

Huffman User Interface for Full Touch Screen Based Mobile Phones

Jae Gon Kim, Jun-Dong Cho

Mobile Systems Engineering Department, SungKyunKwan University

gon9031@skku.edu, jdcho@skku.edu

Abstract— In this paper, we propose a novel Huffman user interface for full touch screen based mobile phones. In the last decade, the mobile phones, referred to as software systems operating on mobile devices, have rapidly evolved from a simple phone to a complex multimedia interactive system device. The proposed Huffman user interface is following Huffman tree theory. Android come out with the new open and comprehensive platform for mobile devices. This user interface with android platform shows synergy. The results show this novel user interface is better than former user interface after comprehensive experimentations.

Keywords— User Interface, Huffman Coding, Mobile Phone, Android Platform, Full Touch Screen

I. INTRODUCTION

Recently, Full touch screen based mobile phones have become increasingly popular in Korea. However, it is hard to display the entire list of menu items on the whole screen of mobile phones because the number of menu items usually bigger than the lines on the screen [1]. Now days, most of menu systems is hierarchical. So, the users must navigate through a series of screens to find a menu item what they required. This process demands to integrate information across the screens during the navigation. So, this process can boost memory load [2].

In this paper, we propose a novel Huffman user interface for full touch screen based mobile phones. User interface design patterns are good solutions to standard user interface design problems. The proposed Huffman user interface is following Huffman tree theory [3]. With this user interface, the users can select menu item, which is using frequently, promptly with no trouble. This user interface with android platform shows synergy. Android platform offers intend function and uses category its own. Android come out with the new open and comprehensive platform for mobile devices. It includes an operating system, middleware, user-interface and applications. It is maker spanning and able to run on every cell phone and notebook. The results show this novel user interface is better than former user interface after comprehensive experimentations.

The organization of this paper is as follows. Section 2 presents some menu system related to the full touch based mobile phones and Section 3 introduces the proposed new user interface and methodologies for evaluating usability.

Then, experimental results are compared with the other menu systems in Section 4 and concluding remarks are given in section 5.

II. REVIEW ON CURRENT FULL TOUCH BASED MENU SYSTEMS

A. iPod Touch

The iPod Touch has a slim rectangular outline with curved edges, with a glass touch screen display covering most of the top surface and a home button off the touch screen. The display functions likewise to the multi-touch track pad as implemented in Apple's current line of PC. The touch and gesture features of the iPod Touch are based on technology formerly developed by Finger Works [4].

The menu system of the iPod Touch is very simple. The main home screen of the iPod Touch shows all main menu items on the top screen. One regularly method used for displaying several tools and an information source on the main screen is a so-called 'portal'. The benefit of the portal is that it provides dashboard-like overviews of the user's information. However, this portal system of iPod Touch exposes its weak points. First, this menu system has limitation for users not to expand their items up to five-page screen. Of course, this system contains sub-menus. However, most users want to find item at once. Sometimes, users often forget where the item is. This is the second weak point.

B. Samsung Jet

Samsung Jet shows user interface that user can customize the screen with rooted widgets such as the clock, scheduler, and other useful applications. Samsung Jet features intuitive 3D media gate menu system and motion-response user interface [5].

To enhance the usability in full touch screen, Samsung introduces 'smart unlock', an attribute which enables users to unlock the handset and perform the menu in a straight line by illustration an alphabet letter on the screen. Users can set up the menu that they want to go to directly according to the alphabet letters from A to Z. The 'smart unlock' system allows users to programmed their own, personalized actions in order to run the handset. However, it is sometimes hard to use. Most people' cursive script is different. Therefore, this system misses many requests.

C. LG Arena

The LG Arena is an LG Electronics multimedia handset. The LG Arena is the first phone to feature LG's new 3D S-Class user interface. The S-Class is a touch-based 3D user interface that lays out menus as if they were on a layer reel, enabling you to drag your finger across the reels to scroll through the accessible options. It is based around a cube layout that provides four home screens that can be modified with different shortcuts [6].

The user interface reacts to the orientation of the Arena, switching between portrait and landscape. However, these user interface and menu systems are almost same as iPod Touch's. Therefore, this system shows similar demerits as iPod Touch.

