

DEVELOPMENT OF A REAL-TIME GESTURAL INTERFACE FOR HANDS-FREE MUSICAL PERFORMANCE CONTROL

Klaus Petersen

Graduate School of Advanced
Science and Engineering,
Waseda University, Japan

Jorge Solis, Atsuo Takanishi

Department of Modern Mechanical Engineering,
Humanoid Robotics Institute,
Waseda University, Japan

ABSTRACT

Our research aims to develop an anthropomorphic flutist robot (WF-4RIV) as a benchmark for better understanding the interaction between musicians and musical performance robots from a musical point of view. As a long-term goal of our research, we would like to enable such robots to play actively together with a human band, and create novel ways of musical expression. For this purpose, we focus on enhancing the perceptual capabilities of the flutist robot to process musical information coming from the aural and visual perceptual channels. In this paper, we introduce, as a first approach, a hands-free gesture-based control interface designed to modify musical parameters in real-time. In particular, we describe a set of *virtual* controllers, that a composer can manipulate through gestures of the body or a musical instrument. The gestures are identified by 2-D motion sensitive areas which graphically represent common control interfaces used in music production: trigger pads and faders. The resulting information from the vision processing is transformed into MIDI messages, which are subsequently played by the WF-4RIV. In order to verify the effectiveness of the proposed gestural interface, we performed experiments to control a synthesizer and then, to musically interact with the WF-4RIV. From the experimental results we concluded that our method satisfies the technical and idiosyncratic requirements for being a suitable tool for musical performance and composition.

1. INTRODUCTION

Since 1990 at Waseda University, we have been researching the Anthropomorphic Flutist Robot (WF-4RIV), which has been developed in order to - from an engineering point of view - clarify the human motor control that is required for playing the flute [13]. Since then, several improvements of the mechanical design of the emulated organs have been realized. As a result of the efforts that have been spent on imitating human flute playing, the research on the flutist robot has focused on:

1. Understanding human motor control [13].
2. Enabling communication between robots and humans in an emotional context [14].

3. Proposing novel applications for humanoid robots [15].

As a result, this research has been actively contributing to a diverse range of fields such as: Human-Robot Interaction (HRI), Music Engineering (ME), etc. The work on the Waseda Flutist Robot is therefore characterized by multidisciplinary research. It demands further efforts to develop a mechanical device, which is not only capable of reproducing human flute playing, but also of autonomously reacting to external input.

In this context, a musician robot is not merely used as a sophisticated MIDI instrument (*Passive Player*); but it is also able to display higher-level perceptual skills to autonomously decide how to react to stimuli from musical partners (*Active Performer*).

Such a kind of extended abilities will enable the Waseda flutist robot not only to interact smoothly with the members of a musical band, but the robot will also be able to assist musicians during the process of creating and composing new ways of musical expression.

The communication of humans within the context of a musical band represents a special case of interaction. Rhythm, harmony and timbre of the music being played represent the emotional states of the musicians. Thus, the development of a musical performance robot that is able to participate in such a kind of interaction, will help us to better understand some of the essential aspects of human communication.

A large amount of research work on musical robots has been accomplished ([7], [16]). Focusing especially on the interactive aspect of the field the GuitarBot and Robotic Musicianship projects are to be mentioned ([12], [17]). The GuitarBot project is a robotic musical instrument composed of four independent MIDI controllable single-stringed movable bridge units. The robot acts as an automatic musical instrument that can be controlled by a human or by a sequencer. Using specialized mechanics the robot can emulate several styles of actuating a guitar string (bowing, plucking etc.). This allows the instrument to automatically generate a sound that is very similar to the sound of a human playing the guitar. The robot is controlled by MIDI commands sent by a keyboard or host computer. The Robotic Musicianship project presents the development of a robotic percussionist. The goal of the

project is not only to construct a robot that is able to play an instrument according to sequencer commands, but to design a machine with a combination of musical, perceptual and interaction skills. To accomplish that the developed robot musician is able to analyze the acoustic actions of live human players. The robot uses this information to produce acoustic responses in a physical and visual manner.

In our case, we design an interaction method for the anthropomorphic flutist player. While working on this humanoid robot, it is part of our research goal to concentrate on developing the different parts of the robot, resembling their human counterparts as closely as possible. This premise encourages us to recognize human gestures visually, using the two cameras built into the head of the robot. As a long-term goal of our research, we would like to enable musical robots to play actively together with a human band, in order to create novel ways of musical performance. For this purpose, we focus on enhancing the perceptual capabilities of the flutist robot to process musical information coming from the aural, visual or tactile perceptual channels. In this paper, we introduce as a first approach a hands-free gesture-based control interface designed to modify certain musical parameters of the performance in real-time.

