JESK
J Ergon Soc Korea 2016; 35(4): 277-291
http://dx.doi.org/10.5143/JESK.2016.35.4.277
http://jesk.or.kr eISSN:2093-8462

A Unit Touch Gesture Model of Performance Time Prediction for Mobile Devices

Damee Kim, Rohae Myung

Korea University, Department of Industrial Management Engineering, Seoul, 02841

Corresponding Author
Rohae Myung
Korea University, Department of Industrial Management Engineering, Seoul, 02841
Mobile : +82-10-8915-3392
Email : rmyung@korea.ac.kr

Received : April 01, 2016
Accepted : May 17, 2016

Objective: The aim of this study is to propose a unit touch gesture model, which would be useful to predict the performance time on mobile devices.

Background: When estimating usability based on Model-based Evaluation (MBE) in interfaces, the GOMS model measured 'operators' to predict the execution time in the desktop environment. Therefore, this study used the concept of operator in GOMS for touch gestures. Since the touch gestures are comprised of possible unit touch gestures, these unit touch gestures can predict to performance time with unit touch gestures on mobile devices.

Method: In order to extract unit touch gestures, manual movements of subjects were recorded in the 120 fps with pixel coordinates. Touch gestures are classified with 'out of range', 'registration', 'continuation' and 'termination' of gesture.

Results: As a results, six unit touch gestures were extracted, which are hold down (H), Release (R), Slip (S), Curved-stroke (Cs), Path-stroke (Ps) and Out of range (Or). The movement time predicted by the unit touch gesture model is not significantly different from the participants' execution time. The measured six unit touch gestures can predict movement time of undefined touch gestures like user-defined gestures.

Conclusion: In conclusion, touch gestures could be subdivided into six unit touch gestures. Six unit touch gestures can explain almost all the current touch gestures including user-defined gestures. So, this model provided in this study has a high predictive power. The model presented in the study could be utilized to predict the performance time of touch gestures.

Application: The unit touch gestures could be simply added up to predict the performance time without measuring the performance time of a new gesture.

Keywords: Keystroke Level Model, Usability, Touch gesture, Model-based Evaluation (MBE), HCI (Human-Computer Interaction)

1. Introduction

Designers or experts carry out UT (usability tests) to minimize errors in designing interfaces (Bennett et al., 1989; Jokisch et al., 2011). Usability means use effectiveness, efficiency and satisfaction to achieve a specific goal (Guidance on Usability; ISO 9241-11). Interface designers improve usability through repetitive evaluation process of the systems causing many errors. However, such a repetitive evaluation process has a weakness causing temporal and economic costs (Landauer, 1995). To overcome such a weakness, Card et al. (1980) applied a Model-based Evaluation (MBE) method to usability evaluation. MBE is to conduct usability evaluation on a system to develop through a model. This is a method that can acquire scientific evaluation results with smaller cost and time in the initial stages of system development (Kieras, 2002). Because existing usability evaluation is carried out by actual users, a lot of time and cost were required for manufacturing a prototype, and selecting and training users. However, MBE not only cuts cost in the initial stage evaluation, but reduces total system development period, since it does not require an experimenter. MBE has a strength that scientific evaluation is possible based on cognitive engineering and psychology (Olson and Olson, 1990), unlike existing usability evaluation methods with lacking theoretic basis (Lewis et al., 1990; Sears and Jacko., 2009; Kieras, 2002). Based on such researches, MBE started to be used for interface evaluation research as well (Bennett et al., 1989; Ghasemifard et al., 2015).

The MBE method includes EPIC (Kieras and Meyer, 1997), ACT-R (Anderson and Lebiere., 1998) and GOMS (Card et al., 1980; John and Kieras, 1996a, 1996b; Kieras, 1999). Among these, GOMS is a model developed by Card, Moran and Newell, and is especially used a lot for interface research, since it is easy to use, and can be easily learned. GOMS is the abbreviation of goal, operator, method and selection rule, and it explains the interactions between humans and systems with the following four factors: goal, operator, method and selection rule. Goal is a symbolic component on what a user wants to achieve, when the user uses a system. Operator means user's action affecting the system. Method means a continuous process in the skilled stage of sub-goals to achieve a goal. Selection rule means a rule through which a user selects a method. Through these components, the framework by which interface designers can systematically analyze usability problems that users face is offered (Irving et al., 1994).