D. Google Android based Mobile Devices

The term "Android" has its origin in the Greek word *andr-*, meaning "man or male" and the suffix *-eides*, used to mean "alike or of the species". This together means as much as "being human." Android platform is a Java-based operating system. The system is very lightweight and full featured. Android is a software stack for mobile devices which means a reference to a set of system programs or a set of application programs that form a complete system. This software platform provides a base for applications just like a real working platform [7].

Android has a three-page main menu as like Samsung Jet, also, widget based system. Therefore, this system shows similar merits and demerits as Samsung Jet.

III. OUR PROPOSED MENU SYSTEM AND METHODS

A. Our Huffman User Interface

New user interface styles are usually created only in response to some specific customer needs, or to solve expected problems in an existing user interface styles. Now the same user interfaces are being extended with new features. If the user interface solutions are flexible enough, novelties can be realized in a way that is reasonably serviceable as long as we look at each one in isolation. However, the amount of new features in itself adds extra layers to the user interface. There will be new menus, new settings, and longer lists of options. The right ones will be more and more difficult to find. The overall complexity increases, and the old interfaces grow dense. Therefore, our novel user interface was planned based on the existing touch screen based mobile phones.

The menu execution module executes a relevant menu item under the control of the touch event control module. For example, when the user releases the touch at a current touched menu item "create message", the touch event control module requests the execution of "create message" from the menu execution module, and thus the menu execution module may execute an application corresponding to "create message".

Our user interface is based on 'sliding' and 'Huffman theory'. First, sliding is executed whenever we find a menu rather than just screen touch once. Figure 1 and 2 explain our user interface compared with the existing user interface.

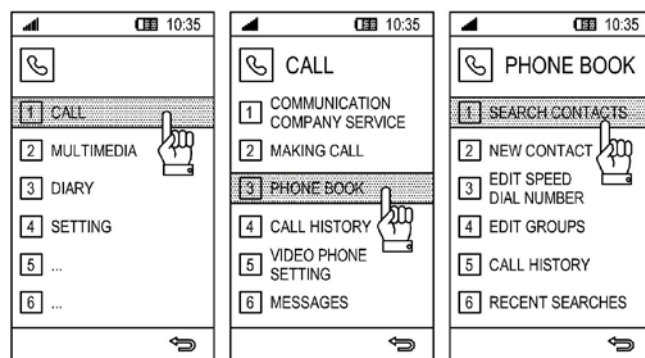


Figure 1. Conventional user interface system

Figure 1 is diagrams showing a method of selecting a menu item from a tree-structured menu using a touch screen in a conventional mobile phone. This is an inconvenience in that the user must touch the touch screen of a mobile phone more than several times to select a desired menu item. Furthermore, there is a problem in that, as the number of touches increases to select a menu item, the lifespan of a touch screen is shortened. Furthermore, there is a problem in that, since buttons must be pressed or the screen must be touched several times, a lot of time is required in order for the user to select a desired menu item.

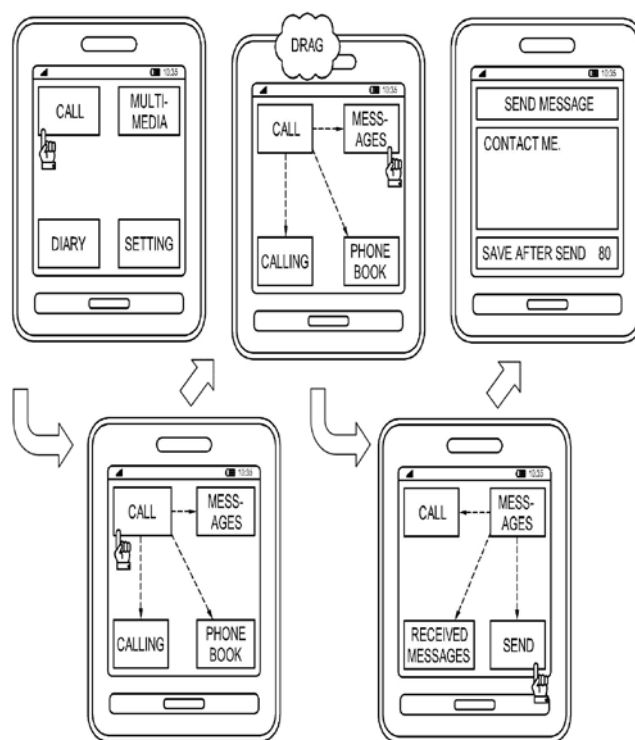


Figure 2. Our user interface using 'sliding' by dragging finger

As you show in Figure 2, sliding is Simple to use and also unlikely to miss wanted items. First, it can be seen that the highest level menu items, such as "call", "multimedia", "diary" and "setting", are displayed on the display device of the touch screen. The second picture illustrates the operation of the case where the user touches a "call" menu item from among the