The interaction system we present in this paper is an extension to the Waseda Flutist Robot WF4-RIV. The WF4-RIV itself already emulates human play of the flute to an advanced degree (level of an intermediate player). As opposed to purely sequenced or human-controlled musician robot like the GuitarBot, with our interaction system, we want to give the robot the ability to actively interact with its fellow musicians. This interaction not only happens acoustically but also visually. Although the overall research approach also covers the creation of an acoustic analysis system, in this paper, we concentrate on the visual interaction system. So far, as in the Robotic Musicianship project, mainly acoustic interaction of a musical robot and human players has been studied. Thus, in this particular part of our research we prepare for a more complete interactivity including visual and acoustic interaction to make playing together with the flutist robot a natural experience.

Specifically, we describe a set of *virtual* controllers that a composer can manipulate through gestures of the body or a musical instrument. The musician performs these gestures in front of two embedded cameras, that are attached to the eyes mechanism of the WF-4RIV. We prefer using a vision interface instead of utilizing physical controllers, considering that in a band with human partners normally the musicians do not touch each other to give cues. The flutist robot is a humanoid robot and one of its purposes is to emulate human behaviour. As a musician uses mainly his vision and hearing to perceive communication from others, we try to equip the robot with similar sensory abilities. In particular, the gestures are identified by defining 2-D motion sensitive areas which graphically represent common control interfaces used in music production: trigger pads and faders. The resulting information from

the vision processing, is then transformed into MIDI messages which can be evaluated by the WF-4RIV. In order to verify the effectiveness of the proposed gestural interface, we first performed experiments to control a synthesizer and then applied our technique to musically interact with the WF-4RIV.

Up to now, several methods for recording user gestures that do not rely on a camera interface have been proposed ([5], [6]). Also similar principles as the one used in our research have been documented. In [18] and [19] different kinds of tools have been proposed to generate musical data from vision processing. Although the two commercially available systems EyeCon [4] and "A Very Nervous System" [2] are generally able to output MIDI data from camera input, the target for these systems seem to be dance-performances. In contrast, we tried to optimize our system to suit the needs of a person who wants to interact with a device musically. For this reason we designed our virtual controllers similar to ones that have been used in mechanical MIDI controller devices.

2. ACTIVE PERFORMANCE ROBOT

A human player in a band receives information from the other musicians mainly through his visual and acoustic perception. To enable the flutist robot to interact with human players in a similar way, we equipped the robot with stereo cameras and stereo microphones.

Due to the complexity of the task, we have several requirements for our interaction system. Through these requirements, we want to guarantee reliable performance in a realistic environment. As we deal with human-to-robot interaction, we have to consider (especially being an engineer with an engineer's perspective) that the users (ie. a musician or composer) might not have insight into the technical functionality of the system. Thus, we have to formulate our requirements in order to provide reliability and convenient usability for people with little technical knowledge. This leads us to the following demands for the interaction system in general:

1. **ROBUSTNESS:**
Performance environments may strongly vary. The system must reliably cope with the resulting difficulties.
2. **ACCURACY:**
The interaction partner's gestures have to be recorded with sufficient accuracy. It must be possible to control parameters in a way that the resulting modification matches the intended one.
3. **LOW LATENCY:**
For interaction with humans fast response times are essential.
4. **RESOURCE SAVING:**
Considering the interaction system as a whole, the single components (ie. vision, acoustic processing,

tactile interface) should use as little computing power resources as possible.

5. USER FRIENDLY:

There should be no need for the interacting person to be previously instructed. The interaction should be intuitive, allowing the user to get acquainted with the system by learning-by-doing.

6. NO PHYSICAL CONSTRAINTS:

No necessity for the user to be physically equipped with any special device (ie. markers for motion detection or wire connections from the user's instrument to the robot). Musicians will use different instruments. Interaction should be possible regardless of the type of the instrument.

