UnitKeyboard: An Easy Configurable Compact Clavier for Various Styles of Musical Expression

Yoshinari TAKEGAWA
Kobe University, Japan
take@eedept.kobe-u.ac.jp

Tsutomu TERADA
Kobe University, Japan
tsutomu@eedept.kobe-u.ac.jp

Masahiko TSUKAMOTO
Kobe University, Japan
tuka@kobe-u.ac.jp

Shojiro NISHIO
Osaka University, Japan
nishio@ist.osaka-u.ac.jp

ABSTRACT

Musical keyboard instrument has a long history, then we have many kinds of keyboards. Since conventional keyboards cannot change their hardware configuration such as the number of keys they have, we have to carry with a too big keyboard for playing music that requires only small diapason. To solve this problem, the goal of our study is to construct UnitKeyboard, which has only 12 keys (7 white keys and 5 black keys) and connectors to dock other UnitKeyboards. We can build various kinds of keyboard configurations by connecting a UnitKeyboard to others. Moreover, since UnitKeyboard has various functions such as the automatic settings by considering the relationship among multiple keyboards. We implement UnitKeyboard, and we discuss usability of suggested UnitKeyboard from reviews of some amateur and professional pianists that actually used the UnitKeyboard.

Keywords

Portable keyboard, block interface, Automatic settings

1. INTRODUCTION

Musical keyboard instrument has a long history, and then we have many kinds of keyboards (ex. piano, choir organ, and accordion). Moreover, there are many kinds of musical forms in classical piano performance: *solo* that is played by one performer, *piano duet* is played with a single piano by two performers, *piano duo* is played with two pianos by two performers and *ensemble* is played by multiple groups that consist of two or more musicians.

On the other hand, various kinds of electronic musical instruments have been developed. These instruments have many kinds of functions, such as diapason change, and tone change, and so on. Since conventional keyboards cannot change their hardware configuration such as the number of keys they have, we have to carry with a too big keyboard for playing music that requires only small diapason. Moreover, they have difficulty to adjust flexibly to various kinds of keyboard instruments, for example, musicians cannot play the music for the organ with a digital piano is equipped with 88 keys.

The goal of our study is to construct UnitKeyboard, which has only 12 keys (7 white keys and 5 black keys) and 4 connectors to dock other UnitKeyboards. We can build various kinds of keyboard configurations by connecting a UnitKeyboard to others. Moreover, since UnitKeyboard has various kinds of functions such as the automatic settings by considering the relationship among multiple keyboards, and intuitive controls using sensors and actuators. We can flexibly play music by using UnitKeyboard.

The remainder of this paper is organized as follows. Section 2 describes the design of UnitKeyboard. Section 3 describes the prototype and Section 4 describe considerations of UnitKeyboard. Finally, Section 5 describes conclusions and our planned future work.

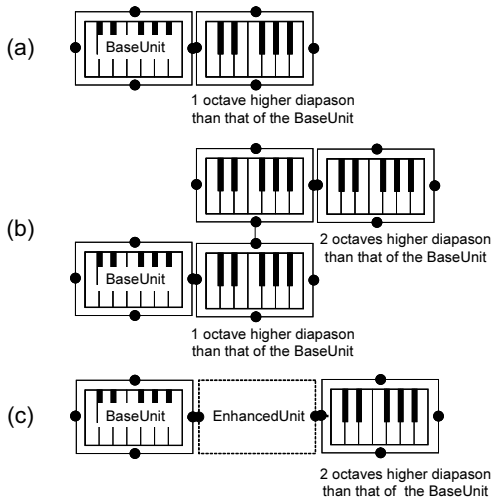

Figure 1: Combination examples of UnitKeyboard

2. DESIGN

UnitKeyboard is a keyboard instrument equipped with 12 keys and 4 connectors which can connect to other UnitKeyboards and build various kinds of keyboard instruments.