highest level menu items. In this case, the screen displays lower level menu items belonging to the current menu item on which the touch was made. Thereafter, in order to select the lower level menu items, such as “messages”, “calling” and “phone book”, the user drags the “call” icon to user’s desired lower level menu item with a finger or a stylus in the state in which the “call” icon is being touched. The third picture illustrates the state in which the user performs a drag from the “call” menu item to the “messages” menu item, which is user desired menu item, while touching the touch screen on which the “call” menu item is displayed. Meanwhile, the handheld terminal displays “received messages” and “send” menu items, which are lower level menu items of the “messages” menu item, on lower left and lower right portions of the screen, in accordance. Of course, the locations at which the lower level menu items are displayed can freely be changeable. Final illustrates the results of the command corresponding to the “send” menu item executed by the mobile. As a result of the execution of the application, a “send message” menu item, a message field into which the user enters message content desired to be sent, and a select option such as “save after send” which enables a sent message to be saved after being sent, are displayed. Also, the mobile waits for the user to enter content into the message field or input the select option.

We can suggest using our new menu trees by Huffman theory, which is the basis of this simple technique to allocate shorter codes for things that are mentioned more frequently and longer codes for less common things. By applying the concept of a Huffman theory to our user interface, where we use shorter sequences of actions for common tasks, we can willingly come up with a better result than current user interface. While the idea of Huffman coding originates from the field of data compression, the ideas can be beneficially applied to our user interface design, leading to applications that are easier to user. Our sample menu trees are organized to mind the mobile phone. The menu item, which include higher frequency of using (i.e. ‘call’), composed by short menu tree, and the opposite case constituted by long menu tree. Figure 3 shows our sample suggested menu tree.