In the first stages of our research, we have concentrated on creating a visual interface, that enables a human to control the robot through gestures with his hands or a musical instrument. Our emphasis here is to create an interface, that allows to control the musical expression of the robot with enough accuracy, but also robustly and computationally efficiently (at a later stage we would like among other things to perform audio processing on the same hardware). In the past we have exhibited the robot on several occasions and we found that each place had very different lighting and background conditions. As our gesture detection method should not only work in an optimized laboratory environment, but also in real environments; we need to find a way to cope with these varying ambiances. There might be a constantly changing background (ie. people passing by, stopping to watch the robot perform) and below-optimum lighting sources (ie. stage lighting facing into the cameras of the robot). Another difficulty is that the humanoid head of the robot can move during a performance. This requires our image processing algorithm to adapt rapidly to fundamental changes of background.

In the course of our research we explored different kinds of computer vision methods (particle tracking [3], real-time SIFT [8], shape analysis [11]) to track the movements of the human body. However we came to the conclusion that the use of a more simple method is most suitable for our purpose. The overall focus of our research is not only to develop a well performing vision system. Rather, we care about the concept of interaction as a whole, which also involves implementing other ways of sensor input. Therefore we focused on finding a method of computer vision that opens the desired communication channels, but is also usable in realistic performance situations while being technically easily maintainable.

About ten years ago the company Sony first introduced an interface extension for its gaming console Playstation called Eyetoy [9]. It enables players to control games by movements of their body in front of a small camera connected to the gaming console. The similarity of these games and our application with the Flutist Robot is that in both cases, we need to extract information about a person's movement in a very deliberate environment. Similar

to the principles applied in the Playstation games, in our research we use only the moving parts of a video for analysis. A related method called delta framing is employed in video compression (ie. MPEG I compression [10]). Thus, if we have a continuous stream of video images, for every frame we calculate a difference image with the previous frame. We threshold the resulting image, and thus create a b/w bitmap of the parts in the video image, that have changed from one frame to the next. To smoothen the result bit we calculate a running average over several of these images [20]:

$$p_r = \alpha * p_p + (1 - \alpha) * p_c \quad (1)$$

p_r : Average for the resulting pixel

p_p : Pixel at the same position in the previous difference image

p_c : Same pixel in the current image

α : Averaging factor

3. HANDS-FREE MUSIC CONTROLLERS

In a video game the information resulting from this delta-framing method might be used to destroy enemies or do a virtual boxing fight. However, we want to use it to control musical content. In music production, composers use switches and faders to control their electronic musical instruments, so we tried to model these controls in image space. The first controller we created is a simple push-button, in functionality similar to a drum pad of an Akai MPC drum machine [1]. These push buttons can be positioned anywhere in the video image. If a push button is triggered, a previously defined MIDI signal is sent to the flutist robot. At this stage of our research, the user can watch the input of the video camera on a monitor located beside the robot. On such a screen the position of the push buttons is graphically displayed. The buttons are drawn in a semitransparent color, so the area covered by the button is clearly defined and at the same time the video image beneath can still be seen. To detect if a button is switched on or off, we employ the algorithm in Fig. 1.

As a second controller we implemented a virtual fader, that can be used to continuously set a controller value. In this case, the position of a fader can be changed for example by a motion of the hand. For each change of the fader position, a MIDI controller message is sent to the robot. The fader slowly resets itself to a default position after it has been manipulated like a mechanical fader. This prevents a fader from remaining in an erroneous position that might have resulted from background noise (ie. a person moves in the background of the image, causing an undesired change of the fader position). A fader can be deliberately positioned in the image and orientated in any angle to allow the user to easily adjust it to his control requirements and physical constraints. To determine the position of a fader, we use the algorithm in Fig. 2. A second implementation using an algebraic calculation to determine

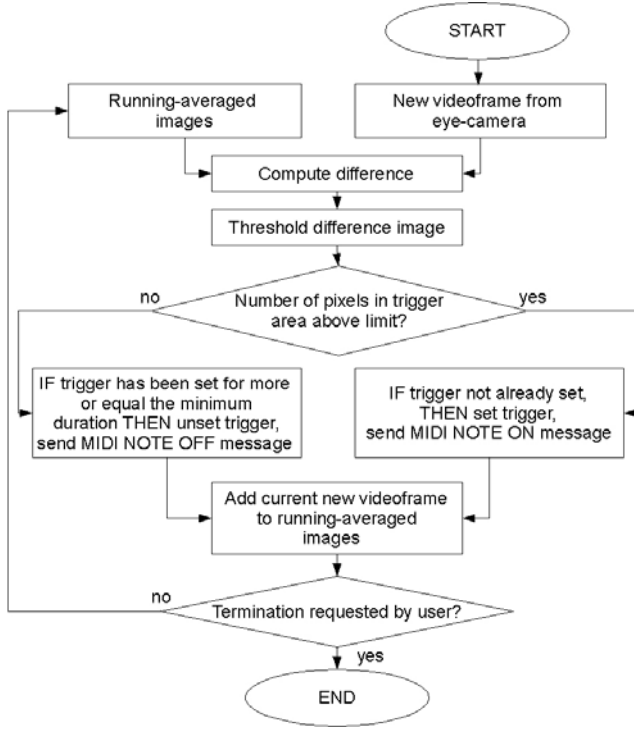


Figure 1. Flow-chart of algorithm to detect, if a pad has been triggered.

