t)] = \sum_{n=1}^N \pi_t^{(n)} s_t^{(n)}, \quad (8)$$

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 a dynamic background. The positions of four finger markers in the guitar coordinate system can be obtained.

5. Tracking Results

Representative results from our experiment are shown in this section. Figure 4 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 320×240 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 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 understood 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 (i.e., the particle filter tries to find objects whose color is the most similar to the color of our markers in the background). 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 fingertip positions which correspond to the 2D colored numbers in the camera 0 and the camera 1 windows. In this way, the system can perform automatic track initialization because it is using particle filtering.

Next, the player changes to 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 holds 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 (returning to the initial stage again). Also, when the ARTag marker cannot be found in the scene, the system cannot determine the accurate 3D tracking results in the

guitar coordinate space. This is because the projection matrix cannot be obtained correctly.

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 (returning to the playing stage again). In other words, the system is able to recover from tracking failure because it is using particle filtering.

The reader is also encouraged to observe the illumination difference between camera 0 and camera 1 windows. Our experimental room has two main light sources which are located opposite to each other. We turned on the first light source of the room which is quite near for capturing images on camera 0, while we turned off the second light source (which is opposite to the first source) and is located near camera 1. Hence, the lighting used to test in each camera is 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 an online adaptation of color probabilities are utilized to deal with this.

We also evaluate the recovering speed whenever tracking the finger markers fails. Figure 5 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 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 the tracker is failing. On the other hand, if the sum of weight probability is higher than threshold, we imply that the 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 5, in frames. We believe this recovering speed is fast enough for recovering of tracking in real-life guitar performance.

Finally, we will refer to a limitation of the proposed system. The constraint of our system is that, although the 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 of a very similar color to the markers' colors, the system cannot sometimes determine the output correctly.

6. Sample Application

Learning to play the guitar usually involves tedious lessons in fingering positions for the left hand. It is difficult for beginners to recognize by themselves whether they are accurately positioning their fingers on the string. For this reason, we developed the application of human-computer interaction

TABLE 1: Qualitative feedback from participants.

Criteria	Subjective feedback (number of respondents)				Total
	Excellent	Good	Fair	Poor	
Interest in application	14	1	—	—	15
System smoothness	8	5	2	—	15
User satisfaction	12	3	—	—	15
Ease of use	7	7	1	—	15
System naturalness (comfortable feeling)	3	5	6	1	15
Overall	10	4	1	—	15

(HCI) to guitar playing, named *InteractiveGuitarGame*, by applying the guitarist fingertip tracking method presented in this paper. This application aims to assist guitar learners by automatically identifying whether the fingertip positions are correct and in accord with the fingertip positions required for the piece of music that is being played.

6.1. Description of the Application

The *InteractiveGuitarGame* application recognizes the guitar chord being played by a guitar learner based on the 3D position of finger markers in the virtual guitar coordinate spaces. Then, the system gives real-time feedback to guitar learners telling them if they are using the correct chords required by the musical piece.

An example user interface of the application is shown in Figure 6. This application contains the lyrics, guitar chord charts, and vocal information. The lyrics are displayed on the screen in color which changes and is synchronized with the music. The guitar chord charts are shown in the top right of the user interface that can be used to guide student guitarists to play a song. Most importantly, this application recognizes if each successive chord contained in the music is being played correctly. It is considered that this would be of a great assistance to guitar learners because they are able to automatically identify if their own finger positions are correct and whether they are matching the correct chords required by the musical piece. This feedback is shown by small right/wrong symbols above the corresponding chords in real time. Finally, this application will show an overall evaluation score, as a percentage, indicating the user’s accuracy when the performance has been completed which provides greater user friendliness. This application would be invaluable as a teaching aid for guitar learners.