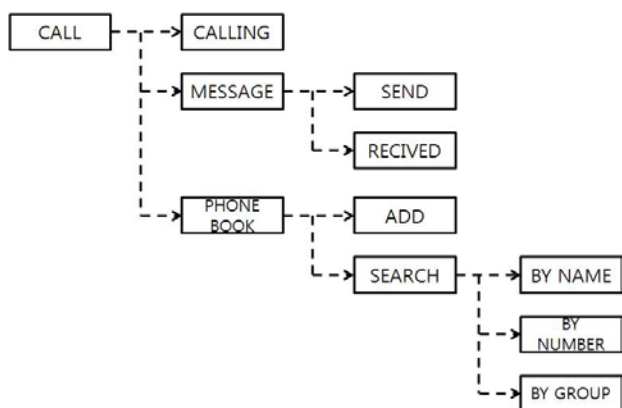


Figure 3. Sample suggested menu tree

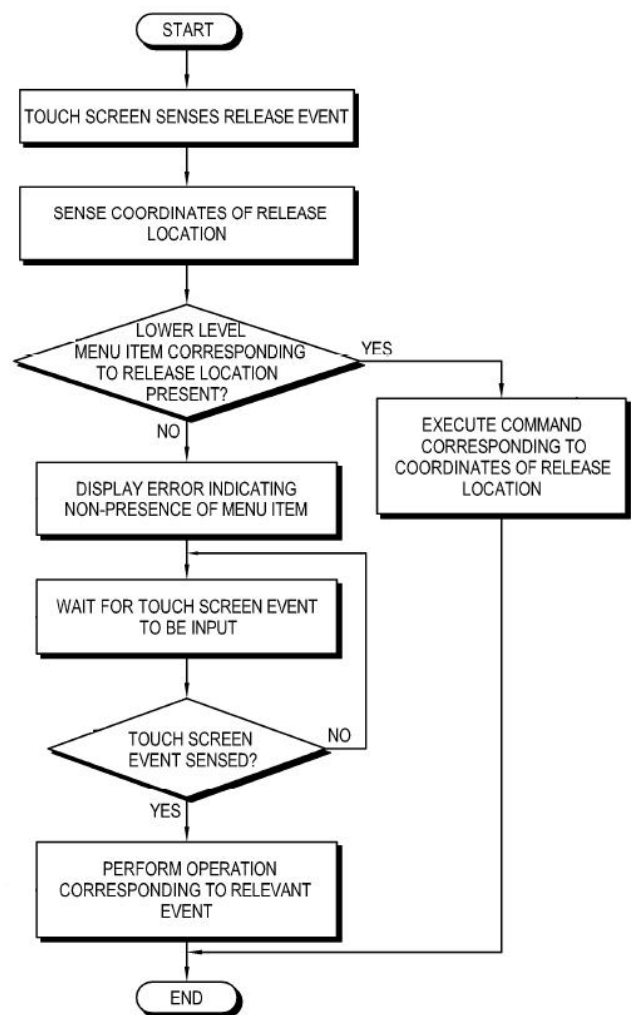


Figure 4. The method of controlling a mobile phone based on a release event

In Figure 4, the touch sensing device of the touch screen senses a release event. Accordingly, the control unit of the mobile phone receives the coordinates of the location at which the release was performed from the touch sensing device. The mobile phone determines whether a lower level menu item corresponding to the location at which a release, ending a touch, was performed. If it is determined that the menu item corresponding to the location at which the release was performed is present, the mobile executes a command or an application corresponding to the menu item. That is, the case whether a release was performed in a region in which icons or images corresponding to respective menu items are not displayed may exist. In this case, the screen may display an error message indicating that a relevant menu item or command is not present. Thereafter, the mobile preferably waits for another touch screen event to be input from the user. The reason for this is that, if setting is made such that the mobile phone is returned to the initial state thereof after displaying the error message, it is difficult to recover the place in the menu selection process where the user was at when the user releases

the touch by mistake. The mobile phone senses again whether a touch screen event has occurred.

In Figure 5, the touch sensing device of the touch screen senses the input of a drag event from the user. In this case, the control unit of the mobile phone receives the coordinates of the location at which a drag was terminated from the touch sensing device. The mobile phone determines whether a menu item corresponding to a location at which the drag was terminated is present. If it is determined that a menu item corresponding to the termination location of the drag is not present, the screen displays an error message indicating that a lower level menu item is not present, and waits for a subsequent touch screen event to be input. If it is determined that a menu item corresponding to the termination location of the drag is present, the screen displays lower level menu items belonging to the menu item or information about the menu items. For example, when a drag is terminated on the icon of the album of a specific singer, information about the album may be displayed. Further, the lower level menu items or information related to the drag termination location are preferably displayed close to the previously displayed menu item.

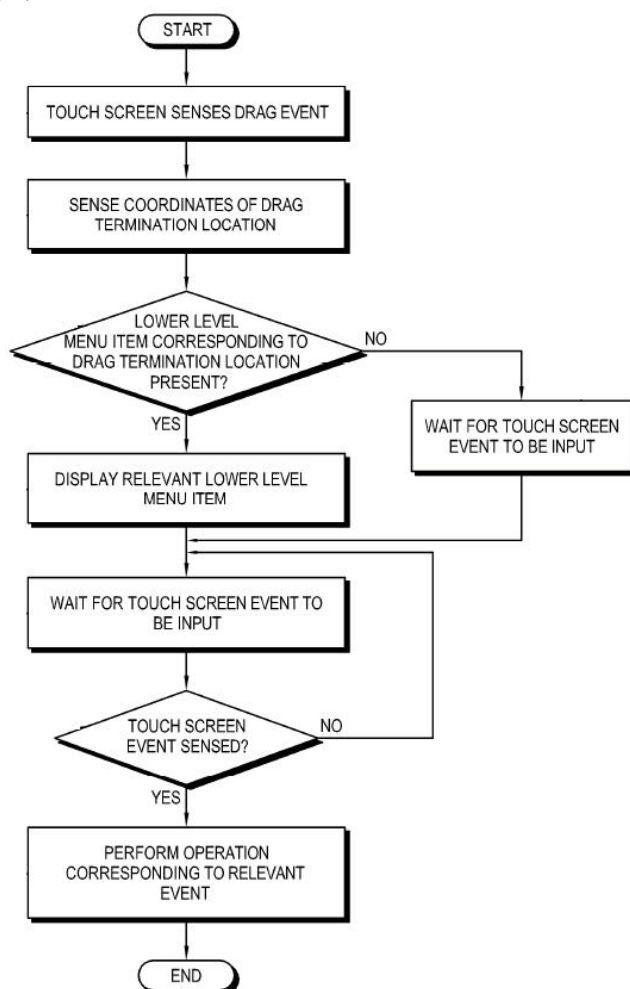


Figure 5. The method of controlling a mobile phone based on a drag event

Android platform shows in various merits. First, Google Android is open source software. In theory this means that the Google Android software platform will have a spirited benefit over other operating systems. Second, the ability for anyone to modify the Google android platform will open up the applications playing field to small. The customer will benefit from having a broad range of mobile applications to choose from since the monopoly will be broken by Google Android. Except these, Android platform give interested functions to developers like Intent and Category.