the new fader value v has also been implemented (see also Figure 3):

$$\sin \alpha = \frac{y_p - y_0}{|r|} \iff \quad (2)$$

$$\alpha = \arcsin \frac{y_p - y_0}{|r|} \quad (3)$$

$$\cos \beta - \alpha = \frac{|v|}{|r|} \quad (4)$$

$$v = \cos(\beta - \arcsin \frac{y_p - y_0}{|r|})|r| \quad (5)$$

\underline{r} : distance vector between (x_0, y_0) and (x_p, y_p)

\underline{v} : fader value vector

α : angle between r and the x -axis

β : orientation angle

The position of the baseline of the fader can be freely defined. For some control parameters of the Flutist Robot a *home* position (where the fader returns to, when it is not being touched) at zero might not be optimal. For example for variations in vibrato expression, the baseline position might be at a moderate vibrato amplitude value, allowing the fader to be moved to actuate slight positive and negative alterations to the vibrato amplitude.

4. EXPERIMENTS AND RESULTS

We performed different kinds of experiments to examine the technical and musical characteristics of our interaction interface. This section is divided into four parts. In

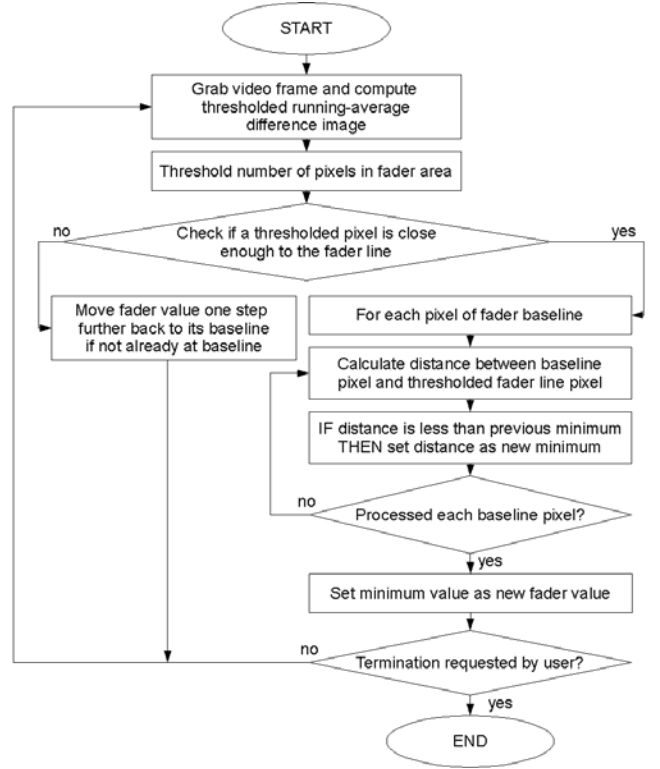


Figure 2. Flow-chart of algorithm to determine the current value of a fader.

the first part, we introduce the technical details of our experimental setup. In the second part, we collect data to determine, how robustly our system can cope with environment alterations. The third part illustrates how the different components of the interaction system are linked. It also verifies the accuracy of our method from the visual input into the camera to the musical output. In the fourth part we show the applicability of our system in a real-world style performance situation.

4.1. Experimental Setup

First, we tried to verify the technical requirements that we have claimed in section 2. Accuracy and robustness of our method need to be quantified. Therefore it was necessary to compare the output of our virtual control elements with a reference input. To provide this reference, we generated a simulation of a typical application on an auxiliary computer screen. The display of the screen was set up in front of the robot camera. The simulation consisted of a looped video clip representing a cluttered background scene. We chose different foreground images for each controller type. In case of the virtual faders, superimposed on the background scene is a contrastingly colored rectangle that moves back and forth horizontally. The rectangle resembled the hand or the instrument of the interaction partner, manipulating the virtual fader. The virtual pads were triggered by a slowly blinking rectangle.

