



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

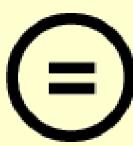
다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원 저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리와 책임은 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)



Master's Thesis

**Demonstration-Based
Vibrotactile Pattern Authoring**

Kyungpyo Hong (홍경표)

**Division of Electrical and Computer Engineering
Pohang University of Science and Technology**

2012



시연 기반 진동 패턴 저작 방법

**Demonstration-Based
Vibrotactile Pattern Authoring**



Demonstration-Based Vibrotactile Pattern Authoring

by

Kyungpyo Hong

Division of Electrical and Computer Engineering

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

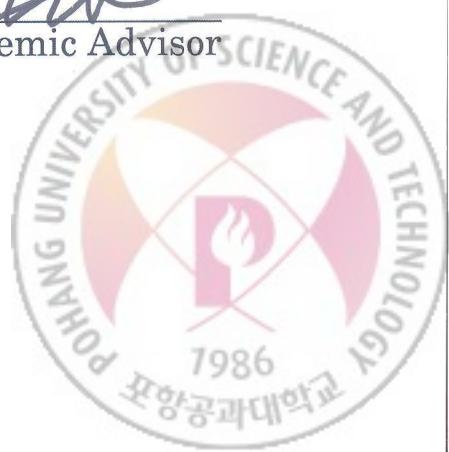
A thesis submitted to the faculty of
Pohang University of Science and Technology
in partial fulfillment of the requirements
for the degree of Master of Science
in the Division of Electrical and Computer Engineering

Pohang, Korea

2012 . 6 . 27 .

Approved by


Seungmoon Choi, Academic Advisor



Demonstration-Based Vibrotactile Pattern Authoring

Kyungpyo Hong

The undersigned have examined this thesis
and hereby certify that it is worthy of acceptance
for a master's degree from POSTECH

06. 27. 2012

Committee Chair

Seungmoon Choi (Seal)

Member

Seungyong Lee (Seal)

Member

Sung Ho Han (Seal)



MECE 홍경표, Kyungpyo Hong, 시연 기반 진동 패턴 저작 방법,
20100716 Demonstration-Based Vibrotactile Pattern Authoring,
 Division of Electrical and Computer Engineering, 2012, 41 p,
 Advisor: 최승문 (Seungmoon Choi). Text in English

Abstract

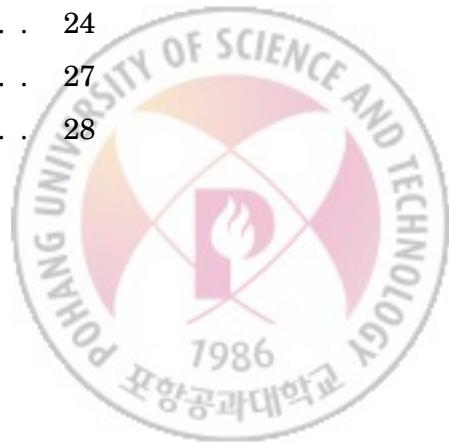
Despite the importance of vibrotactile rendering in a variety of devices, especially in mobile devices, few software tools that can facilitate vibrotactile pattern design have been developed. Most recent concept of the vibrotactile pattern design is producing patterns from simple motions on a touch screen: Apple Inc. released this type of editor in iOS5, but it only controls a vibration duration. Therefore, we propose a demonstration-based vibrotactile pattern authoring method, that the device generates vibrotactile patterns by imitating the touch input from the user. Our demonstration-based authoring method can control three major vibration properties - duration, amplitude and frequency - based on the user's touch motions on a touch screen. We also investigated the usability of our demonstration-based method and compared it with a conventional waveform-based method. For this experiment, we developed a new style of waveform-based editor for a touch screen. Although our demonstration-based method has several disadvantages depending on the type of designing task, the results showed that the demonstration-based method is funnier and easier to use by typical users than the waveform-based method. Moreover, it is especially good for creating a vibration pattern well matched to music.





Contents

1	Introduction	1
2	Vibrotactile Pattern Authoring Methods for a Touch Screen	6
2.1	Demonstration-Based Authoring Method	6
2.2	Waveform-Based Authoring Method	7
3	Implementation	11
3.1	Demonstration-Based Authoring Method	11
3.2	Waveform-Based Authoring Method	16
3.3	Common Things of the Two Editors	20
3.4	Desktop Program	21
4	Usability Evaluation	23
4.1	Methods	23
4.1.1	Participants	23
4.1.2	Apparatus	23
4.1.3	Experimental Conditions	24
4.1.4	Procedure	27
4.2	Results	28



5 Discussion	35
6 Conclusion	38
한글 요약문	39
Bibliography	40
감사의 글	42
PUBLICATIONS	44



List of Figures

1.1	Haptic icon prototyper	2
1.2	posVibEditor	2
1.3	Editor for tactile jacket.	3
1.4	MOTIV.	4
1.5	VibScore.	5
1.6	Custom vibration making in iOS5.	5
2.1	Concept of demonstration-based authoring method.	8
2.2	Concept of waveform simplification.	9
2.3	Concept of waveform-based authoring method.	10
3.1	Demonstration-based editor user interface.	13
3.2	Demonstration-based editor state machine.	14
3.3	Waveform-based editor user interface.	16
3.4	Waveform-based editor state machine.	17
3.5	Desktop program main screen.	19
3.6	Desktop program sending progress.	20
3.7	Desktop program flowchart.	22



4.1	Self-configured current amplifier.	24
4.2	Total apparatus configuration.	25
4.3	Instructions for training.	26
4.4	Instructions for task 1.	26
4.5	Waveform of the music in task 2.	27
4.6	Task completion time results. The asterisk indicates that the two design methods are statistically significant ($p < 0.05$). The error bars represent the standard errors.	29
4.7	Survey results of Task 1. The asterisks indicate that the two design methods are statistically significant ($p < 0.05$). The question marks indicate that the results are marginally significant ($p < 0.1$). The error bars represent the standard errors	30
4.8	Survey results of Task 2. The asterisks indicate that the two design methods are statistically significant ($p < 0.05$). The question mark indicates that the results are marginally significant ($p < 0.1$). The error bars represent the standard errors.	31
4.9	Self-evaluation results. The asterisks indicate that the two design methods are statistically significant ($p < 0.05$). The question mark indicates that the results are marginally significant ($p < 0.1$). The error bars represent the standard errors.	32
5.1	Concept of the hybrid method.	36



List of Tables

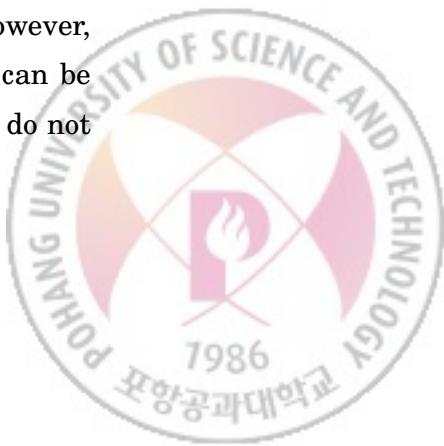
4.1 Results of pairwise T-test.	33
4.2 Summary of verbal comments.	34



Introduction

Vibrotactile rendering plays an important role in improving the usability of a variety of devices, especially mobile devices. Research has demonstrated the efficacy of vibrotactile rendering in providing tactile feedback for virtual button presses on a touchscreen [8], enhancing user experience as an integrated user interface (UI) component [5], and notifying various events by means of tactile icons [3]. In addition, vibrotactile rendering can escalate the users' immersion to multimedia contents and entertainment activities. All these technologies have been actively adopted in commercial mobile devices, including Android smartphones and tablets [4], iPhone [1], and iPod Touch accessory [2].

In parallel with these technological advances, software tools that can facilitate vibrotactile pattern design have been developed by a few research groups. Swindells et al. developed a vibrotactile pattern editor called a haptic icon prototyper (Figure 1.1) [10]. Ryu et al. developed posVibEditor (Figure 1.2) to enable multi-channel editing [9]. Lemmens et al. developed an editor for their tactile jacket (Figure 1.3) [7]. In particular, a commercial design tool, MOTIV Studio (Figure 1.4), has been used for years in the industry [4]. However, all of these tools require direct editing of vibration waveforms, which can be non-intuitive, difficult, and time-consuming for regular consumers who do not



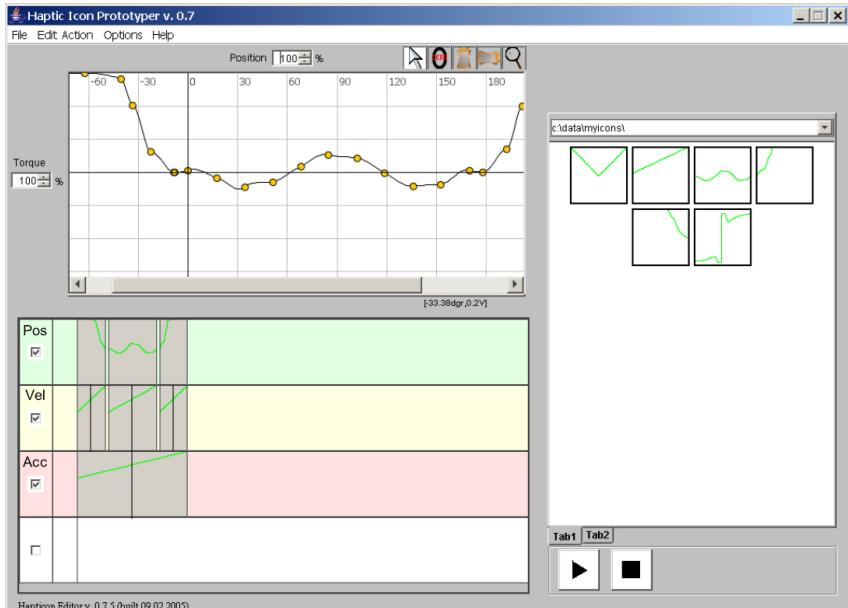


Figure 1.1: Haptic icon prototyper.

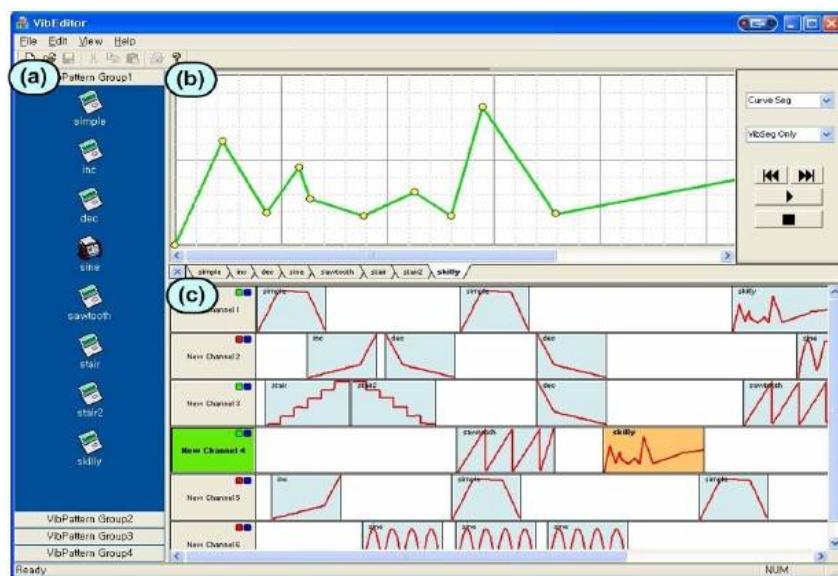


Figure 1.2: posVibEditor.



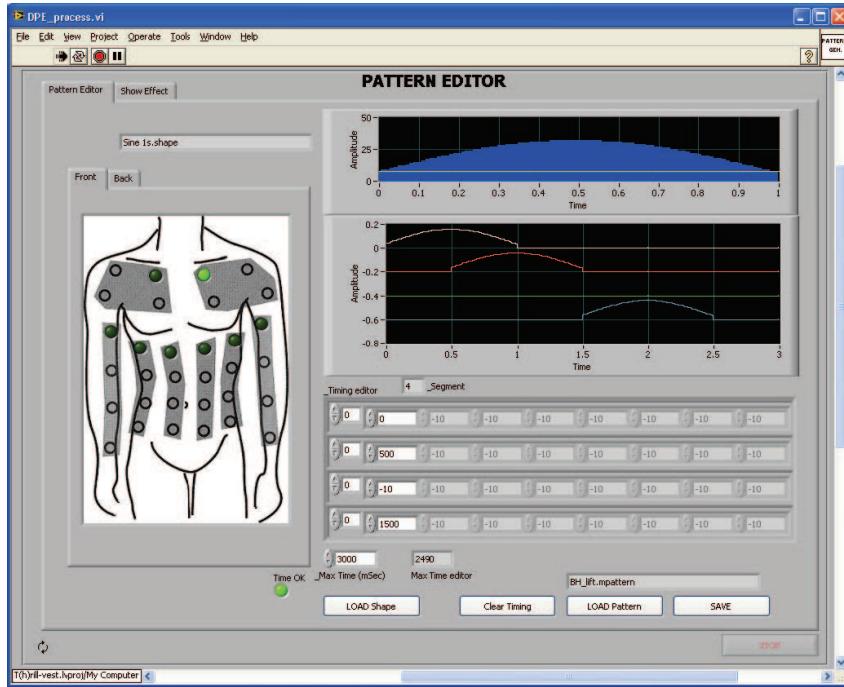
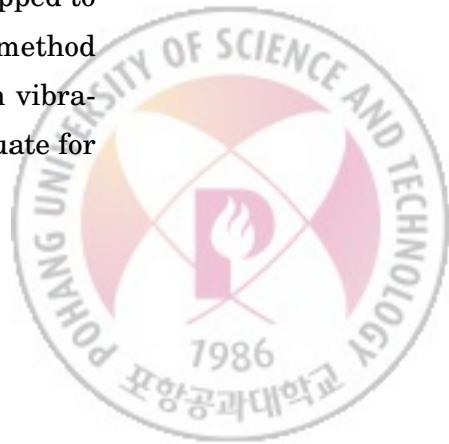


Figure 1.3: Editor for tactile jacket.

have knowledge on the underlying dynamics of vibration generation. To overcome this drawback, Lee et al. proposed a vibrotactile score (Figure 1.5) that uses musical symbols for vibrotactile pattern design and showed its improved performance and user preference with several representative design tasks [6]. This method, however, is tailored to the use of keyboard/mouse interface in a PC, and it may not be an ideal candidate for mobile devices that are equipped with a touchscreen.