NIME08,June 5-8, 2007, Genova, Italy

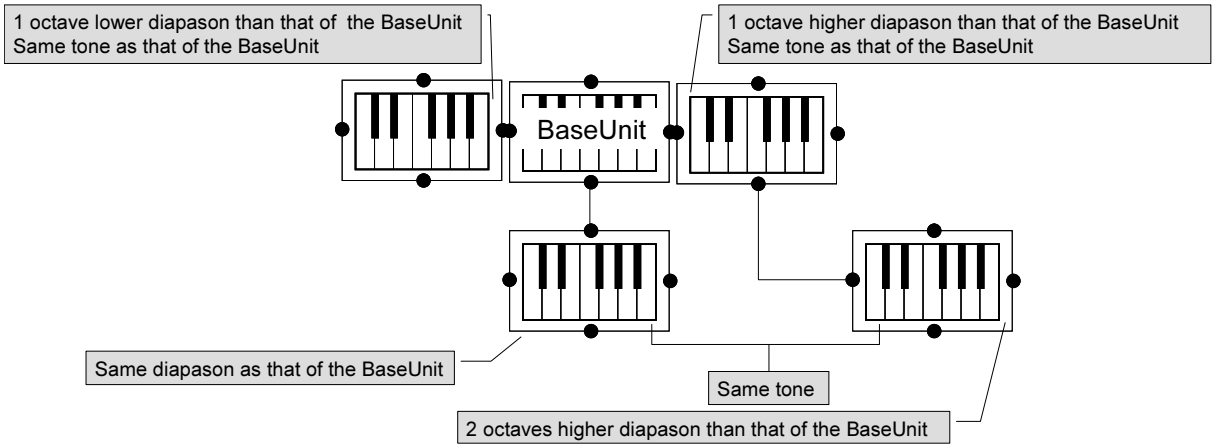

Figure 2: The mapping of tone and diapason based on connection

For example, we can construct a keyboard of two octave by connecting two UnitKeyboards in a horizontal direction, as shown in Figure 1-(a). Moreover, we construct an organ that has two manuals by connecting two UnitKeyboards in a vertical direction as shown in Figure 1-(b). UnitKeyboard makes a distance of diapason by inserting an EnhancedUnit that has various kinds of functions between UnitKeyboards as shown in Figure 1-(c).

2.1 Element Techniques

We discuss element techniques for realizing UnitKeyboard.

2.1.1 Automatic Settings

We can build various kinds of keyboard instruments by docking multiple UnitKeyboards. However, users need to set various kinds of settings for each UnitKeyboard. To reduce the setting cost, we suggest an automatic setting algorithm considering the characteristics of clavier.

Connection position A UnitKeyboard is equipped with one connector for each side: left, right, up, and down, to connect to other UnitKeyboards. Assignments of a tone and a diapason of each UnitKeyboard depend on a connection position. Generally, keyboard instruments equipped with single manual like piano have characteristics: the more left/right the position of a key, the lower/higher the pitch of the key, and all of the keys on the manual are same tone. Therefore, a UnitKeyboard connected to a BaseUnit in horizontal direction inherits the tone of the BaseUnit, and a diapason of the UnitKeyboard is assigned a diapason shifted one octave based on the diapason of the BaseUnit as shown in Figure 2. Note that BaseUnit is a leader and is base of settings such as the tone and the diapason among UnitKeyboards. On the other hand, a UnitKeyboard connected to the BaseUnit in vertical direction has the same diapason as that of the BaseUnit, and the tone of the UnitKeyboard is independent from that of the BaseUnit.

When the connection statuses among UnitKeyboards are changed, a case where assignments of settings are competition on UnitKeyboards arises as shown in Figure 3. In this case, the system gives priority to the previous settings of UnitKeyboards. And the system inform users of the competition.