The rectangles needed to be positioned on the screen in order to cover the whole area of the control, that they

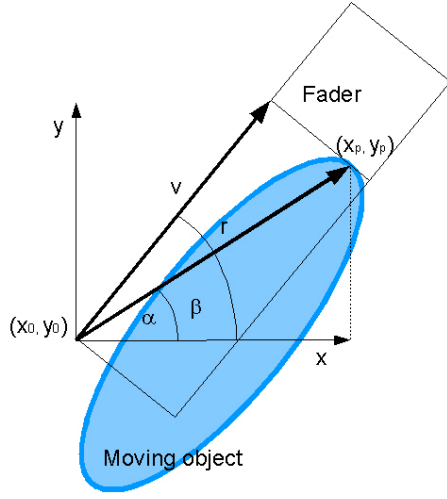


Figure 3. Schematic of a virtual fader

are supposed to trigger. The positioning of our screen in front of the camera could not be accurately determined. Thus, we used a calibration procedure, to find out which pixel position on the screen related to which position in the camera image. This relationship was then used to fix position and movement range of the foreground elements on the screen.

One might argue that the same effect might have been achieved by simply feeding a prerecorded video clip to the algorithm. However, in our case, on the one hand we want to examine the system as a whole. That also includes the actual image acquisition system. On the other hand the video that is being streamed to the algorithm needs to be synchronized to the data being sent to an external data logger. So it seemed the most straightforward method to generate the video input internally synchronized but fed to the camera externally in real-time.

To acquire a measure of the quality of our algorithm we then recorded the MIDI output of the virtual control elements and the position of the test rectangle in our fake scene on an external data logger. We plotted both curves and calculated error bars by subtracting the two data sets from each other. The resulting error gave us some insight about the accuracy and response latency of our system.

4.2. System Robustness

We quantized the robustness of our system by varying the testing technique described above to represent different environments. We simulated lighting changes by using different background movies and colors for the foreground rectangle. In the most simple case, the background would just be black and the foreground rectangle white. Here the contrast of the still and the moving parts of the scene were very strong. We accomplished more difficult scenes by using movies of different environments as backgrounds, choosing lower contrast colors for the foreground objects and also changing the external lighting of our experiment setup. To quantify the result, we compiled a ta-

Lighting Condition (Subject. Characterization)	Error (MIDI tics)	Delay (s)
Normal interior ceiling, cluttered but static background	12.0	0.1
Cluttered background with persons moving	14.3	0.1
Simulated overexposure	19.2	0.1
Simulated underexposure	23.3	0.1

Table 1. Experimental Results: Verifying the robustness of the proposed system.

ble of average fader response errors for the different scene environments. In our experiments we used a 2GHz PC for the image processing and a miniature firewire camera working at 30 frames per second. During operation our system used an average of 40% of the available computing resources. From the experimental results shown in Table 1, we may conclude, that lighting and background changes within certain limits do not strongly affect our system. The differences in average error from situation to situation are small. However, under more extreme conditions larger errors do occur.

4.3. Performance Control Accuracy

The aim of the second part of our experiments was to determine the applicability of our system in musical performance control. For this purpose, we arranged a simple performance in a music sequencer. We generated two monophonic MIDI sequences representing a Major7 chord each. We associated these patterns to be exclusively triggered by MIDI messages sent by our algorithm. In our program we set-up two virtual trigger pads, each representing one tone sequence.

We also prepared a virtual fader controlling the volume of an electronic synthesizer in the same way. In the first case, we used our input simulation as described before, to provide input for the video camera. This time, additionally to the MIDI information (Figure 4), we also recorded the sound output generated by the synthesizer. The resulting sound files were analyzed, extracting volume and pitch information. We plotted this data into a graph with the recorded MIDI and movement data. Thus, the relation between the pad actuation and the subsequently played note pattern, or the fader controller value and the output volume, can be seen. In Fig. 4a we see a piano roll view of the trigger pad MIDI reference (note values 70 and 72) and the MIDI response from the image processing (note values 69 and 71). Fig. 4c shows the corresponding changes in pitch. We observe a switch from chord one to chord two at 7.5s; from 16.5s chord one is played again. In Fig. 4b we can see the alteration of the fader controller value over