In the touchscreen environment, recently released iOS5 includes a notable feature that allows users to design their own vibration patterns by simply tapping on a touchscreen (Figure 1.6) [1]. The duration of each press is mapped to the on-time of a vibration, like in rhythm games. In our experience, this method is extremely simple, intuitive, and easy to learn. However, the custom vibration function of iOS5, which controls only vibration duration, is inadequate for



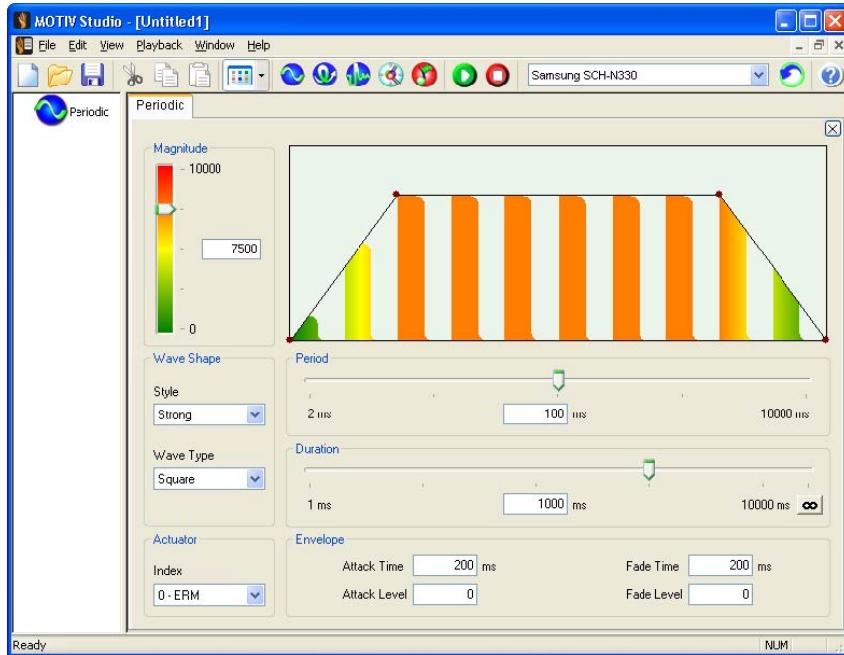


Figure 1.4: MOTIV.

the design of diverse and complex vibrotactile patterns. This is against the current trend of mobile devices that use wide-band vibratory actuators such as Electro-Active Polymer (EAP) actuators [2] or piezoelectric actuators [4].

In this study, we extend the approach of iOS5 to support complex vibration waveform design while preserving original simplicity and intuitiveness. Our *demonstration-based* authoring method integrates various touch input properties such as the position and pressure of a touch and enables to control the frequency and amplitude of vibration as well as its duration. As a result, users can create their own vibration patterns by using simple and fun rhythmic actions like a sequence of tapping and dragging. In addition, we carried out a user experiment to compare performance and user preference between our demonstration-based authoring and the traditional waveform-based authoring. Details are presented in the rest of this paper.



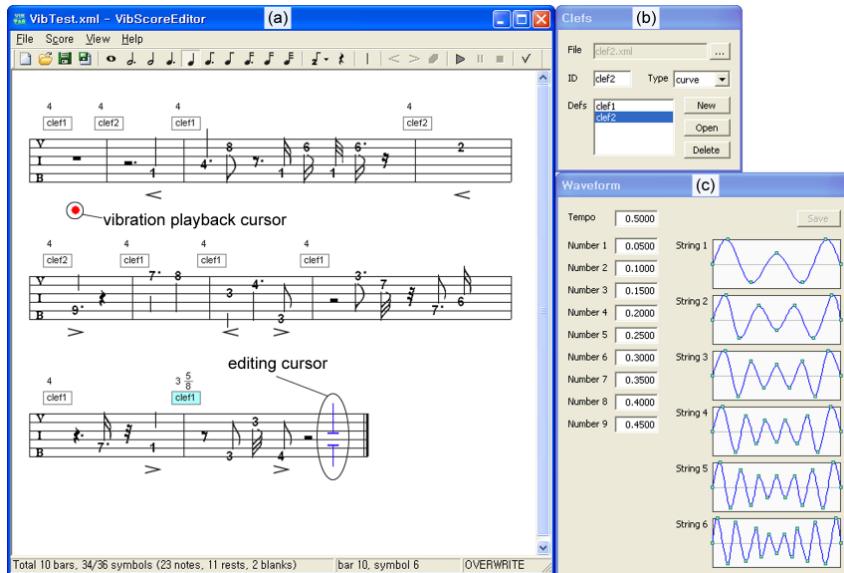


Figure 1.5: VibScore.



Figure 1.6: Custom vibration making in iOS5.



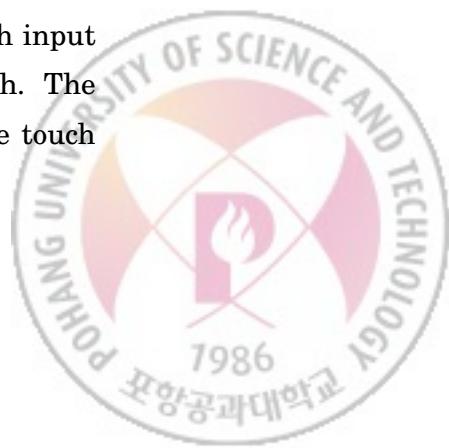
Chapter 2

Vibrotactile Pattern Authoring Methods for a Touch Screen

This chapter describes two vibrotactile pattern-authoring methods that use a touch screen. First, section 2.1 discusses the concept of the demonstration-based authoring method. In order to evaluate the demonstration-based authoring method, it should be possible to use the legacy authoring methods mentioned in Chapter 1 on a touchscreen. However, this has not been realized thus far. We have developed an editor by adopting waveform-based authoring. Section 2.2 discusses the concept of waveform-based authoring on a touch screen.

2.1 Demonstration-Based Authoring Method

In a demonstration-based authoring method, a vibrotactile pattern can be generated on the basis of a user's motions; therefore, a user can make a freeform pattern with only simple gestures. For this purpose, the duration, frequency, and amplitude of the waveform can be controlled by the input of the user on the touch screen. These vibrational properties are mapped to the touch input attributes, which seems to be the most intuitive and efficient approach. The duration of the waveform depends on the duration of contact with the touch

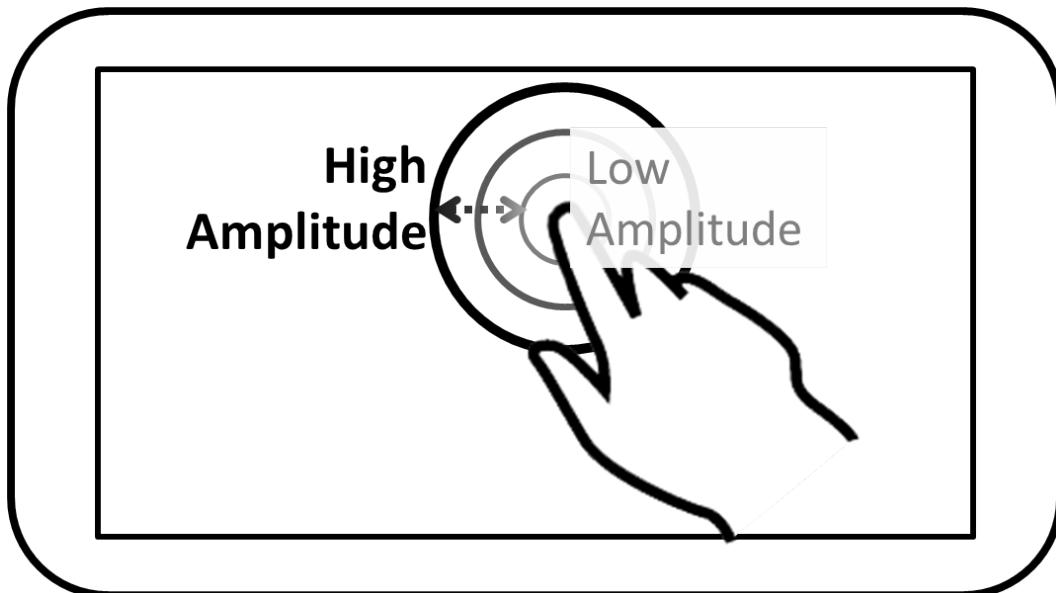


screen, as in the case of Apple devices. This is very intuitive because the pattern duration is determined by the user. The frequency is changed according to the y-coordinate of the touch position. The frequency level is expressed as high or low and the height is similarly expressed in English (as well as some other languages); hence, this method can easily combine the two notions. Therefore, it is suited to intuitive use. In addition, this method corresponds to a musical metaphor. The amplitude is controlled by touch pressure; therefore, when the user pushes the screen harder, the amplitude increases. Figure 2.1 is a conceptual diagram of each control method.

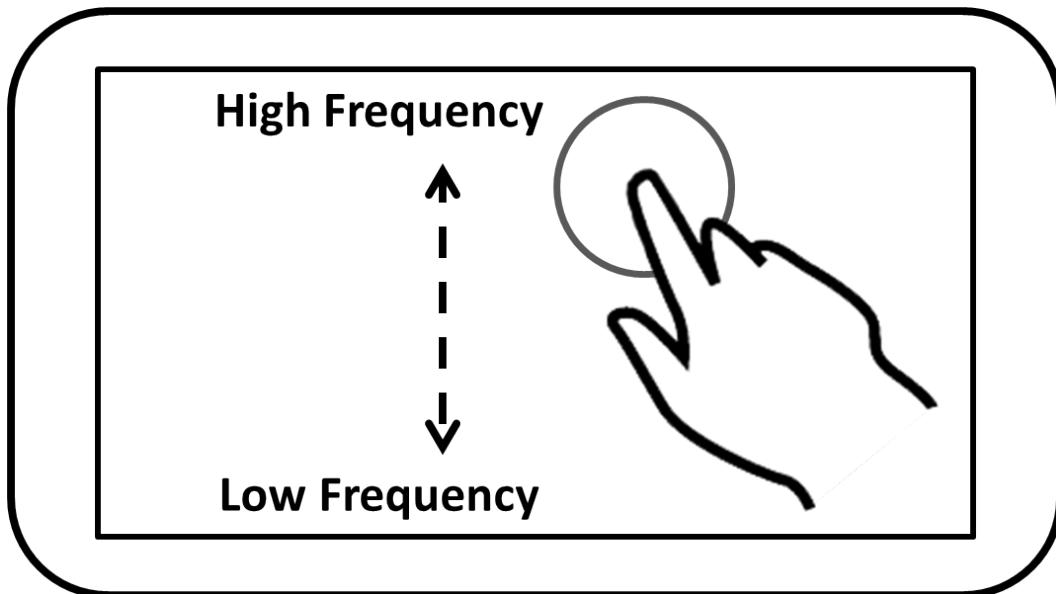
2.2 Waveform-Based Authoring Method

A waveform-based method must adopt a simpler user interface, because the touch screen is not as precise as the mouse or keyboard. Therefore, a continuous vibrotactile pattern is represented by a graph that consists of simple triangles and quadrangles in addition to lines. As shown in Figure 2.2, a continuous sinusoidal waveform is simplified to a triangle or quadrangle and is called a block, while a blank section that is between two blocks is shown by a line and is called a gap. A block can deal with continual modulation (e.g., changes of amplitude or frequency); however, the simplest possible form must be maintained. When a complicated waveform is needed, it should be possible to create one with composite blocks. The duration, frequency, and amplitude correspond to the width of a block or gap, the brightness of a block, and the height of a block, respectively. Figure 2.3 shows how to control each property of a block. To exploit the intuitiveness of the touch screen, the width and height can be directly changed by finger movements, and the brightness can be regulated with a seeker.





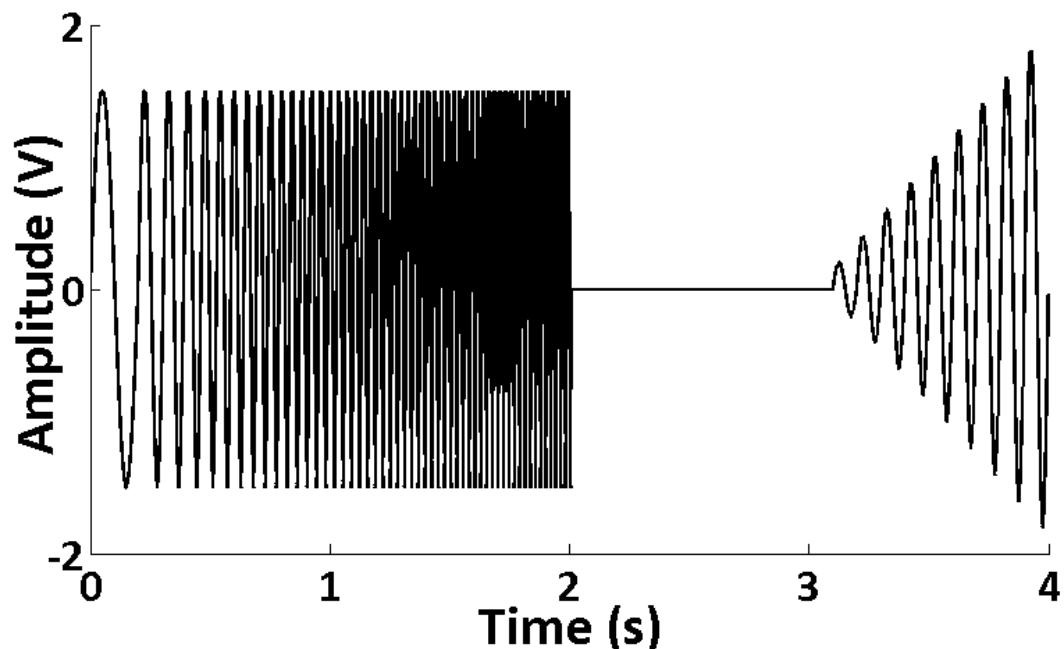
(a) Amplitude control.



