

# Design and Implementation of a Real-Time Fingering Detection System for Piano Performance

Yoshinari TAKEGAWA, Tsutomu TERADA, and Shojiro NISHIO  
 Graduate School of Information Science and Technology, Osaka University  
 {takegawa, tsutomu, nishio}@ist.osaka-u.ac.jp

## Abstract

*Fingering is an important aspect of piano performance because it affects a pianist's musicality. If audience members, especially pianists, shared the real-time fingering of the performer in a concert, they would feel a sense of togetherness, and this could help them in learning techniques of professional piano performance. To realize such a possibility, the goal of our study is to construct a real-time fingering detection system for piano performance. Our system achieves real-time fingering detection by integrating a simple camera-based image-detection method and musical rules. We have developed a prototype system and evaluated its effectiveness by actual use.*

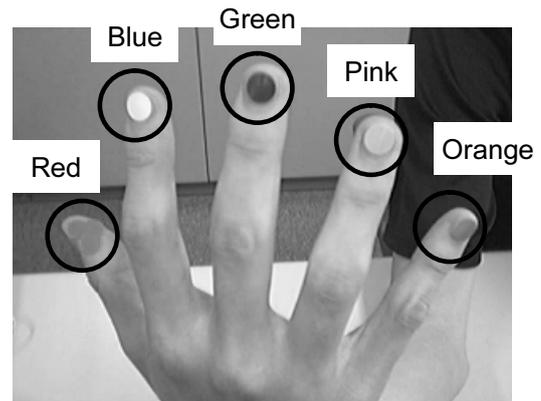


Figure 1: Color markers placed on the nails

## 1 Introduction

Pianists use many techniques to achieve their own desired musical expression in performance. Fingering is one of most important techniques because it affects a pianist's musicality (1; 8). Pianists explore the best fingerings to fulfill their ideal expressions, so there are many excellent styles of fingering, which are different for each pianist.

On the other hand, we can get little information on fingering only from musical scores and instructional books written by composers, professional pianists, or educators. Therefore, it is difficult for teachers/students to teach/learn fingering. In actual lessons, teachers check the actual performances of a student and advise him/her of better fingerings. However, in remote music lessons (10), teachers cannot check the details of fingering. In response to this problem, there is a demand for a real-time fingering detection mechanism for piano playing. This mechanism could also be used for various interactive applications such as a self-study support system that indicates fingerings or a performance education system that allows users to simultaneously learn the execution of playing with the fingering of a professional pianist.

Therefore, the goal of our study is to construct a real-time fingering detection system for piano performance. Our

system captures fingering by using a simple camera-based image-processing method, and musical rules of piano performance.

The remainder of this paper is organized as follows. Section 2 explains the design of our system, and Section 3 describes its implementation. Section 4 explains an evaluation and discusses considerations, and Section 5 describes related work. Finally, Section 6 presents conclusions and outlines future work.

## 2 System Design

There are two strict policies for constructing our fingering detection system:

- (1) **The system does not interfere with piano performance**
- (2) **Musical rules for piano playing are used to correct the results of fingering detection**

As regards (1), our system does not interfere with performance because it is supposed to be used in situations where a performer concentrates in a concert or a lesson. Therefore,

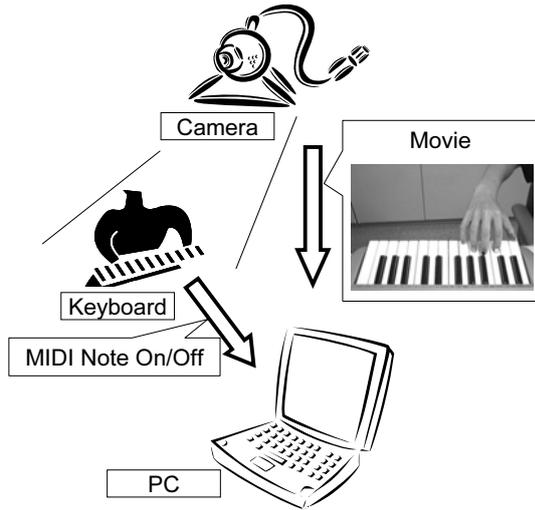


Figure 2: System structure

the performer does not attach any obstructive device to his/her fingers or hand. Assistive devices such as a switch attached to the ends of the fingers enable a system to detect fingering easily, but these devices constrain the motion of fingers. Our system detects fingering by using a simple image processing of color markers attached to the end of the finger nails. These color markers are thin stickers that do not interfere with piano performance and that give a almost no uncomfortable feeling in wearing.