Intent is a conceptual explanation of an action to be performed. Intent provides a capability for performing behind runtime binding between the codes in different applications. Most important use is in the launching of activities, where it can be thought of as the join between actions. It is essentially a passive data formation holding a conceptual description of an action to be performed. An Intent Filter can match against actions, categories, and data in Intent. It also includes a priority value which is used to arrange multiple matching filters. For example, it might suggest a request for an activity to present an image to the user of let the user edit some text. For broadcast receivers, the Intent object names the action being announced. For example, it might announce to interested parties that the camera button has been pressed.

Any number of category descriptions can be placed in an Intent object. For Intent to pass the category test, every category in the Intent object must match a category in the filter. The filter can list additional categories, but it cannot pass over any that are in the intent. In principle, therefore, an intent object with no categories should always pass this test, regardless of what is in the filter. That is mostly true. Using these properties, we built our menu trees easily. Most similar menu items are sorted by category and we can portal these items by Intent. Figure 6 shows our user interface feature.

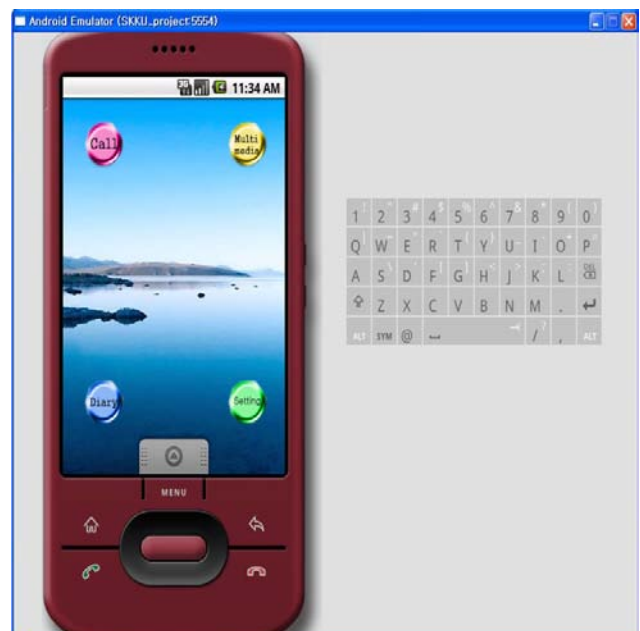


Figure 6. Our user interface main screen using Android emulator

B. Participants

Thirty-two people voluntarily participated in the research. Men are twenty and women are twelve. Teenagers are ten, twenties are thirteen, and thirties are nine. To minimize the personal property of prior experience, the participants who had a familiarity with the menu system in this study were barred from the test. To examine the effects of age, participants were classified into three age groups. On the average, participants in their teens, 20s, and 30s used menu systems of the mobile phones for about 10min, 12min, and 15min per day, respectively. The age difference between the three groups was about 5.2years. This is generally accepted in Korea that behaviour of young age bracket varies with a difference in ages [8].

C. Experimental Device

A user interface example of a mobile device was developed using Android emulator supported by Google. The prototype device emulator ran on an IBM-compatible personal computer with an Intel Core 2 duo 2.66 GHz processor, and it was presented on monitor in Android emulator. Emulator and simulator allow coders to view their application on the computer. The size of the emulator screen was 3.2-inch TFT-LCD flat e screen with 320 x 480 (HVGA) resolutions, which is same as the Android develop phone. This emulator usability test was run identically to desktop application usability test. Participants interacted with this emulator by clicking the emulator screen with the mouse.

D. Experimental Task and Process

Participants were briefed on the experimental tasks and situation before starting the experiment and they performed practice trials to become common with the tasks. In the main experiment, participants were instructed to reach the target menu item as rapidly and properly as possible.