Priority Between a BaseUnit and a non-BaseUnit has hierarchical relationship, that is, the settings of a non-BaseUnit

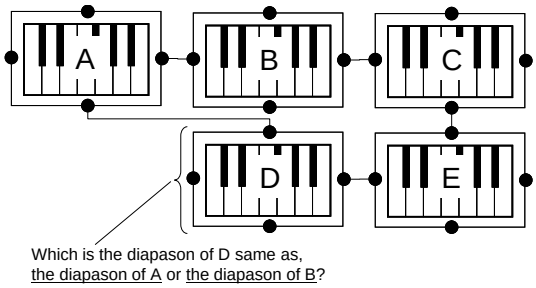

Figure 3: An example of rival diapason

inherit those of a BaseUnit. Moreover, in the case where performers form an ensemble that consists of multiple parts, there are part leaders or a director in the ensemble. Therefore, we defin priority as new relationship to achieve automatic settings. Our system automatically assign some settings of low priority UnitKeyboards based on settings of a high priority UnitKeyboard. For example, the priority of a BaseUnit is higher than that of a non-BaseUnit. The priority of UnitKeyboards possessed by a part leader is higher than that of others, and when the setting of UnitKeyboards of part leader is changed, that of UnitKeyboards of other automatically is changed too.

2.1.2 Real-Time reconfiguration

Since there is a case where configurations and connection statuses of UnitKeyboards are changed during the performance, the system needs to detect them and reconfigures settings of UnitKeyboards in real-time.

We discuss the system design to achieve high real-time processing from the views of data management and network composition.

Data management In a system of UnitKeyboard there are various kinds of system data: connection data to manage connection relationships among UnitKeyboards, setting data that means setting statuses of the diapason and the tone of each UnitKeyboard, keying data that is generated when keys of a UnitKeyboard are pressed/released.

If each UnitKeyboard manages own setting statuses, each UnitKeyboard sends all UnitKeyboards a connection change