As regards (2), our system corrects fingering detected by the image processing by using musical rules. These rules are defined by the features of piano performance, the hardware characteristics of a keyboard, and the characteristics of fingers. High-accuracy fingering detection is difficult only in the image processing, because the fingers frequently overlap each other; for example, the thumb is easily hidden by the other fingers in piano performance. Furthermore, a complicated algorithm requires a high computational cost, so the system cannot process the fingering detection in real-time. Therefore, our system captures fingering by a simple image processing and then corrects the fingering by using musical rules of piano performance.

Figure 2 shows the system structure. The PC analyzes and detects the fingering by capturing images of the keyboard and the markers and using MIDI data generated from the MIDI keyboard.

## 2.1 Outline of fingering analysis

Figure 3 shows the flow of fingering analysis. As a preparation, the system extracts the keyboard area from a camera

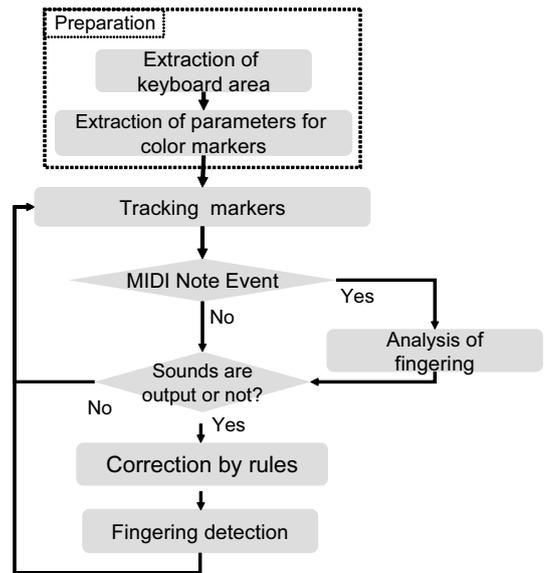


Figure 3: Flow of the fingering processing

image, and then it associates every pixel of the area with a key number. Next, it obtains the performance based on the color markers.

The status of fingers not used in keying is important for fingering detection because it helps to evaluate musical rules. Therefore, the system always tracks all markers and detects the key number under each marker.

## 2.2 Extraction of color markers

The system detects color markers on the figure of the player by comparing the HSV values from captured images with those from the parameters sampled in the preparation phase.

## 2.3 Fingering correction by rules

As explained in Section 2.1, since we detect the position of a marker by using a simple camera-based image-processing method, recognition errors frequently occur. In general, we use complicated pattern recognition techniques and image transformation techniques to improve recognition accuracy. However, they require a heavy computational cost, and then it is difficult to process images in real-time.

Therefore, our system corrects the detected fingering and predicts correct fingering when the system cannot detect fingering by using rules that are defined by the features of piano performance, the hardware characteristics of a keyboard, or the characteristics of fingers.