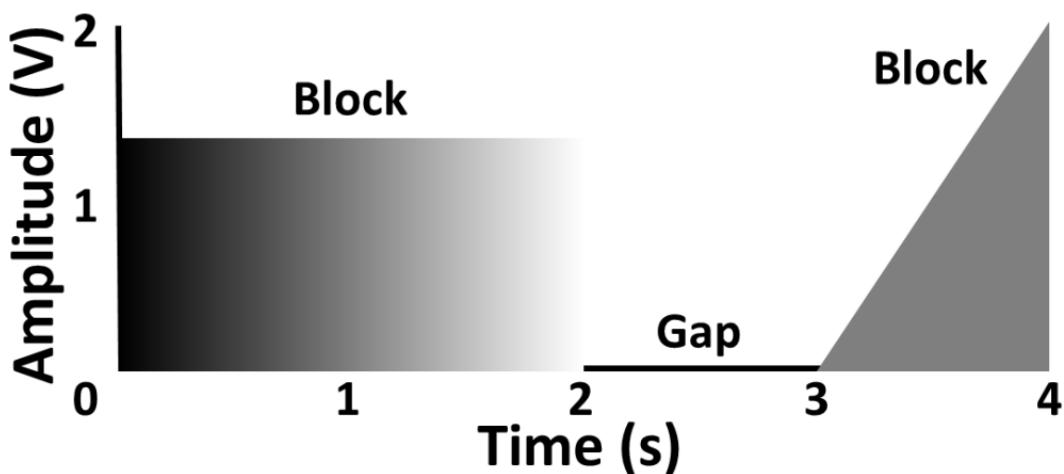
(b) Frequency control.

Figure 2.1: Concept of demonstration-based authoring method.





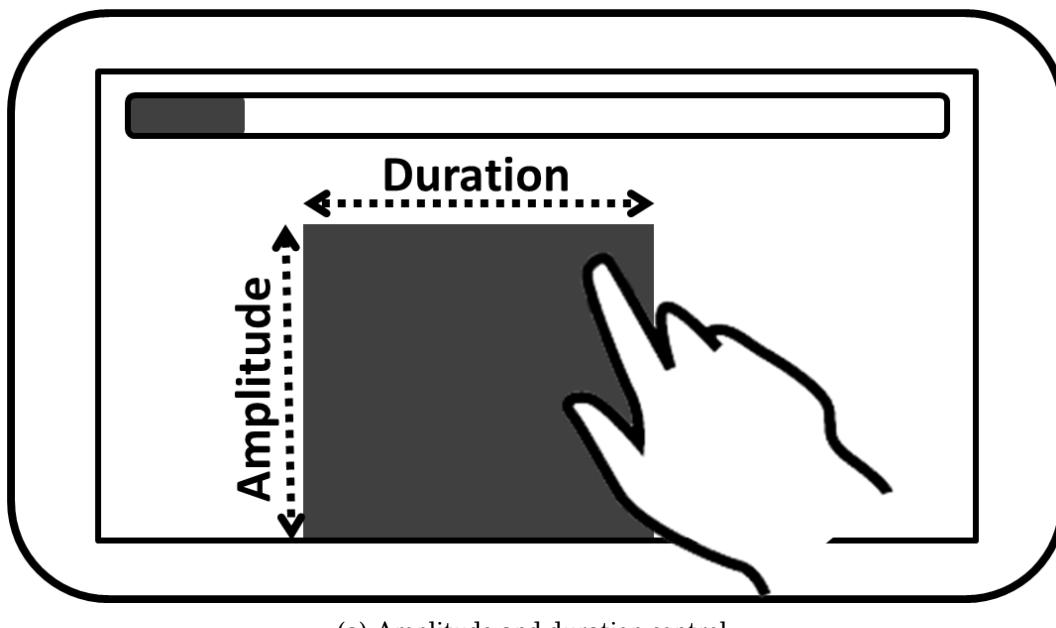
(a) Original waveform.



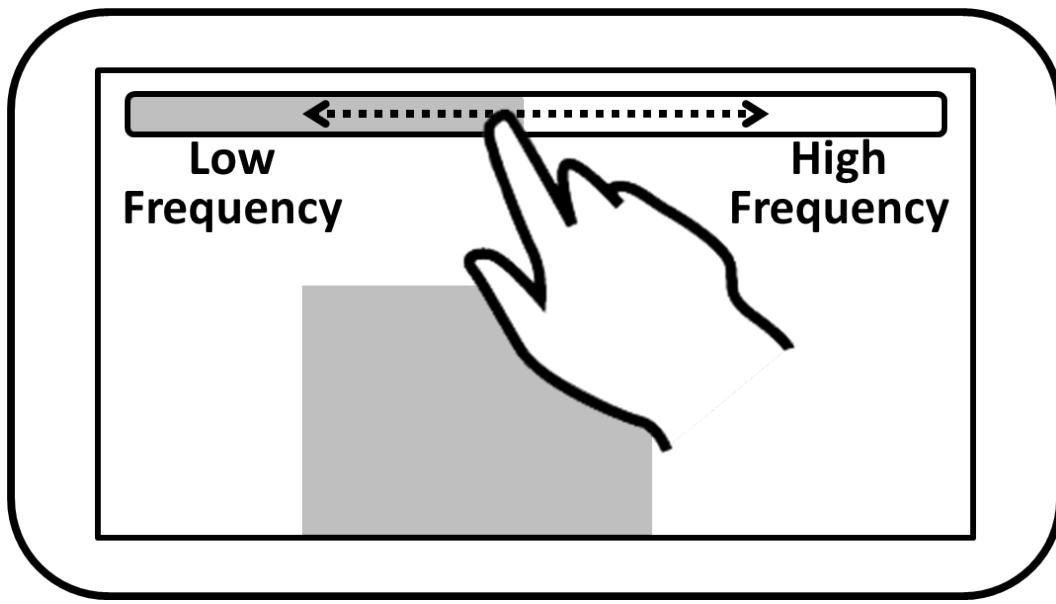
(b) Simplified waveform.

Figure 2.2: Concept of waveform simplification.





(a) Amplitude and duration control.



(b) Frequency control.

Figure 2.3: Concept of waveform-based authoring method.



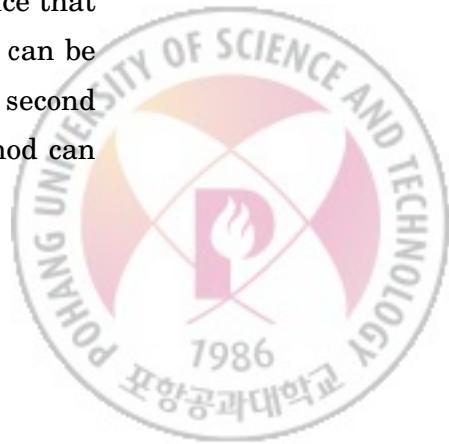
Chapter 3

Implementation

In this chapter, the results of the implementation of the two methods proposed in Chapter 3.2, and the issues that occurred in the process, are described.

3.1 Demonstration-Based Authoring Method

The main issue of the implementing concept in section 2.1 is the control of the amplitude using touch pressure, because there is no mobile device in the current market that provides the capability of measuring the touch pressure. Thus, there is a need to find an alternative touch input property for amplitude control. One of the solutions for this is to use the size of the touch area provided by the capacitive touch screen adopted by most of the present smartphones or tablets. It utilizes a trend in which when the touch pressure is increased, the size of the touch area seems to also increase. This size can be used as an approximate pressure.¹ The main advantage of this method is that it can maintain intuitiveness because it partially achieves the intended goal by using touch pressure for amplitude control. Besides, when a new mobile device that provides touch pressure measurement is released, the proposed editor can be easily applied to such device. The size resolution, however, is low. The second method is to utilize the x-coordinate for amplitude control. This method can

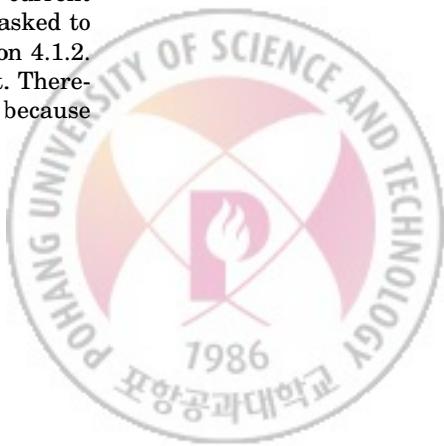


easily and minutely manipulate the amplitude value, and most importantly, it can be applied to all types of touch screen. Its drawback is it is less intuitive than the first one and thus does not correspond to the original concept of the demonstration-based method. The first method, which uses the size of the touch area, was thus selected after contemplating the aforementioned facts.

There is the problem, though, of the produced pattern's amplitudes being discrete due to the low size resolution, as mentioned earlier. To address this problem, a moving average method was used after testing several filters. At that time, the delay becomes longer when the window size is bigger, so the window size was adjusted to 5. Measuring the current size of the touch area at regular intervals is also a condition for using the moving average. Android UI, however, runs by event, and its size renewal interval is erratic. Therefore, the size is saved as a global variable when the value is changed, then the logging thread puts the size on the moving average every 10 ms. The result of the moving average is the amplitude of this time. It was thought that logging on 100 Hz is enough to store every change in size because the human movement is not very fast.

There exist some user differences in the touch width level, and it requires calibration by setting a maximum size for the touch area. Therefore, a function that can measure the maximum size of the touch area and that can fit the maximum size to the maximum amplitude was added. When logging the size of the touch area, the measured size keeps changing because the contact area on the touch screen is continuously altered. The front and rear parts of the logged data series from the first contact to the maximum size and from the maximum

¹Before the first method was considered a candidate, a simple pilot test was conducted to verify if the size of the touch area can substitute for the touch pressure. Six co-workers, all without prior knowledge of the experimental procedure or its purpose, were informed of the current measurement of the pressure of the strong, middle, and weak touch. Then they were asked to randomly touch the touch screen of the Galaxy Tab 10.1, the device described in section 4.1.2. The differences between the sizes of these three touch areas were statistically significant. Therefore, it was decided that the size of the touch area can substitute for the touch pressure because the size seems to be distinguished along with the pressure.



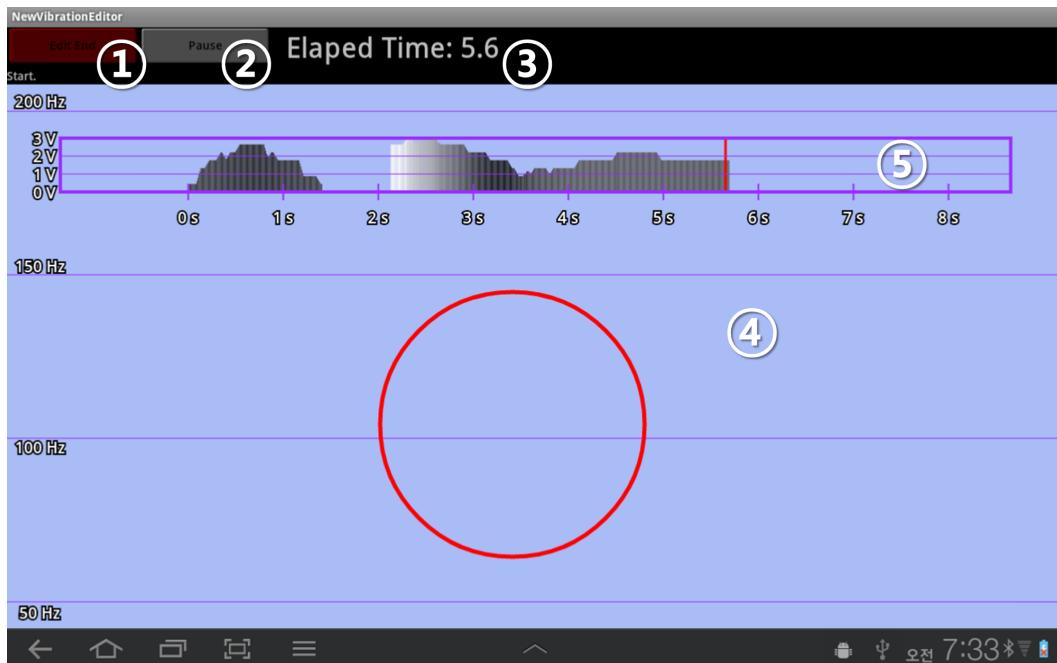
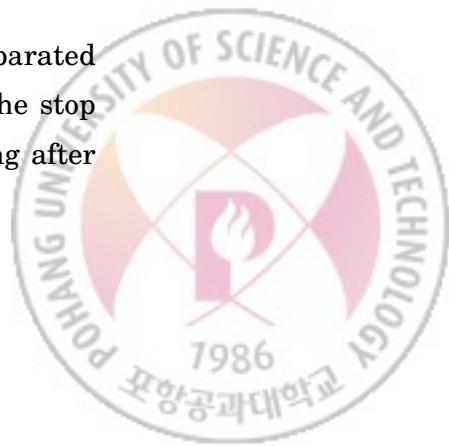


Figure 3.1: Demonstration-based editor user interface.

size to perfect removal from the touch screen must be cut off because they are useless. The algorithm for searching the section that reached the maximum size and that was kept stable is slightly complicated, however, so the first and last 25% parts were just cut off from the logged data series. The average of the remaining part was set as the result of one time, and the mean of three times was selected for the maximum size of the user.

Based on the above concept, an editor was made on the Android platform for the initial study of the demonstration-based method. In addition, visual feedback was considered to inform the user of the currently produced vibrotactile pattern. Below are each part's name and role.