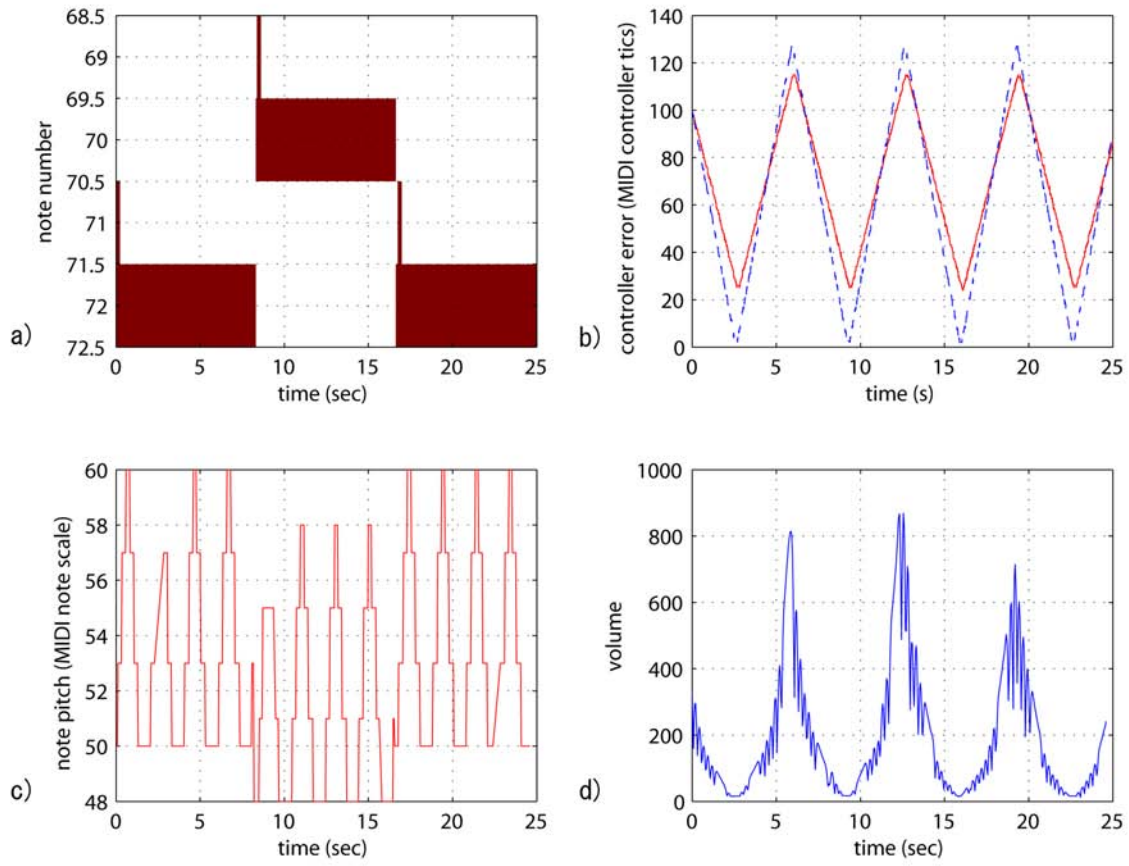


Figure 4. a) MIDI notes triggered by virtual trigger pads (long notes are reference signals, short notes are the pad responses). b) Recorded fader values (the blue/dashed line is the reference, the red line is the virtual fader response). c) Pitch of the recorded synthesizer output. d) Volume of the recorded synthesizer output.

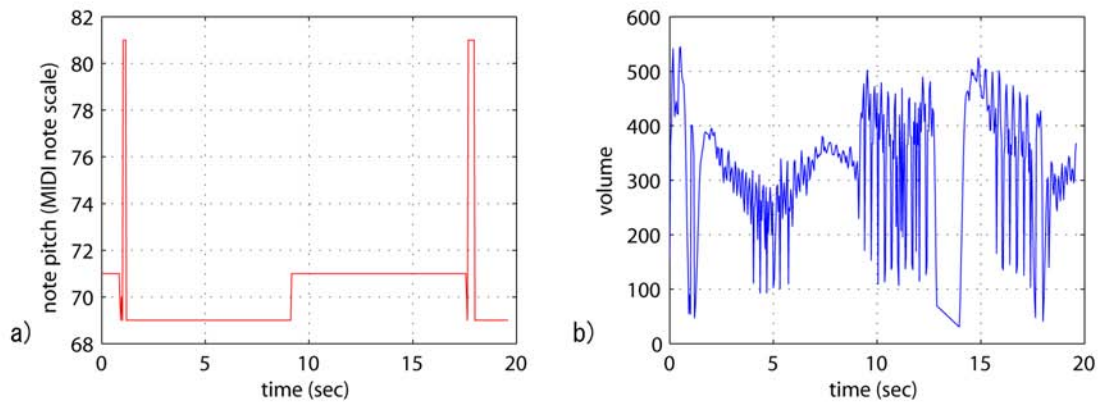


Figure 5. The analyzed sound data of the flute playing of the robot. a) Pitch information. b) Volume information; the parts of the curve with more vibrato show stronger amplitude variation.