Three different search target items were given for each behaviour situations, which makes 15 trials for each participant. Three task scenarios were tested. First is 'Message'. Message sending scenario is starting at the idle screen, enter the message sending screen and input "12345678" at the first line. Second is 'Camera'. Take a picture scenario is staring at the idle screen, enter the camera screen and take a picture. Third is 'Multimedia'. Play MP3 music scenario is starting at the idle screen, enter the MP3 playing screen, play the first song and after 30 seconds, pause the playing. After conclusion of each statement condition, participants were asked to rate the simplicity of search and the personal gratification before going on to the next search state. Upon finishing all the situations, they were asked to give their remarks on the test.

IV. EXPERIMENT RESULTS

A two-way ANOVA (analysis of variance) was used to appraise the effect of trial factors and their relations on the dependent variables. The two-way ANOVA is an extension to the one-way ANOVA. In this experiment, we used two factors. One is age of users and the other is type of user

interface. The results are summarized in Table 1. Differences between the factor levels were examined by the SNK (Student Newman Keuls) test in the analyses. Measuring usability attributes in mobile phone was referred from Zhang [9].

Table 1. Summary of the Analysis of Variance Results (p-Values)

Factors	Task Completion Time	Number of Screen Touches	Simplicity of Search	Gratification
Type	< 0.001	0.12	< 0.001	< 0.001
Age	0.173	< 0.001	0.013	< 0.001
Type × Age	0.585	0.347	< 0.001	< 0.001

*Significant of $\alpha = 5\%$

The mean task achievement time is shown in Figure 7. The type of menu system had a significant effect on the task achievement time, $p < 0.001$. According to the result of the SNK test, it took significantly shorter with using our user interface than conventional at the alpha level of 5%. There was no important difference when consider together age of users and type of user interface. In this result, teens and 20s are more adapted to use new user interface than 30s.

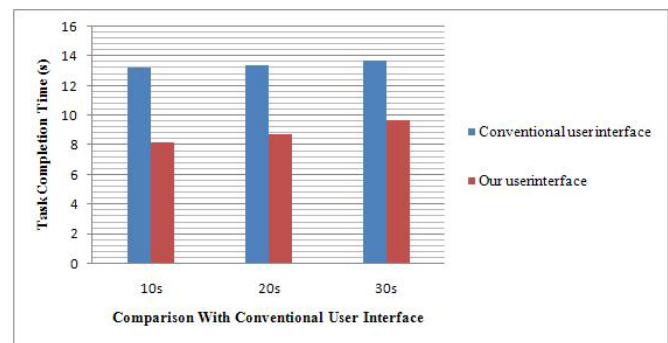


Figure 7. Task completion time comparisons with conventional user interface

The mean number of screen touches is shown in Figure 8. Both the participant age and the user interface type had major effects on the number of screen touches. The 30s ($M = 13.5$ touches and $M = 11.6$ touches respectively) touched screen most many times, and 10s ($M = 12.7$ touches) touched most few times when using conventional user interface and 20s ($M = 11.7$ touches) touched most few times when using novel user interface. 30s shows many mistakes in their search when they touch and sliding.

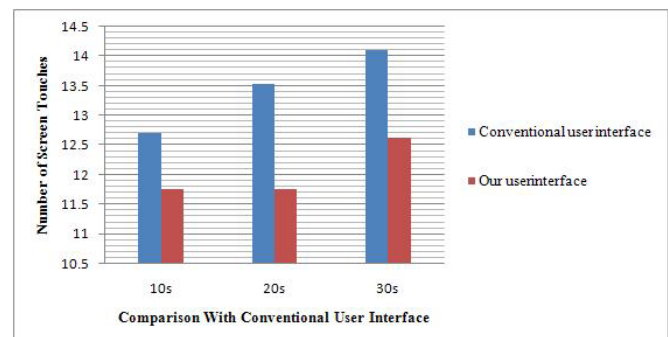


Figure 8. Number of screen touches comparison with conventional user interface

Judged simplicity of search was affected by the type of menu item presentation, $p < 0.001$. The target item was easier to search when using our user interface than conventional. As shown in Figure 9, teens show the significant difference between our user interface and conventional. However, 30s felt similar simplicity when they used both user interfaces.