- ① Edit start/stop button: This button starts or ends the editing, separated by the caption and the colors green and red. When pushed at the stop state, the editor will wait for the user's motion for pattern editing after



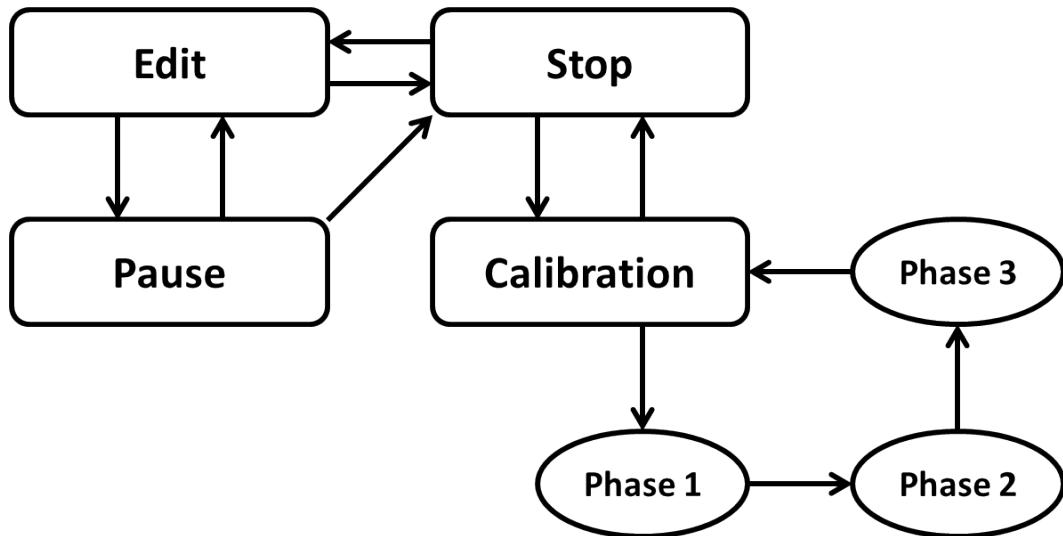
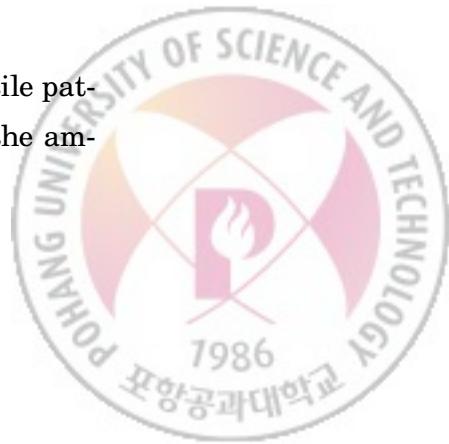


Figure 3.2: Demonstration-based editor state machine.

initialization. On the contrary, in the edit state, this button is used to exit editing and to transmit the recorded data.

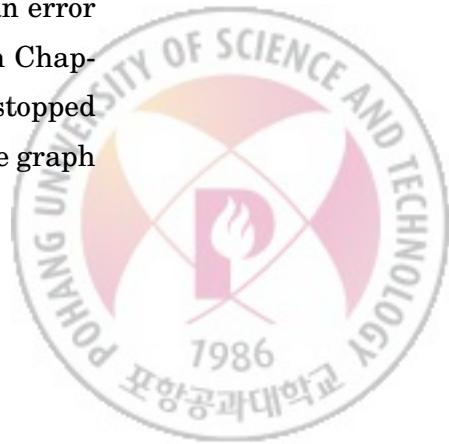
- ② Multifunction button: When pushed at the stop state, this button starts the calibration. It also pauses the editing in the edit state and resumes it in the pause state.
- ③ Elapsed time: It denotes the elapsed time after the start of editing, and represents the length of the produced vibrotactile pattern.
- ④ Workspace: This area takes the user's motion information and draws the visual feedback; therefore, any touch on the external part of this area is not reflected in the pattern making. If the editor waits for the start, as mentioned in ①, vibrotactile pattern production will start from the moment the user touches this area.
- ⑤ Graph: This shows the approximate sketch of the output vibrotactile pattern. The height, width, and brightness of the graph represent the am-



plitude, duration, and frequency, respectively. Dark means low frequency, and bright high frequency. The graph box is 10 seconds long, and the left side is the former time. The pattern at the time is drawn from the red line (editing line), which is positioned at the 7/10 point of the graph box during the editing. The user can also read the approaching time. As the graph remains after the editing is closed, the user can examine the produced pattern.

The procedure of vibrotactile pattern making is as follows. First, as Apple's duration control method was borrowed, the touch screen contact time is set based on the duration itself. As the user touches the screen, as a visual feedback, a circle appears around the finger, and a graph of the current pattern is simultaneously drawn. The screen height is assigned to the frequency; therefore, if the user touches the lower part of the screen, the current frequency is set at a low level, and if the user touches the upper part, the current frequency is set at a high level. Horizontal lines are allocated every 50 Hz so that the user can roughly realize the current height frequency. There is some space above the maximum line and below the minimum line, and the inputs of this area are the fixed maximum and minimum frequencies, respectively. This is the safety zone because the chance of touch miss becomes higher near the end of the workspace. The frequency change is also represented by the frequency of the circle's border thickness change and by the brightness of the current graph. The amplitude is set proportional to the size of the touch area and is denoted by the circle size and graph height, which are also proportional to the size of the touch area.

Lastly, a pause function was supplemented to improve the demonstration-based editor, based on the results of the experiment in Chapter 4. The participants were told that requiring an editor remake from the beginning if an error occurs is too inconvenient. A detailed discussion of this can be found in Chapter 5. When the pause button is pressed while editing, the process is stopped and a graph shows the produced vibrotactile pattern until that time. The graph



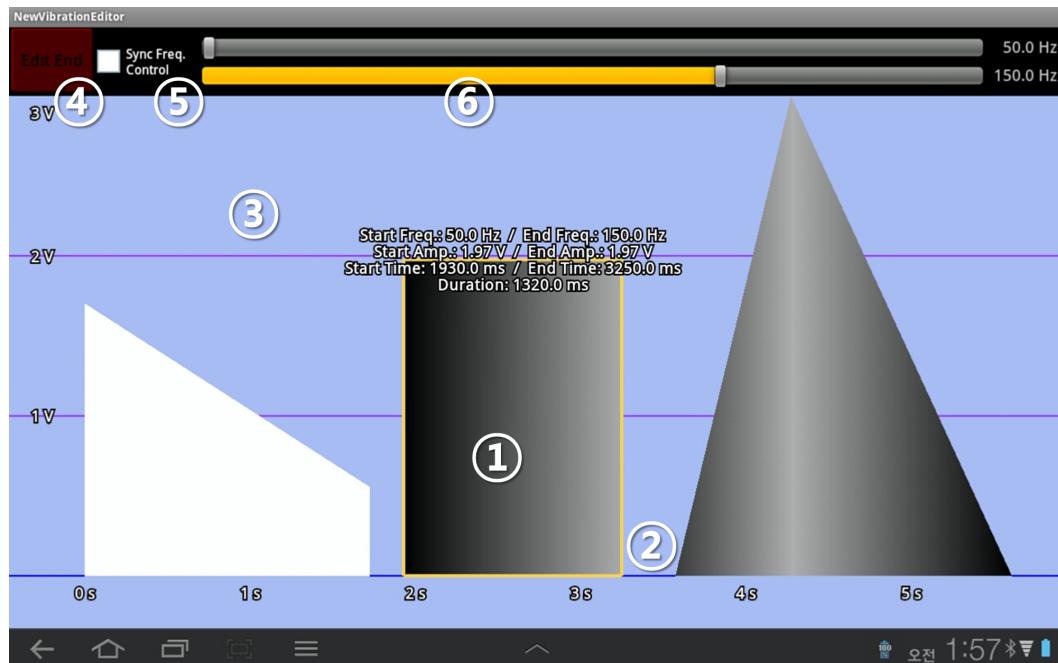


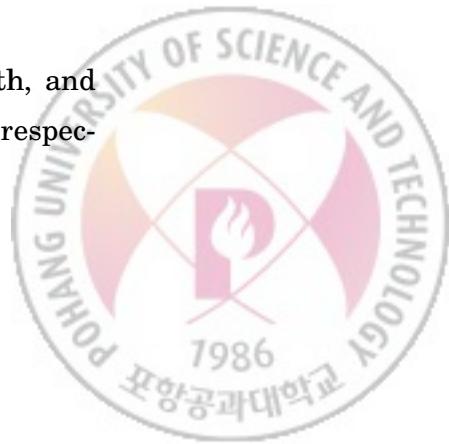
Figure 3.3: Waveform-based editor user interface.

can be moved by scrolling the screen. If the user puts the point that he or she wants to remake on the editing line and pushes the continue button, the pattern below the point is removed, and the editor waits for the user input again. By utilizing this function, the user can also check beyond the 10-second result pattern.

3.2 Waveform-Based Authoring Method

The waveform-based editor was also developed on the Android platform, based on section 2.2. The user interface consists of a form that seems to be capable of fully using the advantages of the touch input in the following manner:

- ① Block: It represents a simple waveform so that the height, width, and brightness correspond to the amplitude, duration, and frequency, respec-



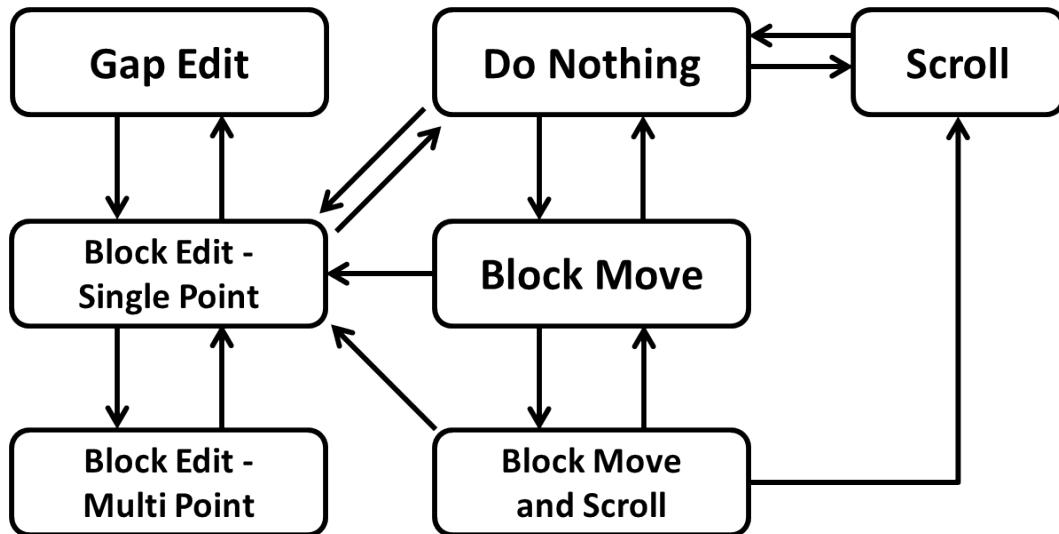
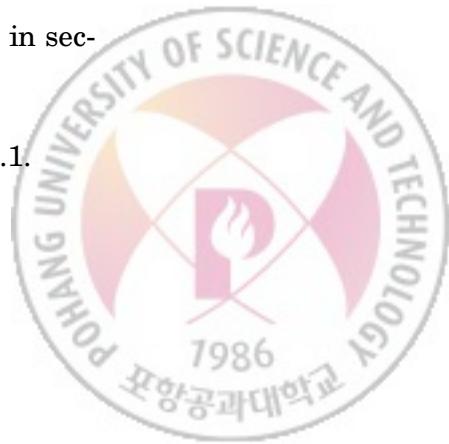


Figure 3.4: Waveform-based editor state machine.

tively; also, each factor can be controlled independently. Especially, the height and brightness have separate left and right end values, and the values between these were linearly changed. Therefore, a slope can appear on the upper side of the block, and if one side's height is zero, then the block will be a triangle. Due to the brightness difference, the block can also have a gradation effect. When the user manipulates the properties of a block, the current frequency, amplitude, and duration are indicated on the side of the block. The minimum frequency control unit is 1 Hz, the duration is 5 ms, and the amplitude is changed per 0.01 V.

- ② Gap: It represents a blank section between two blocks. Its width is mapped onto the duration and can be manipulated every 5 ms. The user can make the width zero to meet the two blocks.
- ③ Workspace: It takes the user input and draws visual feedback, as in section 3.1.
- ④ Edit start/stop button: It starts or ends the editing, as in section 3.1.



- ⑤ Frequency control synchronization: This option decides if the brightness values of the two ends in a block will change together. If it is set, the two brightness values can be controlled simultaneously; if not, they will have to be adjusted independently. When it is switched from off to on, the right end value will be set to the left end value.
- ⑥ Frequency control bar: This seeker can adjust two brightness (i.e., frequency) values of a block. The upper seeker controls the left side and the lower seeker changes the right side. The lower seeker will be disabled when ⑤ is unselected.

The block and gap can be controlled as described in the following. The user first selects any block or gap through one touch tap. The selected block is distinguished by the yellow bound, and its frequency can be adjusted using ⑤ and ⑥. The selected gap is represented by the yellow line, and if it is tapped again, a new block and a gap will be added to the front of the gap. These elements' data will float on the side. To manipulate any block's width and height (i.e., the duration and amplitude of the waveform), just touch the block and drag it; it does not matter if the block is selected or not. Dragging with a single touch can control the duration by x-axis movement, and the amplitude by y-axis movement. Therefore, the diagonal direction can be changed into two values simultaneously. When two fingers touch one block, the boundary of the block becomes red. The left-finger motion along the y-axis controls the left amplitude, and the right finger works accordingly. The duration is altered by changing the x-axis length between the two contacted fingers. If the width or two heights become zero, that block will be removed, and the two neighboring gaps will be united into one. The width of a gap can be manipulated using two neighbor blocks, by touching each of the two blocks and changing the distance between the two fingers.