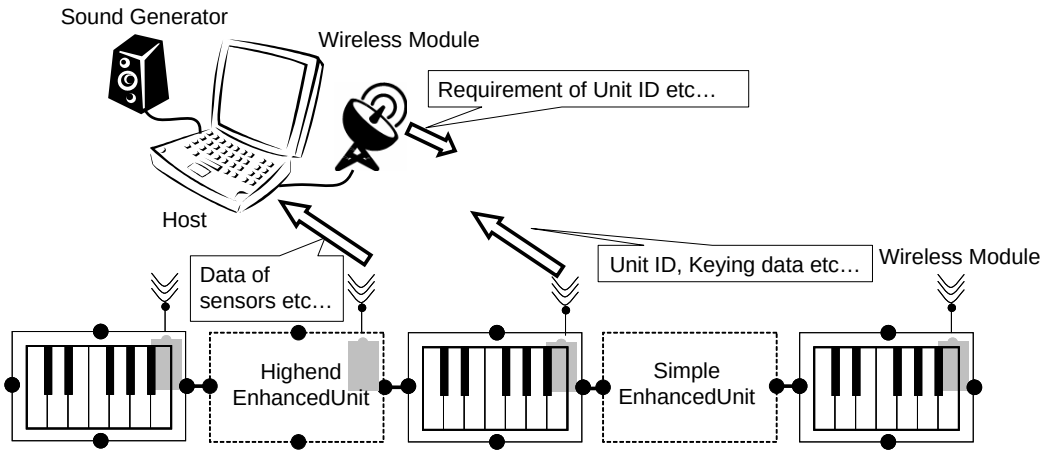

Figure 4: System structure

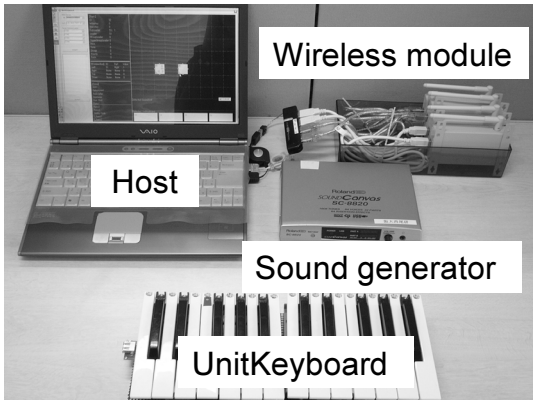

Figure 5: Prototype system

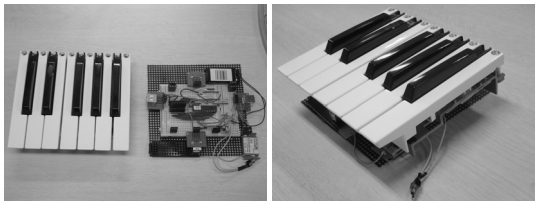

Figure 6: A snapshot of UnitKeyboard

message with each change in own connection statuses. Moreover, UnitKeyboards that are received the message necessitate the recalculation of own connection status and setting status. Because a spec of a CPU and a memory of a UnitKeyboard is generally low, it is difficult to recalculate in real-time.

Therefore, we ready "host computer" to calculate connection statuses, setting statuses of each UnitKeyboard in our system.

Network composition As methods to send system data from UnitKeyboards to the host, there are two methods: sending system data to the host via each UnitKeyboard, sending system data to the host directly by using a communication module on each UnitKeyboard. The former method need not equip a communication module on each UnitKeyboard, but since communication overheads increase, real-time processing is difficult.

Therefore, we adopt the latter method, and reduce the data to communicate between UnitKeyboards as possible.

3. PROTOTYPE SYSTEM

Figure 4 shows the structure of the prototype system that consists of a host, UnitKeyboards, and EnhancedUnits.

Table 1: Setting data

Item
Assigned ID of own Unit
The number of connector and the their position
ID of connected Unit
ID of BaseUnit
Assigned Diapason
Assigned Tone
Assigned Group
Assigned Priority
Assigned Form
Equipped input/output devices

Moreover, Figure 5 shows the snapshot of the prototype system, and Figure 6 shows the snapshot of a UnitKeyboard. We implemented the system using Microsoft Visual C++ .NET 2003, and we use SONY Vaio VGN-S92PS, whose platform is Windows XP as the host of the system, Allow7 UM-100 as a wireless module, Roland SC-8820 as a MIDI sound generator, and M-audio OXYGEN8 as the keyboard. OXYGEN8 had 25 keys but we cut it in half, finally it had 12 keys. We use a programmable integrated circuit (PIC) microcomputer (PIC16F873) as the control of UnitKeyboard and EnhancedUnit. The software on the PIC is programmed in C language on Microchip Technology's MPLAB.

3.1 Host

In the prototype, we use PC as a host. The functions of the host are as follows.

Management of setting data The host manages the setting data of each UnitKeyboard. Table 1 shows setting

Table 2: Control command
From a Host to a Unit
Requirement of Unit ID
Requirement of connector data
Requirement of input device data
Control of an output device

From a Unit to a Host
New Entry
Announcement of ID
Announcement of connector data
Connection Status
Announcement of keying data
Announcement of devices data
Announcement of input device data

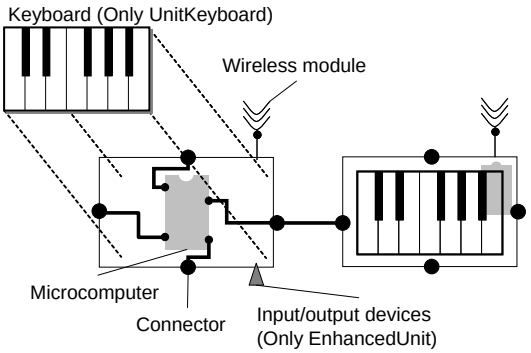

Figure 7: The hardware of Unit

data, and Table 2 describes control commands between the host and a Unit. Note that Unit includes UnitKeyboard and EnhancedUnit.