Test Candidate	No. of Sessions	Avg. Vote
A	3	3.3
B	3	3.7

Table 2. Results of the experiments with test candidates A and B.

time. When looking at Fig. 4d, the maxima (6s, 12s, 19s) and minima (2.5s, 9s, 16s, 23s) of the controller movement and the output volume of the synthesizer occur at the same time. By this we show that the visual input to our system does closely relate to its acoustic output.

Extending the previous experiment we employed the flutist robot instead of the electronic synthesizer to generate the sound (Figure 6 and 7). In this case the trigger pads still switched between MIDI patterns, but the virtual fader controlled the vibrato of the sound of the flutist robot. Again we recorded the result in a plot (Figure 5). The MIDI input to the flutist robot consisted only of two tones (no arpeggio like in the previous experiment is played). The peculiarities of the tone generation of the robot can be examined in the pitch and velocity graphs. At the beginning of a new note the robot tends to *over-blow*, which results, for a short time, in the note being played one octave too high. The robot breathes through artificial lungs. At some point these lungs are empty and need to be refilled. This can be observed in the volume plot at about 12s, when the volume suddenly drops. The alterations of the vibrato amplitude (controlled by the virtual fader) can be examined in 5b. As in the previous experiment maxima (6s, 12s, 19s) and minima (2.5s, 9s, 16s, 23s) are visible.

The results show that our system succeeds to analyze the visual information received by the camera and translate it into musical output. Regarding the accuracy we see a quite large error, especially between the recorded fader data and the reference (the fader is *lagging*). This is a result of the running average we apply to the digitized video input. Depending on the application of the system the amount of averaging can be varied. Basically the averaging can be regarded as a kind of damping mechanism: The more averaging, the more smooth the recorded data will be, but as a trade-off, the response time of the system will be slower. However, we made the practical experience that this error does not strongly affect the applicability of the system.

4.4. Usability and Real-World Applicability

Later, we removed our setup for automatically providing input to the system. Instead we asked two test candidates to control the musical performance configuration. The first subject used his hands to manipulate the trigger pads and faders. The second test candidate used his instrument. Both candidates were wearing their normal clothes, the musical instrument was not especially prepared. Each candidate performed three sessions of two minutes each. After each session we asked the candidate to give a vote



Figure 6. The flutist robot interacting with a saxophone player.

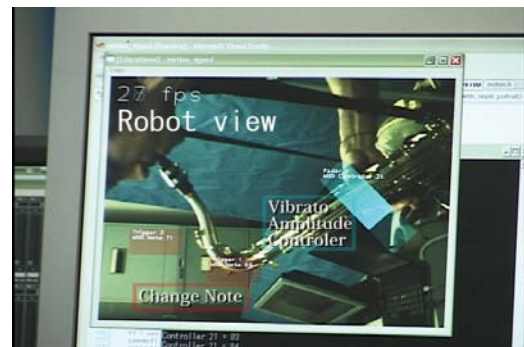


Figure 7. A relatively simple interaction controller configuration from the view of the robot. The camera that has been attached to the robot is rotated by 90° due to mechanical constraints. For that reason a tilted camera image appears on the screen.

(a score from 1 to 6, with 1 = best) about how he felt that he had been able to realize his musical plan (to put into practice his planned performance). With this result we wanted to give an impression of the usability of our system by uninstructed people in a musical performance environment (Table 2). It can be concluded that after three sessions the test candidates grew more acquainted with the usage of the system. In both experiment cases a mark of 2 was finally achieved.

5. CONCLUSIONS AND FUTURE WORK

The experiments show that our system is a valid approach for effective musical performance control. We examined the application of the interface in various lighting environments to prove its robustness. We found that its operation is not only limited to control the Waseda Flutist Robot, but can also be used with synthesizers in electronic music production.