The GOMS model defines user's action affecting the system as an operator, and the examples include input using the keyboard, mouse clicking and button pressing. The performing time of GOMS operator is the performing time extracted based on desktop-based input device. The reason is that WIMP (window, icon, mouse and pointing device)-based GUI (graphical user interface) was mainly used in the 1990s when GOMS was developed. However, the environment is currently changing to NUI (natural user interface) in which user's action is directly perceived like a touch-based mobile device (Back et al., 2015). Therefore, the performance time extracted based on desktop environment at the time of GOMS development has such problems as the lack of the number of operators (Lee et al., 2011) and decline in predictive power of performance time in explaining and predicting the various current HCI environments. A study of Lee et al. (2011) points out the operator in the existing GOMS model may have low precision, since the performance improvement of computers and mobile devices, and learner's device learning and familiarization are not considered. For these reasons, the need for revised research on GOMS operator emerged. Studies on revised models on touch gestures reflecting the mobile environment are actively carried out recently, when smart devices such as cellphones and tablet PCs embedded with a touch screen are vigorously released in the markets (Amant et al., 2007; Holleis et al., 2007; Lee and Myung, 2009; Back and Myung, 2011; Choi et al., 2013). These studies predicted gesture performance time by revising or re-defining the existing operator. A study of Holleis et al. (2007) presented a keystroke level model by extracting various operators' performance time used upon mobile interaction targeting mobile devices in which cellphone screen and input pad are classified. However, the model has a weakness in that it can be applied to only specific devices. With such a research method, performance time needs to be revised or re-defined, whenever a new gesture is generated. To overcome such a limitation of the existing study method, an integrated method that can explain all existing gestures is needed, and the method should be extended for an operator to be used, when a new gesture is defined. This study analyzed small unit gesture constituting one gesture, and defined it as a unit touch gesture. The unit touch gesture extracted in such a manner enables to predict performance time of the gestures not defined yet, as well as existing gestures. The aim of this study is to extract the unit touch gesture, and define it as a model.

1.1 GOMS

The GOMS model is an evaluation method predicting how a user uses a system, and is used for prediction of user's system performance time in each stage. GOMS explains human's actions as four factors: goal, operator, method and selection rule.

Goal is a symbolic component on what a user wants to achieve using a system. Operator means user's action affecting the system, and the examples include input using the keyboard, mouse clicking, button pressing and menu selection. Table 1 is concerned with manual operators provided by the current GOMS model. Through six manual operators, Table 1 presents a prediction model for performance time prediction using the keyboard and mouse on the desktop.

Table 1. Exiting GOMS manual operator

Operator	Description	msec
K	Keystroke or button press	280
P	Pointing to target on a display with a mouse	1100
H	Homing the hands on the keyboard or other device	400
HD	Hold down to device	100
R	Release from device	100
C	Click to object	200

$$T_{execution} = T_k + T_p + T_h + T_{hd} + T_r + T_c \quad (1)$$

Manual operator performance time can be predicted through the sum of the operators used in a specific work. It can be expressed as equation (1). Method means the continuous process of skilled stage of sub-goals to achieve one goal. Selection rule means the rule of user's selection of the method. GOMS assumes a skilled-person not to make an error in performing the task to predict human's performance time. Thus, GOMS is an applicable model under the premise that a mistake or a trial and error is not caused under the clear goal. GOMS model was developed by Card, Moran and Newell. The GOMS model has the following types: CMN-GOMS (Card, Moran and Newell GOMS), KLM-GOMS (Keystroke Level Model GOMS, Moran et al., 1983), CPM-GOMS (Cognitive Perceptual Model GOMS, John, 1988) and NGOMSL (Natural GOMS Language, Kieras, 1999) KLM-GOMS used in this study is a technique based on human information processing's subsequent model as a simpler method than other GOMS models. This is a model predicting performance time with the sum of basic keystroke levels (keystroke level model) required for task performance. This study is to present methodology to predict touch gesture's execution time, and therefore this study was focused on the operator among the four components of GOMS.

1.2 Touch gesture

Touch gesture is a manipulation mode enabling interaction between a user and a computer through user's movement without an existing input device. It has a merit that intuitive and natural control is possible, due to user's direct interaction (Hong and Woo, 2008). GOMS defined an input device as an operator, and provided performance time. An operator is performance time based on the input device, namely, keyboard and mouse at the time of the model development. Since the unit touch gesture to be presented in this study reflected touch gestures through fingers, which is an input mode, the most used input mode in the HCI environment, the concept of operator in GOMS has been applied by reflecting the current HCI use environment.