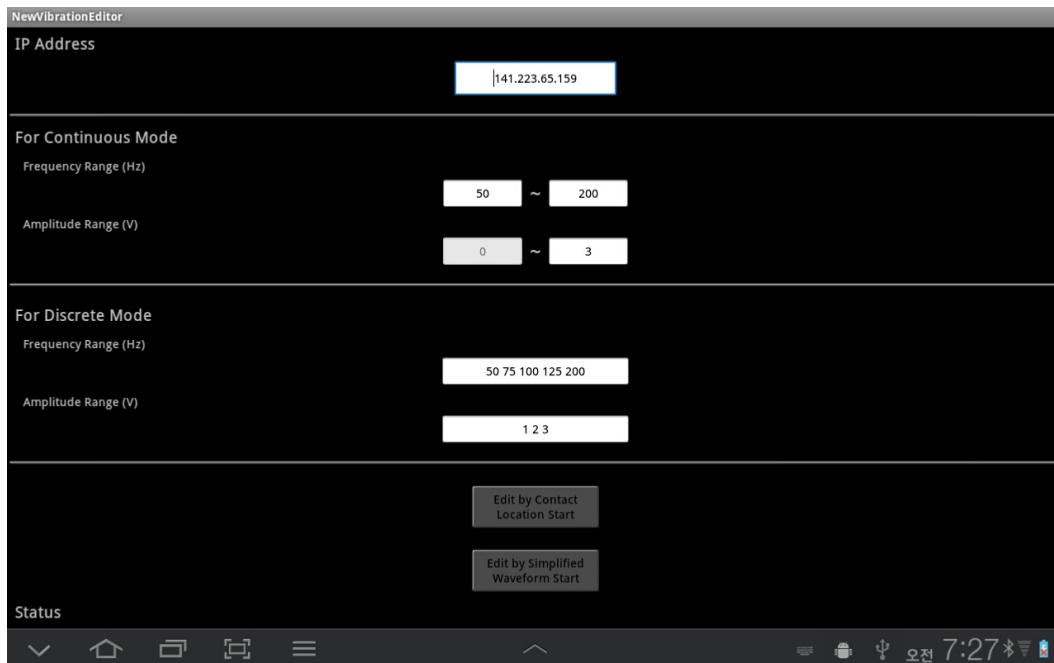


Figure 3.5: Desktop program main screen.

The rest functions are as follows. The scroll function moves the currently visible screen to the side by touching the workspace where no block or gap exists, and just dragging towards the goal. If a block is touched for 0.5 seconds, the block moving function will start with a short vibration. The selected block will float on the screen, following the user's finger; therefore, the user simply drags it towards the goal, and when he or she removes his or her finger from the screen, the block moves into that position. If the goal is not visible in the current screen, the user can shift to another screen by dragging the contacted finger to the side, touching the blank workspace using another finger, and dragging it in the same way as with the scroll function.



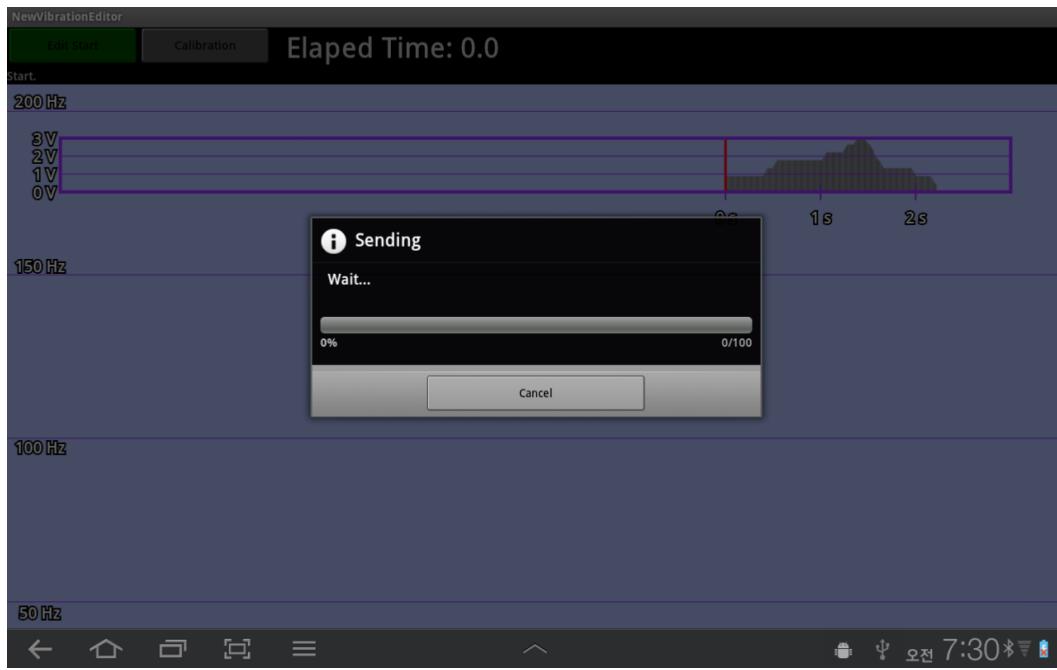


Figure 3.6: Desktop program sending progress.

3.3 Common Things of the Two Editors

At the main screen, some configurations can be adjusted, like the IP address of the connecting desktop and the minimum and maximum values of the amplitude and frequency, also chosen to run the two editors. When the edit stop button is pushed, each editor tries to connect the desktop through the Wi-Fi, and to transmit the current data, because Android 3.0 (i.e., Honeycomb) or the above is strict about “application not responding (ANR)” and prohibits communication on the main thread, therefore creating a new thread for transmission. In addition, it is impossible to send all the logged data at one time, so the user must transfer 30 items at one time and must wait for acknowledgement (ACK). During the transmission, the user can be informed about the transfer state through the progress window.



3.4 Desktop Program

This program receives data from the editor and creates a real vibration waveform based on such data, and then prints out the produced waveform through the output board. It has two classes because the data from each editor is different. These classes also separately have member methods about saving each data and making the waveform, but their names are the same. When the program is started, it waits for TCP connection and accepts application from the editors, and then makes a variable of the correspondent class depending on the first received packet, and sends ACK to the editor. The program receives and records the contents of the packets and also sends ACK to the editor until the last packet is received. After the transmission, the program builds the vibration waveform and plays it through the output board.



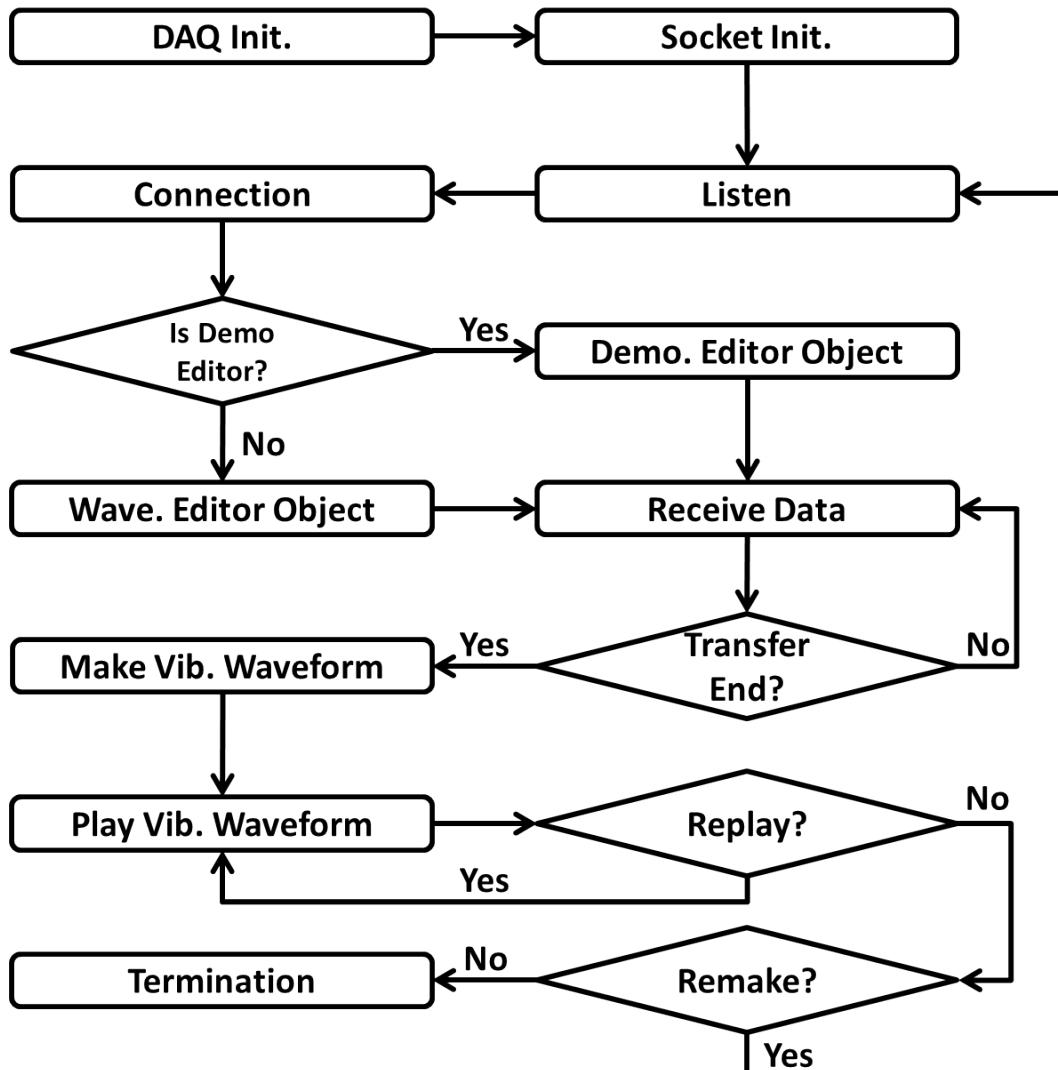


Figure 3.7: Desktop program flowchart.



Chapter 4

Usability Evaluation

In this chapter, an experiment aimed at the common users was conducted to find the pros and cons of the demonstration-based method and of the traditional waveform-based method, and to determine which method will be preferred by the users in particular situations.

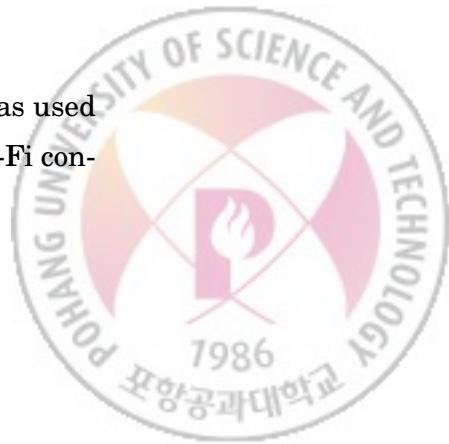
4.1 Methods

4.1.1 Participants

The ten participants were 20-25 years old (average: 21.6), male, and students of the author's university. They answered that they used a mobile device with a capacitive touch screen more than once a year, and most of them indicated that they can handle the device skillfully. They also answered that they do not have any sensorimotor abnormality on their hands and arms, and that they do not have any experience of making vibrotactile patterns by themselves. All the participants were paid for their participation in the study.

4.1.2 Apparatus

A Samsung Galaxy Tab 10.1 (SHW-M380W; Android Honeycomb 3.2) was used for running the application and for sending the data to a desktop via Wi-Fi con-



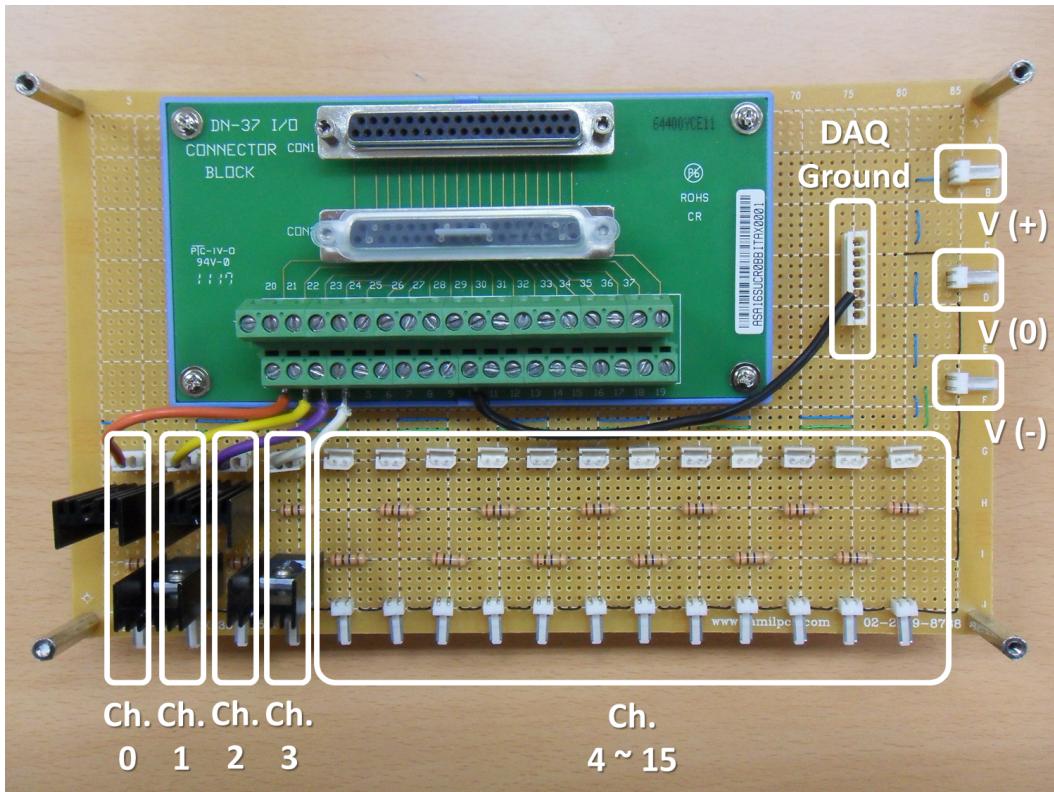


Figure 4.1: Self-configured current amplifier.

nection. To output the vibration waveform, an ICPDAS PISO-DA16U output board was used. A current amplifier was also configured using BURR-BROWN OPA544, which can extend to 16 channels, and it was combined with the terminal of the output board. This amplifier uses a power supply that can support two AC 12V channels (Figure 4.1). The actuator was Tactile Labs Haptuator (TL002-14-A) and was attached to an Acrylic Mockup. Figure 4.2 shows the whole configuration of the apparatus.