Management of connection statuses The host manages connection statuses of all Units, and can freely change connection statuses of all Units. Moreover, the host calculates setting data of each Units configuration from connection data of all Units.

Process of sound generation The host generates MIDI Note On/Off message based on the Units setting data and keying data received from a UnitKeyboard.

3.2 UnitKeyboard

The hardware structures of Unit are shown in Figure 7. A UnitKeyboard consists of PIC, a keyboard equipped with 12 keys, connectors to construct connection relationship on each side, and a wireless module to communicate to the host. UnitKeyboard has the following functions.

Connection establishment to the host A UnitKeyboard broadcasts "New Entry" command after its power-on, and when the UnitKeyboard receives acknowledgement from the host, it sends "ID" and "connector data", such as the number of connectors, to the host.

Announcement of keying data A UnitKeyboard sends keying data to the host, when the status of the UnitKeyboard keys is changed.

Announcement of connection data A UnitKeyboard sends connection data to the host, when the status of the UnitKeyboard connectors is changed.

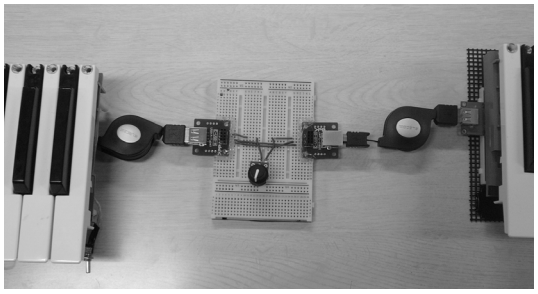

Figure 8: A simple EnhancedUnit with a variable resistance

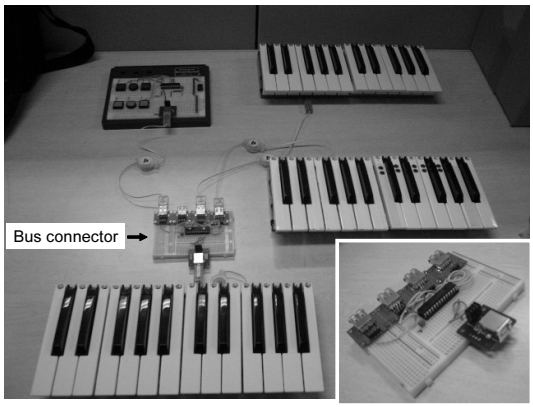

Figure 9: An EnhancedUnit with a bus connector

3.3 EnhancedUnit

EnhancedUnit has two models: simple model that only controls diapason of UnitKeyboard, highend model that equipped with sensors, actuators, and a wireless module to operate setting of UnitKeyboards. The former one is inserted between UnitKeyboards to make a distance of diapason, and it is simple structure that consists of two connectors and a variable electric resistance as shown in Figure 8. Since the connectors of a UnitKeyboard can measure the change of voltage that works with the number of a variable resistance, UnitKeyboards that interleave simple EnhancedUnit convert the number of voltage to the distance of diapason.

Figure 7 shows hardware of the highend EnhancedUnit. Main differences of EnhancedUnit and UnitKeyboard is that EnhancedUnit does not have keyboard and has various input/output devices. Highend EnhancedUnit has the following functions.

Connection establishment to the host EnhancedUnit broadcasts "New entry" command after power-on, and establishes connections with the host just like UnitKeyboard.

Announcement of connection data EnhancedUnit is monitoring the status of connectors, and it sends "Connection Status" command to the host when it detects changing of connection statuses just like UnitKeyboard.

Moreover, the EnhancedUnit equipped with multiple connectors work for a hub function that connects to various kinds of devices on a side shown in Figure 9.