6.2. Evaluation of the Application

We conducted a user study to evaluate the effectiveness of the aforementioned application. Fifteen users (nine inexperienced and six experienced guitar players) were asked to test our system. Each user took approximately 5–10 minutes to run the study. All users were able to use the system after a short explanation. After the individual tests, the users were asked to give qualitative feedback (Table 1). This included interest in the application, smoothness of the system, user satisfaction, ease of use of the interface, naturalness of

the system, and overall impression. General comments on the test were also collected from the users. The system environment while users were testing our system is shown in Figure 7.

From this user study, we received mostly positive comments from users. Many users agreed that it is useful for learning to play the guitar. Also, they were satisfied with the smoothness and the speed of our system. They reasoned that this was because the system can run in real time and was synchronized with the real music. Moreover, many users indicated that they were impressed by the system, especially by the idea of developing this application. They gave the reason that this was their first time to see how a computer using cameras can give a feedback to players interactively for an actual stringed guitar. Interestingly, several users seemed to be more enthusiastic with the overall evaluation score when the song finished playing. They gave their reason since this made it more interesting as a real guitar game. Hence this evaluation score could attract learners to learn and enjoy the guitar lessons in music school.

However, the most common complaint was about finger markers. Some users indicated that, although they preferred the idea of this application, it was slightly difficult to use because colored markers were placed on the fingertips. In other words, since fingertip markers were required, some users felt less comfortable using our system. Thus the system seemed to be slightly unnatural. This was the reason why they gave lower scores to system naturalness (comfortable feeling), as presented in Table 1.

7. 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. This implementation can be used to develop instructive software such as a chord tracker for a guitar learner.

Although we believe that we can successfully produce an accurate system output, the current system has the limitation about the colored finger markers because finger markers are required in our current system. This sometimes makes it unnatural for playing guitar in real life. 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] T. Maki-Patola, J. Laitinen, A. Kanerva, and T. Takala, "Experiments with virtual reality instruments," in *Proceedings of the 5th International Conference on New Interfaces for Musical Expression (NIME '05)*, pp. 11–16, Vancouver, Canada, May 2005.
- [2] F. Liarokapis, "Augmented reality scenarios for guitar learning," in *Proceedings of the 3rd International Conference on Eurographics UK Theory and Practice of Computer Graphics*, pp. 163–170, Canterbury, UK, May 2005.
- [3] Y. Motokawa and H. Saito, "Support system for guitar playing using augmented reality display," in *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '06)*, pp. 243–244, Santa Barbara, Calif, USA, October 2006.
- [4] 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, June 2005.
- [5] 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.
- [6] A. A. Argyros and M. I. A. Lourakis, "Tracking skin-colored objects in real-time," in *Cutting Edge Robotics*, pp. 77–90, Advanced Robotic Systems International, Vienna, Austria, 2005.
- [7] A. A. Argyros and M. I. A. Lourakis, "Tracking multiple colored blobs with a moving camera," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 2, p. 1178, San Diego, Calif, USA, June 2005.
- [8] C. Kerdvibulvech and H. Saito, "Real-time guitar chord estimation by stereo cameras for supporting guitarists," in *Proceedings of the 10th International Workshop on Advanced Image Technology (IWAIT '07)*, pp. 256–261, Bangkok, Thailand, January 2007.
- [9] O. Cakmakci and F. Berard, "An augmented reality based learning assistant for electric bass guitar," in *Proceedings of the 10th International Conference on Human-Computer Interaction (HCI '03)*, Crete, Greece, June 2003.
- [10] H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," in *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, pp. 85–94, San Francisco, Calif, USA, October 1999.
- [11] A. M. Burns and M. M. Wanderley, "Visual methods for the retrieval of guitarist fingering," in *Proceedings of the 6th International Conference on New Interfaces for Musical Expression (NIME '06)*, pp. 196–199, Paris, France, June 2006.
- [12] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34–58, 2002.
- [13] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, Upper Saddle River, NJ, USA, 2003.
- [14] K. Jack, *Video Demystified*, Elsevier Science, Oxford, UK, 2004.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