4.1.3 Experimental Conditions

The experiment consisted of a training task and two main tasks. The training had five subtasks intended to make the participants at ease with each editor,



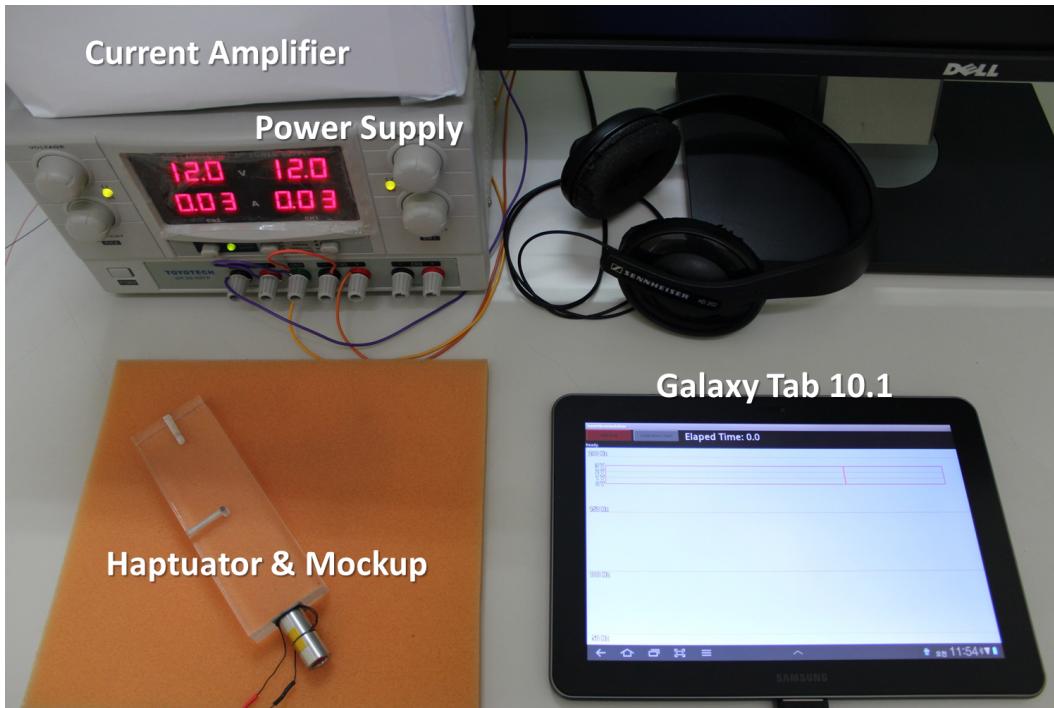


Figure 4.2: Total apparatus configuration.

and presented a description of the simple vibrotactile pattern that the participants needed to make in each subtask (Figure 4.3).

Task 1 was designed for measuring the time that elapsed until each participant making a vibrotactile pattern matching to given description of a tactile icon which contains a message. Another purpose of the experiment was determining the subjective usability of each method. Task 1 consists of two subtasks, each of which consisted of implementing a compound tacton by combining three elements randomly chosen from three groups: the action, object, and result. No element was chosen twice by the same participant. A one-second blank was inserted between the groups; therefore, the compound tacton was an 8.5-second-long pattern. The set of elements was taken from [6], and it was adjusted to suit the frequency-changing tasks (Figure 4.4).



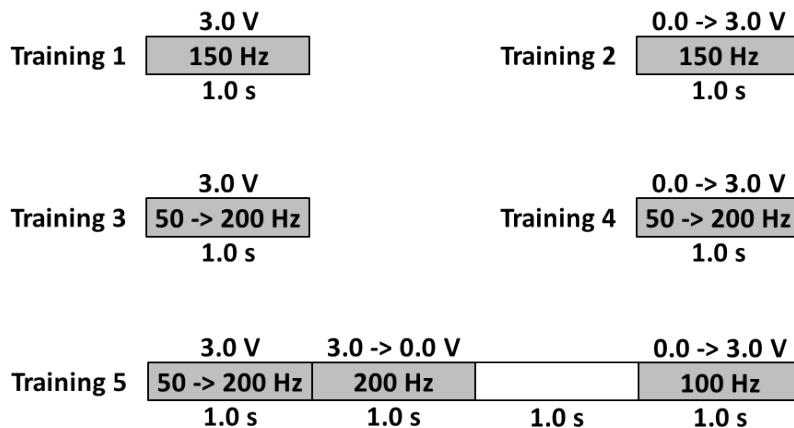


Figure 4.3: Instructions for training.

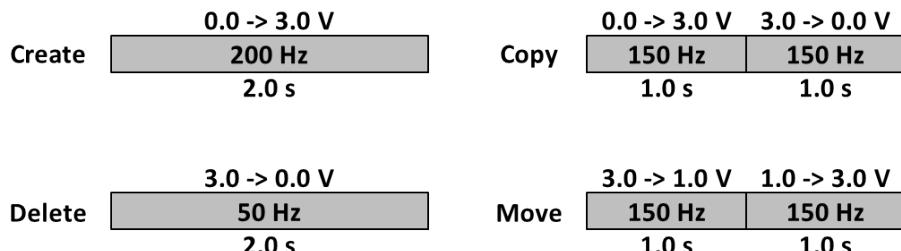
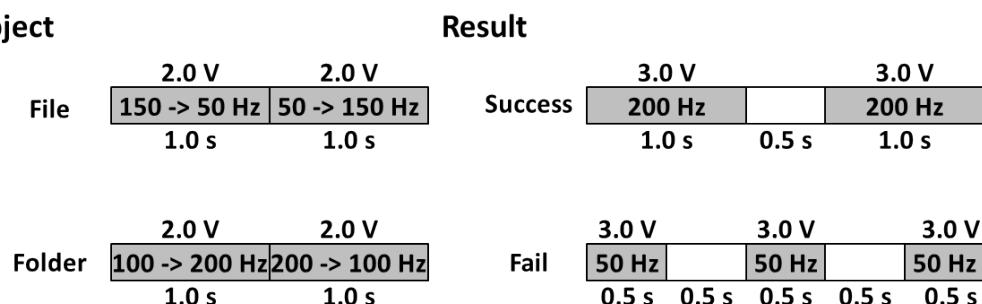
Action**Object**

Figure 4.4: Instructions for task 1.



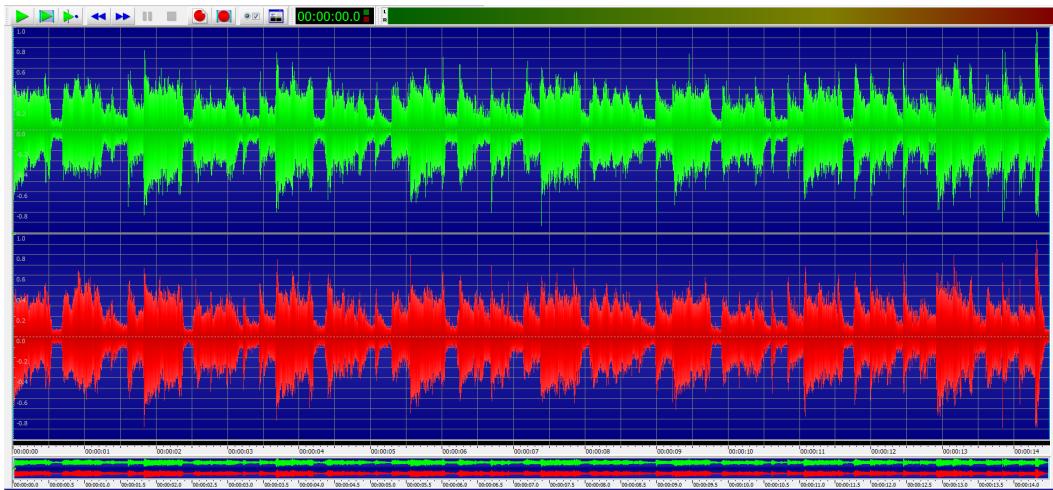
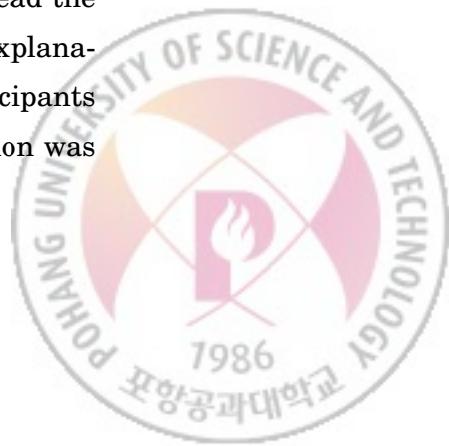


Figure 4.5: Waveform of the music in task 2.

Task 2 was aimed at checking how much time it takes a participant to consider a vibrotactile pattern suitable for a currently heard song, and to produce it extemporaneously. Similar to task 1, the subjective usability of each method was also looked into. Therefore, in task 2, the participants were asked to make a proper vibrotactile pattern for the given part of a song. The song was the 2'30"-2'45" part of “Dragostea din tei” sung by O-Zone, a group of Moldovan musicians. For the participants’ convenience, they were allowed to repeatedly play the section that they wanted to play, and to see the real waveform and time of the song on the monitor (Figure 4.5).

4.1.4 Procedure

The experiment was performed for two consecutive days for the two editors, per participant. The editor that was used on the first day was selected alternately for balancing. Before each experiment, all the participants read the editor manual of the day, and after that, they were again given an explanation of the editor by the experimenter, with a demonstration. The participants were able to manipulate the editor as they wanted when the explanation was

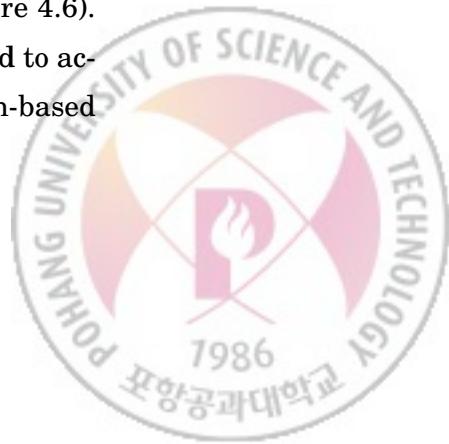


over. When the participants used the demonstration-based editor, they measured the maximum size of the touch area in the usual situation. All the tasks, from the training to task 2, were performed in order, and before each task, the experimenter explained what the participants were to do in such task. All the explanations were based on prepared scripts. During the experiment, each task was repeated until the participant was already satisfied with the produced vibrotactile pattern - that is, until the participant is convinced that the pattern is already in accordance with the problem. If errors were found, the participants assumed that they could find and fix the errors themselves. For each experiment per day, it usually took the participants 40-60 minutes to complete all the tasks.

Upon the completion of the tasks per day, the participants filled out some questionnaires to subjectively evaluate each method and not to assess the editor (interim survey). After completing the whole experiment, all the participants were asked to participate in a survey to assess their overall preferences between the two editors, and conducted self-evaluations of how well match the vibrotactile patterns resulting from the tasks to the problem (final survey). The vibrotactile patterns were randomly proposed by the tasks. All the questions were rated on a continuous scale by checking the position of a horizontal line.

4.2 Results

The results were analyzed through pairwise T-test. Figure 4.6 to 4.9 show the results, and Table 4.1 and 4.2 summarize the T-test results and verbal comments. The task completion time of the training, task 1-1 and task 1-2 (the first and second subtask of task 1) was not significantly different (Figure 4.6). However, the completion time when the waveform-based editor was used to accomplish task 2 was 5.9 times longer than that when the demonstration-based editor was used, and the difference was statistically significant.



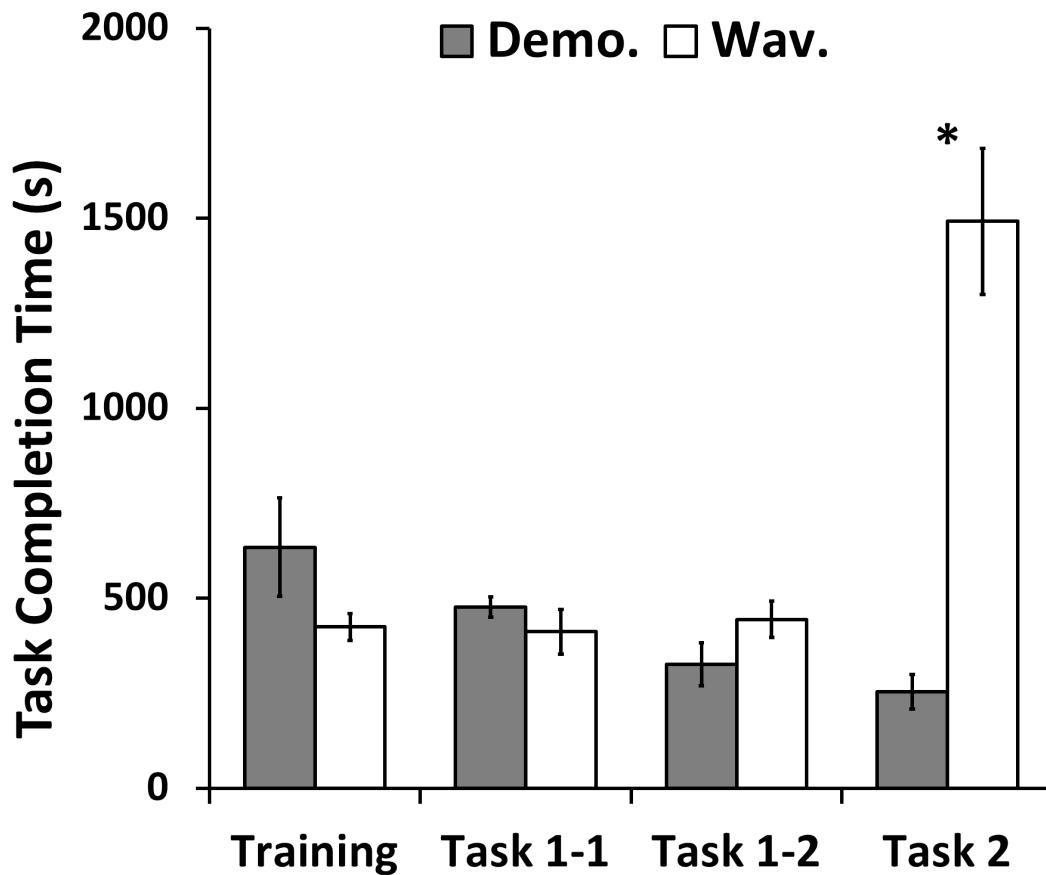
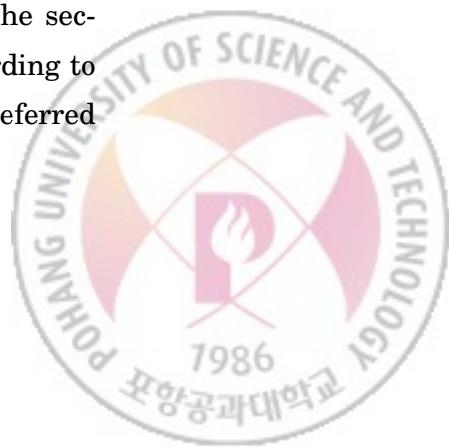


Figure 4.6: Task completion time results. The asterisk indicates that the two design methods are statistically significant ($p < 0.05$). The error bars represent the standard errors.