We define the following 4 rules:

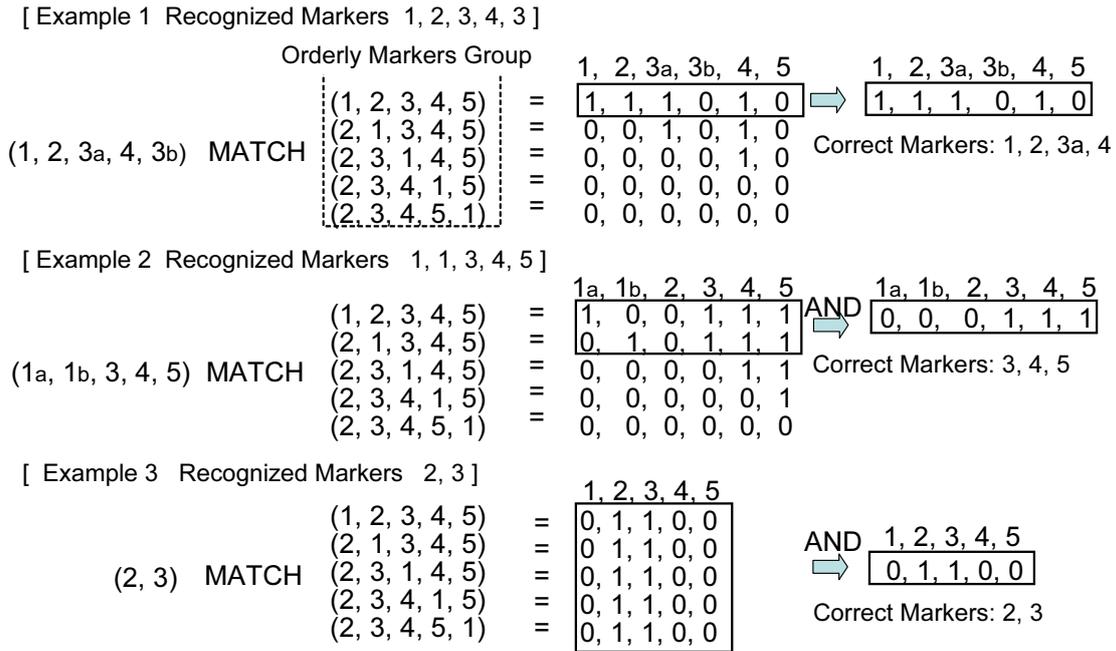


Figure 4: Example of correction of recognition markers

**Rule 1: The horizontal order of fingers 2 to 5 does not change.**

Rule 1 can detect the non-detection of a marker and correct the recognition error of markers. Note that Rule 1 supposes that the performer does not press multiple keys with only one finger at the same time nor press one key with multiple fingers.

The possible combinations of orderly markers, which are named OMG (Orderly Markers Group), are only the five patterns shown in Figure 4. The figures of OMG in Figure 4 and the markers recognized by the system are associated with the finger number. Subscripts of “a” and “b” are attached to distinguish the same markers which are recognized by the image processing.

The procedure of correcting in the recognized markers and predicting the markers from Rule 1 is performed in the following steps.

1. We define MATCH operator. The input parameters of MATCH are the array of recognized markers and OMG. MATCH calculates the highest concordance rate between the recognized markers and the OMG. Note that the order of markers array and OMG are maintained in the calculation of the concordance. MATCH outputs “1”, which is the conformable marker, and “0”, which is non-conformable marker.

2. The candidate of fingering is an element of OMG that has the largest number of “1” counts.
3. If there are multiple candidates, they are integrated by AND operations. As a result of these procedures, the markers attached with “1” are correct recognition, and the markers attached with “0” are undecidable.
4. If the key under the correct marker is pressed, the system can decide the fingering uniquely. Otherwise, the system predicts the fingering from the true markers and the pressed key.

We show three examples in Figure 4. In Example 1, the candidate of the fingering is unique, and the misdetection of marker 3b is corrected to marker 5. In Example 2, marker 1a and marker 1b are false markers, but if the key under marker 1a or marker 1b are pressed, the system identifies the fingering as fingers 1 or 2. In addition, in Example 3, the correct marker is only the markers of 2 and 3. If the key to the left of the key under marker 2 is pressed or that in the middle of the keys under marker 2 or 3 is pressed, the fingering number is decided as 1.