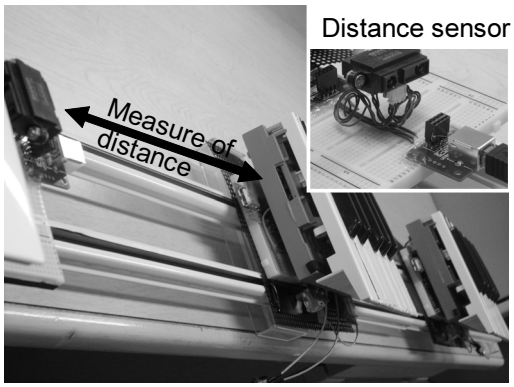

Figure 10: An EnhancedUnit with distance sensors

Announcement of input data from input devices An EnhancedUnit collects data of input devices, and informs the host of it according to the requirements of the host.

Control of output devices An EnhancedUnit controls output devices according to commands sent from the host.

3.3.1 Input/Output devices

We implemented the prototype of highend EnhancedUnits equipped with various kinds of input/output devices. We describe the examples of use with EnhancedUnits.

Distance sensor Users can control diapasons of two UnitKeyboards that interleave an EnhancedUnit equipped with distance sensors as shown in Figure 10. For example, the longer the distance between the UnitKeyboard and the EnhancedUnit, the higher the diapason of the UnitKeyboard.

Acceleration sensor Users control the tone of UnitKeyboards with a users' posture that is calculated and detected from data of the acceleration sensor as shown in Figure 11.

Mortar Users can move UnitKeyboards automatically by using EnhancedUnit equipped with motors attached a propeller and tires. For example, if musicians use an EnhancedUnit equipped with tired motor, they can complement the lack of diapason by autonomously moving a UnitKeyboard as shown in Figure 12.

4. CONSIDERATIONS

We discuss usability of suggested UnitKeyboard from reviews of 5 amateur pianists and 5 professional pianists that actually used the UnitKeyboard.

Many pianists played with this prototype. In this subsection, we describe the usability of the prototype[1].

4.1 Performance Evaluation

Visibility We checked the function hat automatically assignments the settings of UnitKeyboard by considering the relationship among multiple keyboards are work well. The host can settle competition of settings among UnitKeyboards well. Moreover, the suggested algorithm is intuitive from users review.

[1]You can see a demonstration video at http://www-nishio.ise.eng.osaka-u.ac.jp/~takegawa/DownLoad/UnitKeyboard/top_english.html

Figure 11: The assignment of tone based by postures

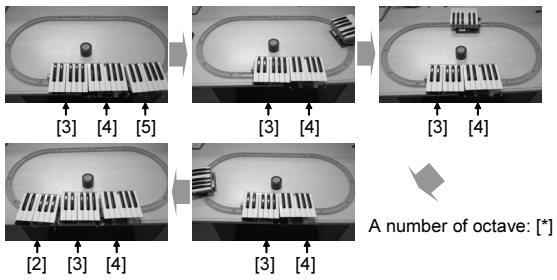

Figure 12: An EnhancedUnit with electric motor

On the other hand, it is difficult to recognize the absolute diapason of whole UnitKeyboards. Because users can visibility recognize connection relationships between UnitKeyboards, it is easy to recognize the relative diapason of each UnitKeyboard. In present implemetation, users cannot recognize the BaseUnit and the diapason of the BaseUnit visibility. Therefore, users check the diapason of UnitKeyboards by keying of UnitKeyboard directly.

As the future work, we implement EnhancedUnit with LEDs and a display to check the settings of UnitKeyboard with them.

Wireless connection vs. Wired connection We adopt a wireless connection method in the communication between the host and Units.

In the wireless connection method, although there is some delay from the keying to output sound. The delay is not so noticeable in music performance. However, the more the number of UnitKeyboard, the higher the possibility of packet loss and the more the delay.

On the otherhand, the delay of the wired connection less than that of wireless connection, it is difficult to change Unit configrations.

Figure 13: A snapshot of collaborative execution

Because both methods have advantage and disadvantage, we will conduct more detail evaluation, and investigate the situations that enable to utilize each method as the future work.