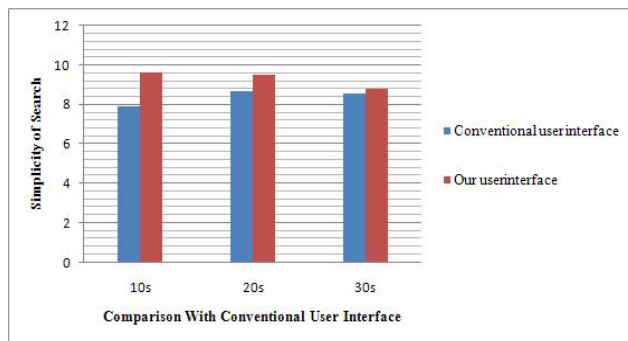


Figure 9. Simplicity of search comparison with conventional user interface

The participants preferred our novel Huffman user interface and felt fun when sliding between menu items. The gratification was generally higher when using our user interface. Especially, teens gratified more than 20s and 30s when using our user interface than conventional. Figure 10 shows the user gratification level. Teens felt that conventional user interface is boring; however our user interface is so fresh.

Users' menu search performance was analysed based on the result of the test. The effects of the various settings of the menu item presentation on the performance and favourite were examined, and the reimbursements and drawbacks of using this system for menu item search are discussed. The participants preferred our novel Huffman user interface even though there was no difference in performance between the two. Both the judged simplicity of search and the user satisfaction were higher with our menu system than normal system. 90% (29 of 32) of participants reported that they preferred our user interface. The major reason was that it is fast and easy to use after learning our system.

The results showed that our user interface shows good simplicity to use. The participants preferred our novel Huffman user interface and felt fun. The type of menu system had a significant effect on the task achievement time. Most participants are finished their task quickly when they used our user interface. This result may be attributable to the changes in users' item search strategy.

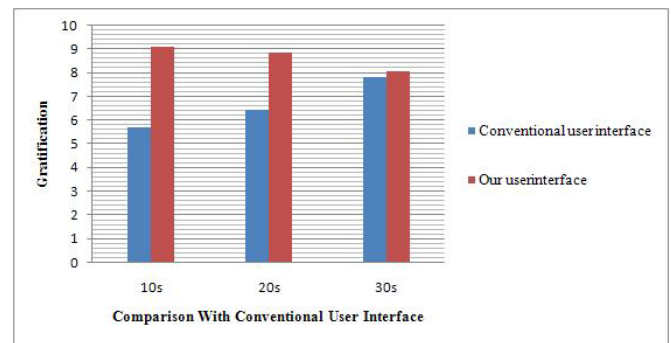


Figure 10. User gratification comparisons with conventional user interface

V. CONCLUSIONS

This study proposed a novel Huffman user interface as a menu system for enhancing the usability of full touch screen based mobile phones. The results of the study have led us to conclude that our user interface could decrease time to find target menu item and improve the user gratification, when presented correctly. The results showed that our user interface shows good learnability and simplicity to use. This result may be attributable to the changes in users' item search strategy.

The novel user interface menu system proposed in this study is expected to find target menu in efficient and effective use of poorly organized menus. Further studies are needed to sort the hierarchical structure of menu items by Huffman theory, and the implications of higher level menu items. In addition, the differences between age groups could also be examined for extra age groups. Finally, we should mention that more constant efforts are required to revise the usability as novel mobile technologies come forward.

REFERENCES

- [1] Tang, K.-H. E, "Menu design with visual momentum for compact smart product," *Human Factors.*, vol. 43, pp. 267-277, Nov. 2001.
- [2] Ziefle, M, "The influence of user expertise and phone complexity on performance," *Behavior and Information Technology.*, vol. 21, pp. 303-311, Nov. 2002.
- [3] Huffman, D. A, "A method for the construction of minimum redundancy codes," *In Proc. IRE.*, vol. 40, pp. 1098-1101, Feb. 1951.
- [4] Apple Inc. [Online]. Available: <http://www.apple.com/ipodtouch/>
- [5] Samsung Electronics Co. [Online]. Available: <http://jet.samsungmobile.com/>
- [6] LG Electronics. [Online]. Available: <http://arena.lgmobile.com/uk/>
- [7] Android Official Website. [Online]. Available: <http://www.android.com/>
- [8] Lee, J. "Lifestyles and media usage of teenagers." *Advertisement Information (Korea Broadcasting Advertising Corporation)*, pp.128-133, Apr. 2003.
- [9] Zhang, d & Adipat, B. "Challenges, methodologies, and issues in the Usability testing of mobile applications." *International Journal of Human-Computer Interaction*, vol. 18(3), pp. 293-308, Dec. 2005.