**Rule2: The keying finger does not change from pressing the key to releasing the finger from the key.**

When a certain key is pressed, this candidate of the finger of the pressed key is not the fingering that is recognized on

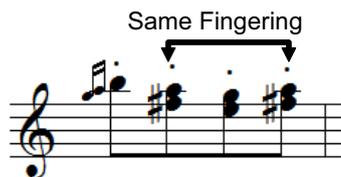


Figure 5: Example of Rule 3



Figure 6: Example of Rule 4

the other key. For example, assume the situation that there are two fingers on the same key, and the camera cannot detect the keying finger because it is hidden by the other finger. In this case, if the latter finger moves to another position before releasing the keying finger from the key, and the system can correct the fingering by this rule.

**Rule3: Piano players usually use the same finger for the same pitch tones that appear at a nearby site.**

This rule is activated only in the case where the system cannot distinguish the fingering by applying both Rules 1 and 2, since this rule is not always true. In that case, the system extracts the fingering by using the same note from the past four notes as shown in Figure 5.

**Rule4: The same musical structure usually takes the same fingering.**

Rule 4 is activated after Rules 1, 2, and 3 are applied. The system retrieves the phrase using the same musical structure as the recently played notes by exploring the playing history, and then it applies the fingering of the extracted phrase to the recent notes as shown in Figure 6.

The optimal threshold value of the number of tracing notes in Rule 3, the number of notes in a group in Rule 4, and the priorities of Rule 3 and Rule 4 should be considered in future works.

### 3 Implementation

We implemented a prototype of the fingering detection system as shown in Figure 7. We used the SONY VGN-S92PS (CPU 2.13 GHz, RAM 2 GB), whose platform is Win-

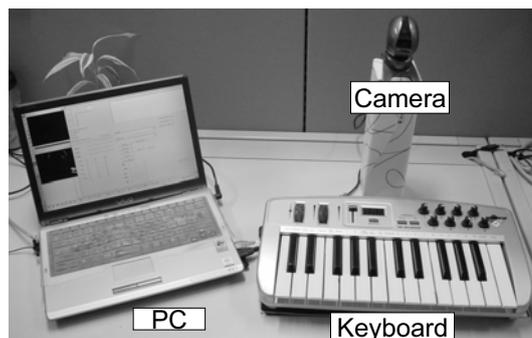


Figure 7: Snapshot of prototype system

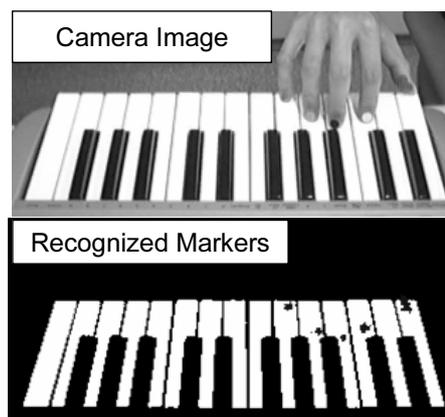


Figure 8: Camera image and result of image processing

dows XP as the PC, the Roland OXYGEN8 equipped with 25 full-sized keys, and the Creative WebCam Live! Motion (Resolution 320 x 240 pixels, 30 fps) as a camera. We used five commercial colored stickers, and applied a sticker to each nail of the right hand as shown in Figure 1. We used red, blue, green, pink and orange stickers. We placed the camera in a position that afforded a good view of the stickers. If the recognition area of the camera becomes large, a marker will become relatively small and the recognition rate will become low. Therefore, the recognition area of this prototype is limited to two octaves. Figure 8 shows a camera image and the result of image processing. We implemented the system using Microsoft Visual C++ .NET 2003 and Intel OpenCV Library.

### 4 Evaluation and Considerations

We conducted an evaluative experiment using three pianists to look into the effectiveness of the prototype and the

Table 1: Trial pieces

Music	Composer	Tempo (beat/minute)	Abbreviated name
Traumerai [Kinderszenen]	R. Schumann	64	Music A
Frohlicher Landmann	R. Schumann	107	Music B
Turkischer Marsch	W. A. Mozart	122	Music C

recognition rate. We compared the recognition rates of the case of applying rules with the case of not applying rules. We also investigated the effectiveness of musical rules.