The preference results were rearranged the four results (intuition, easiness, efficiency and fun) from the interim survey and the overall preference result from the final survey to each task. This results shows a tendency that in the first situation, where a message was passed with vibration, as with task 1, the waveform-based editor mainly obtained higher scores (Figure 4.7). The second situation, however, in which vibrotactile patterns were made according to media (e.g., music or movies), the same as task 2, the participants preferred



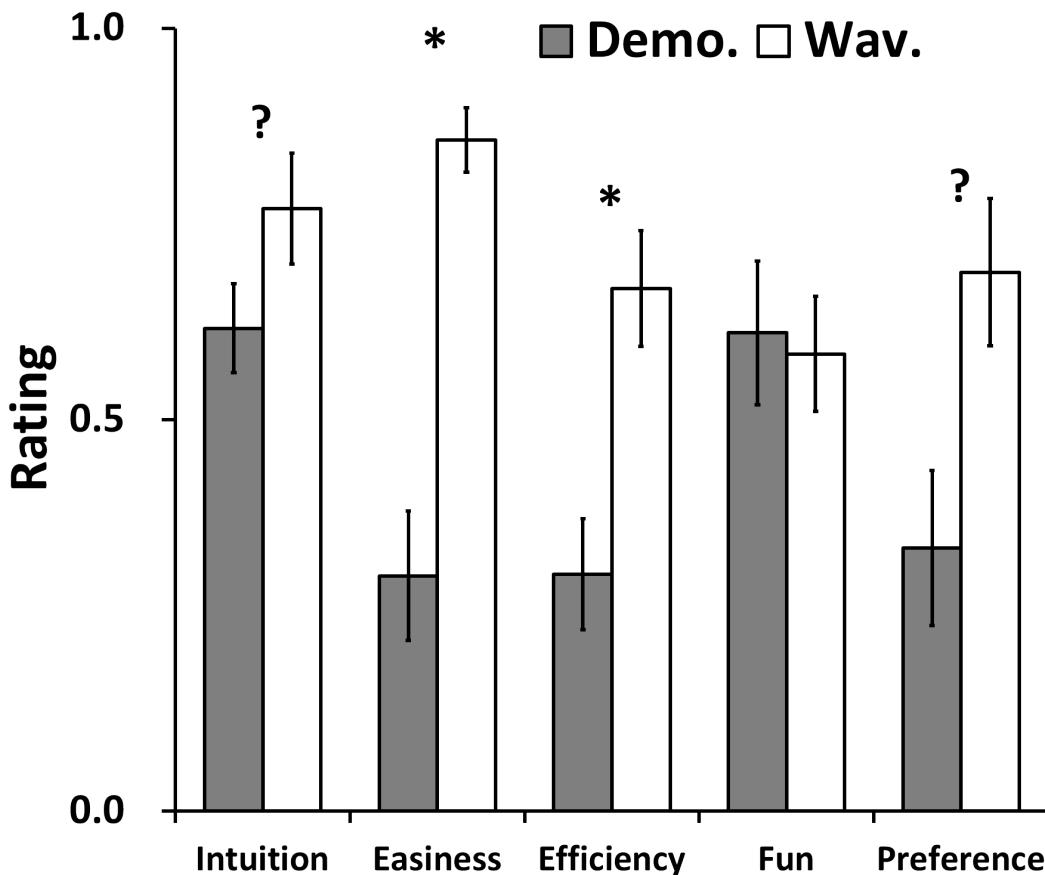
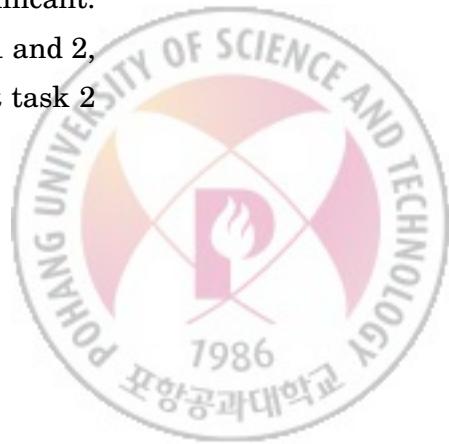


Figure 4.7: Survey results of Task 1. The asterisks indicate that the two design methods are statistically significant ($p < 0.05$). The question marks indicate that the results are marginally significant ($p < 0.1$). The error bars represent the standard errors

the demonstration-based method much more (Figure 4.8). In particular, the demonstration-based editor was generally funnier than the waveform-based method in all the tasks, because the demonstration-based editor was significantly fun and even in task 1, the difference was not statistically significant. In addition, the values corresponding to the ease of carrying out tasks 1 and 2, the efficiency in carrying out task 1 and the preference in carrying out task 2 were statistically significant.



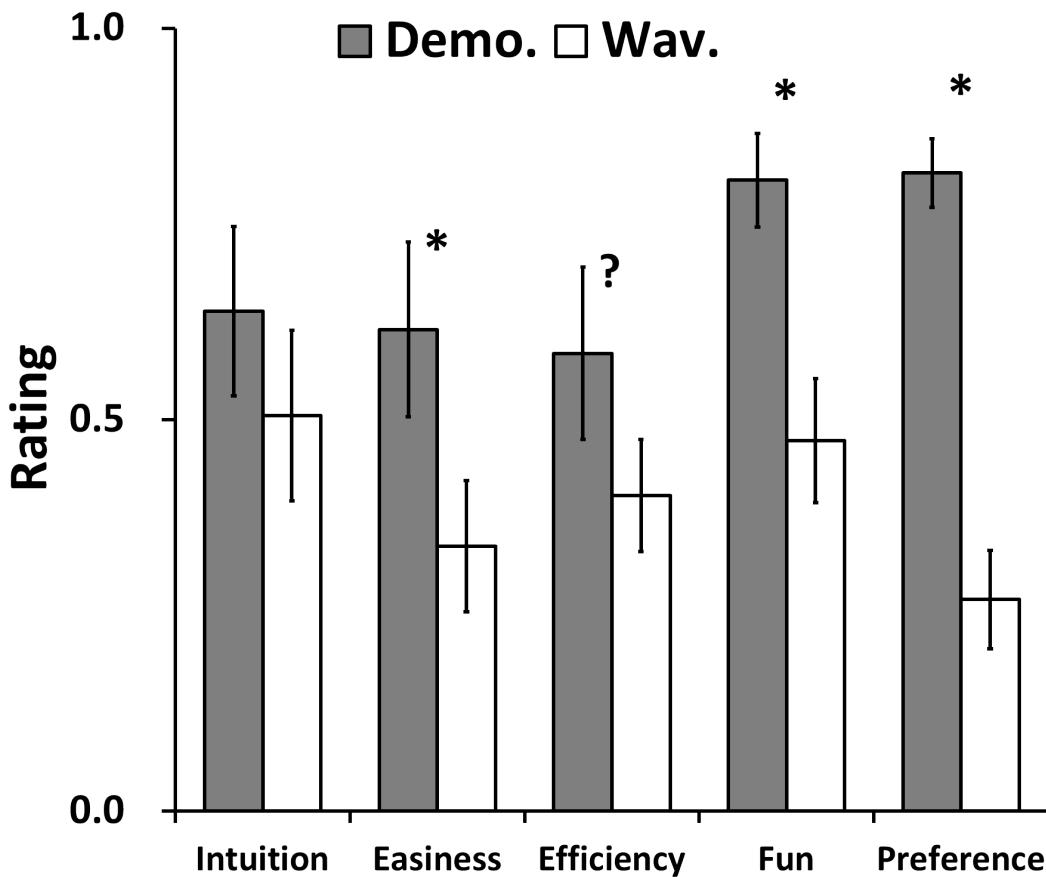


Figure 4.8: Survey results of Task 2. The asterisks indicate that the two design methods are statistically significant ($p < 0.05$). The question mark indicates that the results are marginally significant ($p < 0.1$). The error bars represent the standard errors.

The self-evaluation results of task 1 in the final survey show that the vibrotactile pattern produced by the waveform-based editor obtained higher scores in task 1-1, and that the difference was statistically significant, but the two editors obtained almost the same scores in task 1-2 (Figure 4.9). In task 2, the demonstration-based editor obtained higher scores, and they were statistically significant.



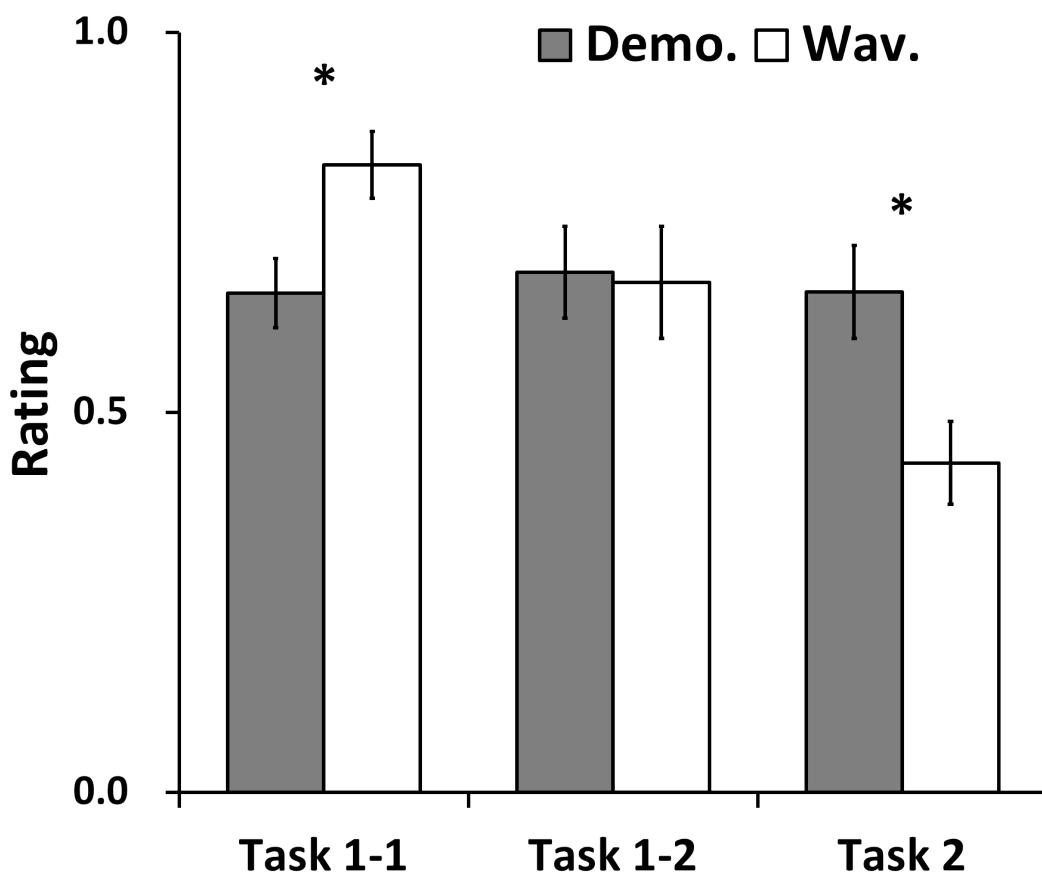


Figure 4.9: Self-evaluation results. The asterisks indicate that the two design methods are statistically significant ($p < 0.05$). The question mark indicates that the results are marginally significant ($p < 0.1$). The error bars represent the standard errors.



Table 4.1 Results of pairwise T-test.

Task completion time		<i>p</i>
Training		0.13019
Task 1-1		0.39668
Task 1-2		0.15057
Task 2		0.00005 *

Interim survey		<i>p</i>
Intuition	Task 1	0.06577 ?
	Task 2	0.24758
Easiness	Task 1	0.00019 *
	Task 2	0.04126 *
Efficiency	Task 1	0.00968 *
	Task 2	0.06752 ?
Fun	Task 1	0.80532
	Task 2	0.00072 *
Learnability		0.02083 *

Preference		<i>p</i>
Self-evaluation	Task 1-1	0.01396 *
	Task 1-2	0.87152
	Task 2	0.02064 *
Preference	Task 1	0.07027 ?
	Task 2	0.00004 *
	Real mobile	0.36789

- Marginally significant cases ($p < 0.1$) are marked by ?.
- Statistically significant cases ($p < 0.05$) are marked by *.