UnitKeyboard of one octave In this study, we employed one octave from C key to B key. This diapason effective in music of only C major or C minor.We can solve this mismatch in the keys by using the method of Mobile Clavier[7], which enables the diapason to be changed smoothly.

4.2 New performance

We suggest novel piano performance with using UnitKeyboard and EnhancedUnit.

For example, as shown in Figure 13, when the diapason of keyboard lacks during the performance, a musician solves it by borrowing a UnitKeyboard from another performer. Moreover, as shown in Figure 12, the appearance that a keyboard move automatically to demanded location is visual intersting. These performance have not only musical entertaiment but also visual attraction.

4.3 RELATED WORK

There have been many researches whose main goal has achieved scaling up the function by combining units. For example, there are the system, that is, users make a object by combining LEGO blocks and utilize the object in the game[1], users make a object by combining triangle boards and the system open web sites depending on the object automatically[2], users controls the programming with combined blocks[3]. Moreover, there are the block equipped input/output devices[4]. These targets are not musical interaction like our study.

On the other hand, a system whose target is music composition by combining blocks assigned mood music[5].

There are DoublePad/Bass[6] and Mobile Clavier[7] as a study whose target is improvement of mobility of acoustic instruments. DoublePad/Bass is base instruments using two PDAs, and musicians who can play an electric bass should be easily able to play it. Mobile Clavier enables the diapason to be changed smoothly by having additional black keys inserted. These instruments do not have a concept that is combining units, and do not construct various kinds of keybaord/string instruments

5. CONCLUSIONS

We have proposed and implemented UnitKeyboard, which can apply various kinds of keyboard instruments with combination of one octave keyboard. Moreover, UnitKeyboard has various functions such as the automatic settings considering the relationship among multiple keyboards, intuitive controls and new performance with using EnhancedUnit.

We intend to evaluate hardware characteristics of UnitKeyboard and the usability in the future.

Acknowledgments

This research was supported in part by a Grant-in-Aid for Scientific Research (A) (17200006) from the Japanese Ministry of Education, Culture, Sports, Science and Technology, by a Grant-in-Aid for Scientific Research from JSPS Research Fellowship, and by Hayao Nakayama Foundation for Science and Technology and Culture.

6. REFERENCES

[1] D. Anderson, J. Frankel, J. Marks, A. Agarwala, P. Beardsley, J. Hodgins, D. Leigh, K. Ryall, E. Sullivan, J. Yedida: "Tangible Interaction Graphical Interpretation: A New Approach to 3D Modeling", In Proceedings of SIGGRAPH 2000, pp.393–402, 2000.

[2] M. Gorbet, M. Orth, and H. Ishii: "Triangles: Tangible Interface for Manipulation and Exploration of Digital Information Topography", In Proceedings of CHI1998, pp.49–56, 1998.

[3] S. Hideyuki, K. Hiroshi: "Interaction-level support for collaborative learning: AlgoBlock an open programming language", In Proceedings of CSCL2002, pp.349–355, 2002.

[4] W. Ryoichi, I. Yuichi, A. Masatsugu, K. Yoshifumi, K. Fumio and K. Hideo: "The Soul of ActiveCube - Implementing a Flexible, Multimodal, Three-Dimensional Spatial Tangible Interface", In Proceedings of ACE 2004, pp. 173–180, 2004.

[5] N. D. Henry, H. Nakano, and J. Gibson: "Block Jam", In Proceedings of SIGGRAPH 2002, pp.67, 2002.

[6] T. Tsutomu, T. Masahiko, N. Shijiro: "A Portable Electric Bass Using Two PDAs", In Proceedings of IWEC 2002, pp. 286–293, 2002.

[7] T. Yoshinari, T. Tsutomu, T. Masahiko, N. Shijiro: "Mobile Clavier: New Music Keyboard for Flexible Key Transpose", In Proceedings of NIME 2007, pp. 82–87, 2007.