## 4.1 Experimental procedure

### Trial pieces

Table 1 shows the trial pieces and their profiles. All trial pieces were played with only the right hand passage from the beginning to bar 20.

Music A has the slowest tempo among the trial pieces. However, examinees must cross the thumb and other fingers to express the slur indicated on the score. Moreover, Music B has many instances of two or three chords played in staccato. Therefore, the system needs to recognize multiple fingerings at a moment. In addition, Music C has the fastest tempo and is the most difficult piece, since it has many demiquavers and grace notes and thus much crossing of fingers.

### Examinees

Three examinees took part in this experiment. Everyone could play the trial pieces well at the indicated tempo. Two of them study at a college of music and their major is piano playing, while the other examinee does not study at a college of music but has been playing piano for 22 years.

### System structure

The prototype system maintains logs of MIDI Note On/Off, the time of pressing/releasing a key, and the fingerings in both cases of applying musical rules and not applying them. In addition, we recorded the hands of examinees with a digital video camera to collect fingerings.

### Flow

The examinees played all pieces twice. We instructed all examinees to play with their favorite fingering and at the tempo written in the score.

## 4.2 Experimental results

Figure 9 shows the fingering recognition rate when applying rules and when not applying them. The average rate in the case of applying rules is always higher than in that of not ap-

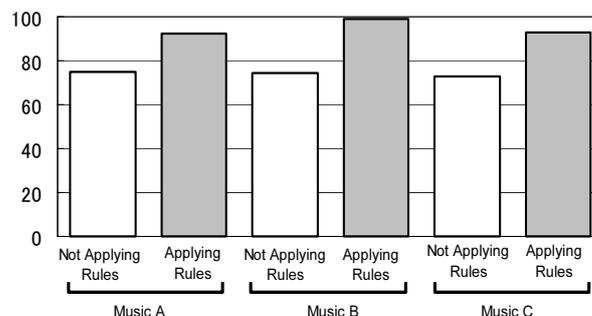


Figure 9: Recognition rates

plying them. Using musical rules, the system can recognize the fingering during the crossing of fingers, the simultaneous detection of two or more fingerings, a fast tempo, and complex execution. On the other hand, the average rate in the case of not applying rules is only 74%. We clarified the significance of the recognition rate at a level of 5%.

In the following, we describe three examples of how incorrect recognition was solved by musical rules. The asterisks in Table 2 designate the rules that contribute to the problem solving.

### Case 1

In this case, the recognized markers are out of order. This problem is resolved by Rule 1, which detects impossible crossing.

### Case 2

The marker of finger 1 or finger 5 cannot become clearly detected, since those fingers are hidden by other fingers. Therefore, in the case of not applying rules, those fingers cannot be recognized, thus creating a major problem for detecting errors. There are many detection errors of finger 1 in, for example, “E note in bar 2 of Music A”, “the staccato chord in Music B”, “A note in bar 1 of Music C” and “last C note in bar 2 of Music C”.

On the other hand, in the case of applying rules, Rule 1 corrects the fingering from other detection markers and MIDI

Table 2: Cases of applying rules

Case	Item	Rule 1	Rule 2	Rule 3	Rule 4
1	Improvement of marker recognition	*			
2	Non-detection of marker	*	*	*	*
3	Covering of finger	*	*		