Table 4.2 Summary of verbal comments.**Advantages of the performance-based method**

- Easy to learn and can produce the desired vibrotactile pattern intuitively
- Imperfect but can roughly make the intended pattern; therefore, it will be helpful for making an outline
- Can quickly make a pattern in real time and allows ad lib making with music
- Fun and feels like playing a musical instrument

Disadvantages of the performance-based method

- Hard to control, and has difficulty setting the exact value of the time, the position, and especially the size
- It is difficult to anticipate what the size will be when the device is touched.
- Cannot be easily modified as a high concentration and memory are required

Suggestions for the performance-based method

- The remaking from any part of the already produced pattern seems better.
- Systematic practice in using the method can lead to good performance.
- Changing the time after adjusting the amplitude and frequency or after switching these to the opposite direction provides greater comfort.

Advantages of the waveform-based method

- Short vibrotactile patterns can be quickly, easily, and precisely made.
- Can be learned fast because of its simplicity and intuitiveness

Disadvantages of the waveform-based method

- Realizing the extempore pattern is hard and especially demands bothering with (or considering) too many things when making songs.
- Producing a complex pattern is inconvenient and takes more time when the pattern becomes longer.
- Running on mobile phones seems slightly tough as a larger screen may be needed.

Suggestions for the waveform-based method

- Allowing the typing of an exact value will make the users more comfortable with this method.
- Provide some functions that can copy and paste a block, or that can undo such.
- If an outline is automatically made, based on the sound waveform, when hearing music, a pattern can be easily and quickly made.

Overall appraisal

- More improvement is needed for moving situations because the current editors work well when the device is placed on a desk.
- Making an outline using the demonstration-based method and then revising it using the waveform-based method may be more efficient.
- Use the perceived intensity to represent the amplitude and not the voltage because even if the voltage is the same, the amplitude will be different, based on the frequency.



Chapter 5

Discussion

As for task 2, the task completion time of the demonstration-based method was very short, and the scores for ease, fun, and preference were significantly higher than those of the waveform-based method. Most of the verbal comments were that when vibrotactile patterns are made while hearing songs using the demonstration-based method, the user can use it intuitively and can make the desired vibration patterns freely and extemporary. Perhaps the reason for this is that the waveform-based method needs to match the waveform to the music play time one by one to synchronize the vibration pattern and the music, but the demonstration-based method can easily make the vibration pattern, as if playing simple musical instruments. Therefore, when the made vibration patterns especially link with media like music and movies, the demonstration-based method can be employed in a fast and funny way by the typical users.

What is most difficult and inconvenient when making a vibration pattern using the demonstration-based method is causing a heavy mental load. As the demonstration-based editor works in real time, the participants needed to be adept at all gestures which describing the vibration patterns if they wanted to make the patterns well. Besides, the vibration pattern should be recreated when any error or mistake occurs, which puts a strain on the users. Second, it is



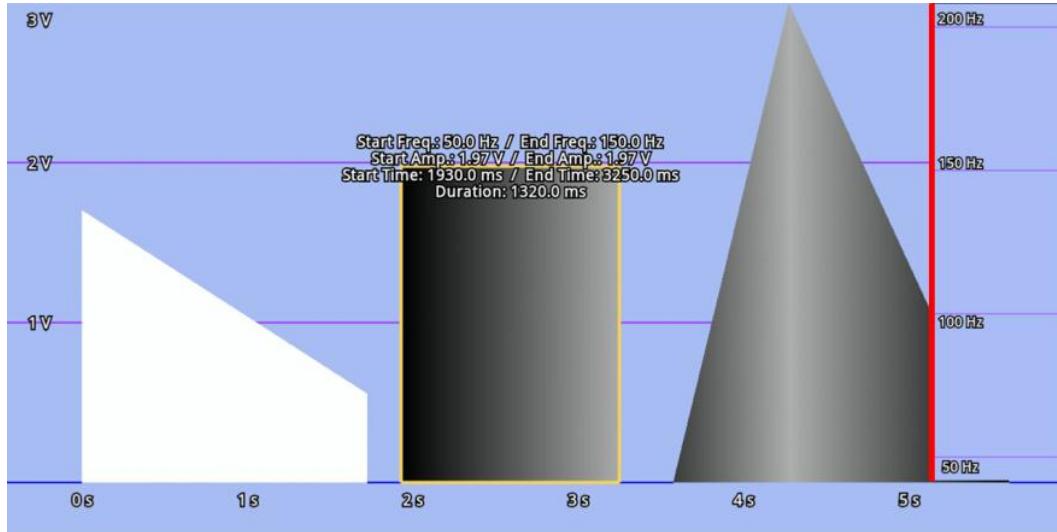
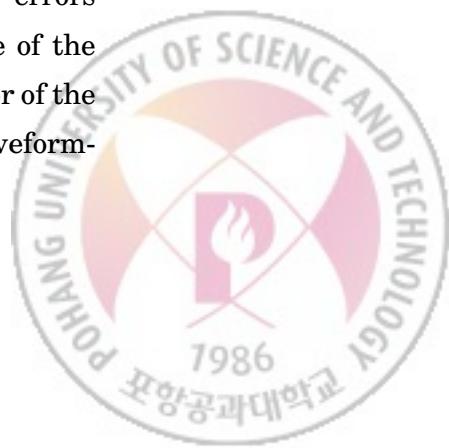


Figure 5.1: Concept of the hybrid method.

hard to control the size of the touch area because the sensing resolution is very low due to the limitations of the mobile devices that are currently being sold in the market. Another reason is that the estimation of the pressure through the size of the touch area is often inaccurate. Some participants also pointed out these matters. The first weakness can be more improved if the users can pause the making of a vibration pattern when they want to, and can remake it from any place; therefore, a pause function was added, as mentioned in section 3.1. Furthermore, if the mobile devices can measure the real pressure precisely, the second weakness can also be improved.

The drawback of the demonstration-based method is that it is difficult to make an exact vibration pattern when using such method. Such imperfections, however, seem not to be a big problem because the common users are not pay attention to details of produced vibration. For example, the participants' errors when using the two editors were compared by calculating the average of the absolute sum of the time differences in each section in task 1-2. The error of the demonstration-based method was roughly 865 ms, and that of the waveform-



based method was about 10 ms, moreover waveform-based method were more preferred at task 1, but the self-evaluation results of task 1-2 were really almost the same. Thus, the current demonstration-based method sufficiently meets the needs of the common user.

Furthermore, if the demonstration- and waveform-based methods will be used together, the vibrotactile authoring will be improved more. For example, an outline of the vibration pattern can be made using the demonstration-based method, and then it can be refined using the waveform-based method. The hybrid method, which combines the two methods, seems better. This method divides the screen into two then puts the demonstration-based method on the right side and the waveform-based method on the opposite side (Figure 5.1). The hybrid method can make blocks and gaps using the demonstration-based method, and can also simultaneously change these using the waveform-based method. These advanced methods can satisfy the expert users, who need more specific editing.



Chapter 6

Conclusion

In this paper, a demonstration-based vibrotactile pattern authoring method that can intuitively make vibration patterns by imitating the user's motion on the touch screen is proposed. To evaluate the usability of this method, two editors were developed on an Android platform: one based on the demonstration-based method and the other representing the legacy waveform-based method. A usability experiment was then conducted using these two editors. Based on the results, the demonstration-based method was highly rated in terms of qualitative and quantitative measures when creating a vibration pattern well matched to a given song, like main task 2. Furthermore, it is concluded that the demonstration-based method is funnier and easier to use by typical users than the waveform-based method, although it also has several disadvantages depending on the type of design task, and it can be utilized in various ways with different media. Lastly, the weakness of the demonstration-based method identified from the experiment was addressed. At present, these authors are developing various additional features to overcome the limitations and disadvantages of the demonstration-based method, and intend to develop an editor that integrates the proposed method with the waveform-based method.



요약문

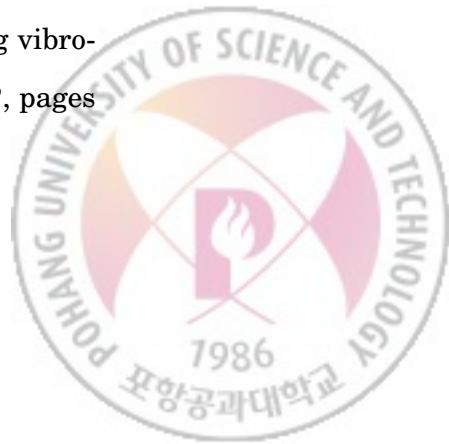
시연 기반 진동 패턴 저작 방법

최근의 다양한 기기들, 특히 휴대용 기기들에서 진동 렌더링의 중요함이 부각되고 있음에도 불구하고, 이러한 진동 패턴의 설계를 용이하게 해주는 저작 도구는 그다지 개발되어 있지 않다. 그 중에서 가장 최근에 소개된 진동 패턴 설계의 개념은 미국의 애플사가 iOS5에서 선보인 것으로 터치 스크린 상에서의 간단한 동작만으로 진동 패턴을 저작하는 것이다. 하지만 이 방법은 현재 진동의 시간 길이만 조작 가능하다. 따라서 우리는 사용자의 터치 움직임을 바탕으로 진동 패턴을 생성할 수 있는 시연 기반 진동 패턴 저작 방식을 제안하였다. 이 시연 기반 저작 방식은 터치 스크린 상에서의 터치 움직임으로 진동의 세 가지 요소인 시간 길이, 진폭 그리고 주파수를 조절할 수 있다. 또한 우리는 시연 기반 방식과 기존의 파형 기반 방식을 비교하여 두 방식의 사용성을 알아보았다. 이 실험을 위하여 우리는 터치 스크린 상에서의 새로운 형태의 파형 기반 방식 진동 패턴 저작 도구를 제작하였다. 실험 결과, 비록 시연 기반 방식이 패턴을 설계하는 상황에 따라 몇 가지 단점을 가지고 있기는 하지만, 파형 기반 방식에 비해 일반인들이 쉽고 재미있게 사용할 수 있을 것이라는 결론을 내렸다. 특히 시연 기반 방식은 음악에 맞는 진동 패턴을 만드는데 특히 좋을 것으로 생각된다.



Bibliography

- [1] Apple Inc. iOS5, features that go further. <http://www.apple.com/ios/features.html>.
- [2] Artificial Muscle Inc. Vivi Touch. <http://www.vivitouch.com/>.
- [3] L. M. Brown and T. Kaaresoja. Feel who's talking: Using tactons for mobile phone alerts. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, pages 604–609, 2006.
- [4] Immersion Corp. What products use haptics? <http://www.immersion.com/haptics-technology/haptics-in-use/>.
- [5] K. Kyung, J. Lee, and M. A. Srinivasan. Precise manipulation of gui on a touch screen with haptic cues. In *Proceedings of the Third Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (World Haptics)*, pages 202–207, 2009.
- [6] J. Lee and S. Choi. Evaluation of vibrotactile pattern design using vibrotactile score. In *Proceedings of the IEEE Haptics Symposium 2012*, pages 231–238, 2012.



- [7] P. Lemmens, F. Crompvoets, D. Brokken, J. van den Eerenbeemd, and G.-J. de Vries. A body-conforming tactile jacket to enrich movie viewing. In *Proceedings of the Third Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (World Haptics)*, pages 7–12, 2009.
- [8] G. Park, S. Choi, K. Hwang, S. Kim, J. Sa, and M. Joung. Tactile effect design and evaluation for virtual buttons on a mobile device touchscreen. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI)*, pages 11–20, 2011.
- [9] J. Ryu and S. Choi. posvibeditor: Graphical authoring tool of vibrotactile patterns. In *Proceedings of the IEEE International Workshop on Haptic Audio Visual Environments and Games (HAVE)*, pages 120–125, 2008.
- [10] C. Swindells, E. Maksakov, K. E. MacLean, and V. Chung. The role of prototyping tools for haptic behavior design. In *Proceedings of the Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 161–168, 2006.



Curriculum Vitae

Name : Kyungpyo Hong

Education

- 2003 - 2010 : B.S. in Computer Engineering,
Sungkyunkwan University
- 2010 - 2012 : M.S. in Computer Science and Engineering,
Pohang University of Science and Technology (POSTECH)
Thesis Title :
**시연 기반 진동 패턴 저작
(Demonstration-Based Vibrotactile Pattern Authoring)**
Advisor: Prof. Seungmoon Choi



Publications

1. Sunghwan Shin, In Lee, Hojin Lee, Gabjong Han, **Kyungpyo Hong**, Sunghoon Yim, Jongwon Lee, YoungJin Park, Byeong Ki Kang, Dae Ho Ryoo, Dae Whan Kim, Seungmoon Choi, and Wan Kyun Chung, "Haptic Simulation of Refrigerator Door," In *Proceedings of the IEEE Haptics Symposium*, pp. 147-154, 2012.
2. In Lee, **Kyungpyo Hong**, and Seungmoon Choi, "Guidance Methods for Bimanual Timing Tasks," In *Proceedings of the IEEE Haptics Symposium*, pp. 297-300, 2012.
3. Hojin Lee, Gabjong Han, In Lee, Sunghoon Yim, **Kyungpyo Hong**, and Seungmoon Choi, "Effect of Active and Passive Haptic Sensory Information on Memory for 2D Sequential Selection Task," In *Proceedings of the International Symposium on Ubiquitous Virtual Reality (ISUVR)*, pp.52-54, 2011.
4. 홍경표, 이인, 한갑종, 최승문, "다중 감각 도움을 이용한 드럼 연주 기능 교육 시스템", 한국 HCI 학술대회 논문집, pp. 161-163, 2011.
5. 이호진, 한갑종, 이인, 임성훈, 홍경표, 최승문, "이차원 순차 선택에서의 능동 및 수동 햅틱 정보 제공을 통한 기억력 향상", 한국 HCI 학술대회 논문집, pp. 248-250, 2011.
6. 홍경표, 서종만, 김현경, 최승문, "모바일 기기에서 두 개의 선형 공진 진동자 를 사용한 합성 진폭 변조의 인지적 특성", 한국 HCI 학술대회 논문집, pp. 434-436, 2011.



7. 최승문, 이재봉, 흥경표 “진동 패턴 생성 방법 및 장치”, 10-2012-0035575,
2012.04.05 (출원중).