The musical context that we have put our system into is momentarily quite sparse. We plan to conduct experiments with more complex musical material as soon as possible. By working closely together with musicians we will try to obtain a more detailed understanding of the perceptive process involved in musical interaction. The amount

of musical variation of the interaction with the robot depends on the amount of parameters that we can modify. As the flutist robot is a very complex mechanical construction for each parameter that we want to modify we need to especially consider the physical constraints given by the design of the robot. This is very similar creating musical material for a human, as a human also has certain bodily and also mental restraints. Taking these difficulties into account we would like to concentrate on parameter variations mainly regarding the lung and lip system of the robot. As a result we want to implement various sound-alterations that can be used in natural interaction with human players.

As future work, we want to utilize this interface, to act as one part of the interaction system, to enable the robot to interact with a human band. We plan to extend the proposed system in various ways: One problem in the current implementation is that the motions in front of the camera are only two-dimensionally perceived. Our algorithm cannot distinguish between an object in the foreground or the background of the camera. There are several ways to generate a depth-map of the image. As our robot has two eyes, we could use a bi-ocular method. So far we have experimented with tracking objects of interest using a Particle Filter in two parallel images. In further research we will try to combine this technique with the method proposed in this paper. Three-dimensional perception would also allow us to detect *mouse-clicking* like gestures (the user moves his hand over a switch sensitive area and pushes a virtual button by moving his hand forward).

6. REFERENCES

- [1] Akai MPC 2000, Akai Professional Japan, www.akaipro.com, 2007.
- [2] A Very Nervous System, Installation created by David Rokeby, "Arte, Tecnologia e Informatica", Venice Biennale, Venice, Italy, 1986.
- [3] Deutscher, J., Blake, A., Reid, I., "Articulated Body Motion Capture by Annealed Particle Filtering.", IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2000, pp. 2126.
- [4] EyeCon in "Lob der Anwesenheit", Dance Performance created by Frieder Weiss with Palindrome Intermedia Dance Group, Ludwigsforum Aachen, 2001.
- [5] Goto, S., "The case study of an application of the system 'BodySuit and RobotMusic: Its introduction and aesthetics' .", Proceedings of the International Conference on New Interfaces for Musical Expression, 2006, pp. 292-295, 2006.
- [6] Hasan, L., Yu, N., Paradiso, A., "The Thermenova: A Hybrid Free-Gesture Interface.", Proceedings of the 2002 Conference on New Instruments for Musical Expression 2002, pp. 1-6.
- [7] Kapur, A., "A History of Robotic Musical Instruments", Proceedings of 2005 the International Computer Music Conference, pp. 21-28, 2005.
- [8] Lowe, S., Little, J., "Vision-based Mobile Robot Localization and Mapping using Scale-Invariant Features.", IEEE International Conference on Robotics and Automation 2001, pp. 2051-2058.
- [9] Playstation Eyetoy, Sony Computer Entertainment Japan, www.eyetoy.com, 2008.
- [10] Richardson, I., "H. 264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia.", Wiley Publishing, 2003.
- [11] Shi, J., Malik, J., "Motion segmentation and tracking using normalized cuts.", Sixth International Conference on Computer Vision 1998, pp. 1154-1160.
- [12] Singer, E., "LEMUR GuitarBot: MIDI robotic string instrument", Proceedings of the 2003 conference on New interfaces for musical expression, pp. 188-191, 2003.
- [13] Solis, J., Takanishi, A., "An Overview of the research approaches on Musical Performance Robots.", International Conference on Computer Music 2007, pp. 356-359.
- [14] Solis, J., Chida, K., Suefuji, K., Takanishi, A., "Towards an automated transfer skill system.", International Computer Music Conference 2005, pp. 423-426.
- [15] Solis, J., Isoda, S., Chida, K., Takanishi, A., Wakamatsu, K., "Learning to play the flute with an anthropomorphic flutist robot.", International Computer Music Conference 2004, pp. 635-640.
- [16] Solis, J., Takanishi, A., "An overview of the research approaches on Musical Performance Robots", International Conference on Computer Music, pp. 356, 2007.
- [17] Weinberg, G., Driscoll, S., "Toward Robotic Musicianship", Computer Music Journal, Vol. 30, pp. 28-45, 2006.
- [18] Wanderley, M. and Battier, M., "Trends in Gestural Control of Music.", IRCAM Centre Pompidou, 2000.
- [19] Winkler, T., "Making Motion Musical: Gestural Mapping Strategies for Interactive Computer Music.", International Conference of Computer Music 1995, pp. 261-264.
- [20] Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.P., "Pfinder: Real-Time Tracking of the Human Body.", IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 780-785, 1997.