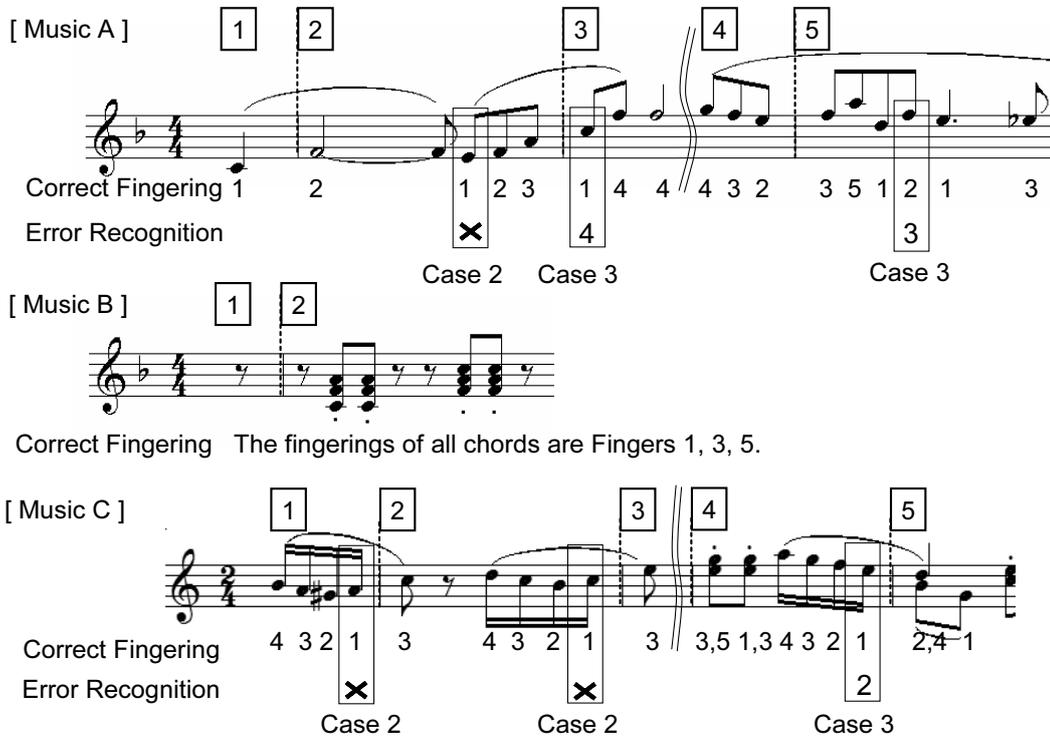


Figure 10: Details of recognition results

Note On/Off. Moreover, when the fingering is not detected at the key press, Rule 2 tracks all markers, and a marker that is detected on an other key is eliminated as a fingering candidate. In this way, we can improve the recognition rate of fingering when applying both rules. In addition, the system sometimes cannot recognize fingering with Rule 1 or Rule 2 in the staccato chord of Music B. The system complements fingerings by Rule 3, which is the prediction with old fingering data.

### Case 3

When “first C note at bar 2 of Music A” in Figure 10 is pressed, the system recognizes that finger 4 is in advance of fingering 2 by accident. Moreover, when “last E note at bar 4 of Music C” is pressed, the system cannot detect finger 1. After that, because finger 2 or finger 3 shows up in advance

of finger 1, the system recognizes that the fingering is finger 2 or finger 3, not finger 1.

The case of applying rules in Music A, the system recognizes that the first C note is pressed with finger 4 at first. However, finger 4 moves to play the next F note. At this point, the hidden finger 1 appears. The system eliminates finger 4 as a fingering candidate of the C note by Rule 2 and recognizes that finger 1 presses the C note.

In addition, in Music C, when the E note is pressed, the system recognizes that the E note is pressed with finger 1 by Rule 1, and thus finger 2 or finger 3, which covers finger 1, is eliminated as a fingering candidate by Rule 2.

### Case of failure

The “second F note at bar 5 of Music A” in Figure 10 is pressed with finger 2, but even in the case of applying rules,

the system recognizes finger 3. This is an example of Rule 2 not working well in Case 3. Rule 2 rectifies a false fingering when the false fingering, which covers the true fingering, moves to another key. However, the false finger (finger 3) does not move from the pressing of the F note to its release in this case, and thus the system misidentifies the fingering.

### Processing speed

Our system uses a simple image-processing method and corrects fingering with rules. Therefore, we can recognize the fingering in real time. The average processing time per frame is 20 msec, and the average frame rate of the camera that the prototype uses is 30 frames per second. The prototype system can complete the image processing within this frame rate.

As future work, we will use a camera that can recognize a wider keyboard area and reduce the processing cost of the system by selecting rules dynamically depending on conditions.

### Improvement of marker recognition rate

In this prototype, we use commercially available stickers as markers. However, these stickers are glossy and reflect the lighting. The marker recognition rate decreases for this reason.

As future work, we will attach a polarized filter to cut the reflection at the camera and use stickers that cause less reflection. Moreover, we will change the algorithm of the system so that it recognizes not only the marker but also the flesh color of the finger. Even if it cannot recognize the marker, it can obtain the existence of the finger and correct the recognition of the marker.

## 5 Related Work

There are a few research works whose main goal is fingering detection at the keyboard. These detect fingering with the use of Data Glove (9), FingerRing (2) and Lightglove (4), and so on. However, sensors attached around the fingers, circuits located on the wrist, and various cables disrupt the performance of pianists.

On the other hand, there are many research works that generate fingering automatically. For example, such systems generate fingering by using of the difficulty of performing an interval with finger pairs, the moving distance of a hand, and the structure of music (3; 6; 7). These approaches generate an optimal fingering that strains the hand and fingers of the pianist with static information, and they do not strive for real time recognition. This is different from our goal of detecting the fingering of performance in real time. Moreover, it is difficult for these studies to generate a musical fingering, since they have the practical demerit of generating fingering only

in monophony. However, these studies define some fingering rules, and these rules have improved the fingering recognition of our system.

In studies of hand gestures, proposed systems detect the end of the finger with cameras using RGB output and infrared cameras (5). Recognition in these research efforts are based on the camera. Such works do not pursue improvement of the recognition rate by using features of image processing, as our study does.

## 6 Conclusions

In this study, we have constructed a fingering detection system that uses color markers placed on the finger nails and a camera. The proposed system does not use a complex image processing to actualize real-time processing. It improves the fingering detection rate with the use of rules defined from the features of a performance, the fingers, and the keyboard. From the results obtained from a prototype system, our system can detect fingering with a high recognition rate.

Future work includes the extension to both hands and evaluative experiments based on various skill levels of piano execution.

## Acknowledgments

This research was supported in part by The 21st Century Center of Excellence Program “New Information Technologies for Building a Networked Symbiotic Environment.”

## References

- [1] Bernstein, S.: *With Your Own Two Hands: Self-Discovery Through Music*, 1997.
- [2] Fukumoto, M. and Tonomura, Y.: “Body coupled FingerRing” : wireless wearable keyboard, *Proc. of Computing Systems (CHI '97)*, 1997, pp. 147–154.
- [3] Hart, M. and Tsai, E.: Finding Optimal Piano Fingerings, *The UMAP Journal*, Vol.21, No.2 (2000), pp. 167–177.
- [4] Howard, B. and Howard, S.: Lightglove : Wrist-Worn Virtual Typing and Pointing, *Proc. of the Fifth IEEE International Symposium on Wearable Computers (ISWC '01)*, 2001, pp. 172–173.
- [5] Kolsch, M., Turk, M., Hollerer, T. and Chainey, J.: Vision-Based Interfaces for Mobility, *Proc. of the 1st Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous '04)*, 2004, pp. 86–94.

- [6] Parncutt, R., Sloboda, J. A., Clarke, E.F., Raekallio, M. and Desain, P.: Refinements to the Ergonomic Model for Keyboard Fingering of Parncutt, *Music Perception*, Vol. 18, No. 4 (2001), pp. 505–511.
- [7] Parncutt, R.: Physics and cognition of a virtual pianist, *Proc. of International Conference Music Association (ICMC '97)*, 1997, pp. 15–18.
- [8] Miller, M.F.: Piano Fingerings and Musical Expression, *Significance for Interpretation and Performance, Society for Music Theory, Baton Rouge*, 1996.
- [9] The homepage of Data Gloves.  
<http://www.vrealities.com/glove.html>
- [10] The homepage of YAMAHA Remote Music Education System.  
<http://www.yamaha-mf.or.jp/onken/soft/theme5e.html>