According to Wigdor and Wixon (2011), touch gestures can be classified into four stages: First stage is registration stage, which is the moment when a gesture is perceived in a device. Second stage is continuation stage in which movement occurs, after registration. The third is termination stage in which gesture is terminated. The fourth stage is the out of range stage preparing in the state that a hand does not contract the device. A user performs a gesture through the four stages mentioned above. The representative gestures used a lot by users are shown in Figure 1. As such, one gesture can be segmented according to the performance aspect of gesture.

A recent new trend of touch gesture is user-defined gesture. Concerning the user-defined gesture, a user newly defines and uses the required gesture suitable for situation in addition to existing gestures. The gesture made in designing a system cannot be always the optimum gesture. And, the user-defined gestures were presented as a method to solve the problem that the gestures proposed by researchers are sometimes comprised of complex and time-consuming motions (Morris et al., 2010). System developers offer user-defined gesture function so that mobile device users can define gestures suitable for their own use environment. Users can improve the given mobile environment, and use the environment more efficiently through the function. As such, the environment using touch gesture becomes diverse environments, and it gradually becomes difficult to predict user's performance time with the existing touch gestures. According to the current studies on operators, only performance time prediction on the existing gestures is possible, and the studies that can predict the performance time on the gestures to be developed are insufficient.

2. Method

This study extracted unit touch gestures through the process as shown in Figure 1. To extract unit touch gestures, this study selected representative touch gestures among the touch gestures used by mobile device users. Most gestures consisted of basic gestures, such as tap, flick, drag, zoom-out and zoom in, and the gestures that repetitively use or apply the basic gestures. For example, if a basic gesture, tap, is conducted twice, it becomes double tap, and dragging the tapped state becomes the performance of press and drag. Due to such a reason, this study removed repetition and application factors, and selected representative touch gestures. Based on them, touch gesture-performance image is examined through video analysis. Using the results through a video analysis, the unit touch gesture was extracted, and the performance time was defined, and the unit touch gesture model was presented. Lastly, this study carried out a verification experiment targeting the currently existing gestures and user-defined gestures used by users by defining them, based on the unit touch model.

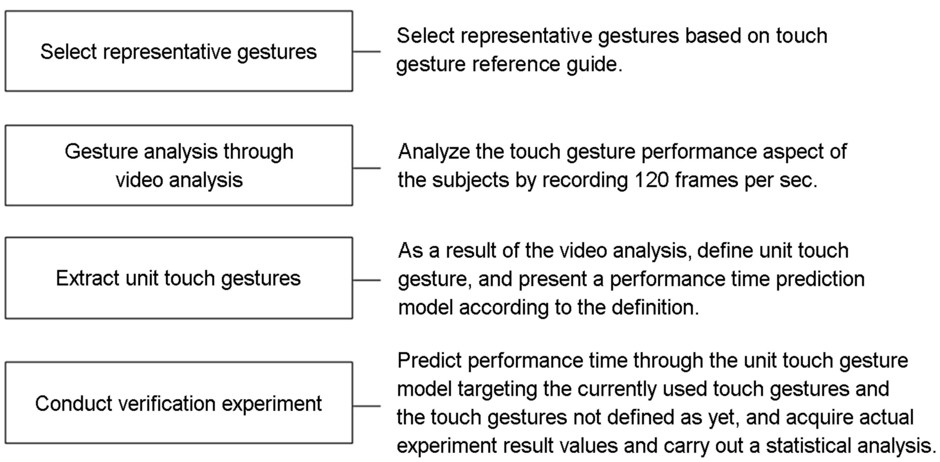

Figure 1. A framework of methodology

2.1 Select representative gestures

To extract unit touch gestures, a process to analyze the currently used gestures and extract components is needed. As a result of analyzing gestures, based on the Touch Gesture Reference Guide of Villamor et al. (2013) that arranged currently used gestures, most gestures consisted of the repetitively used and utilized basic gestures including such basic gestures as tap, flick, rotate, drag, zoom-out and zoom-in. For example, if tap, which touching the screen once and then off the screen, is conducted twice, it becomes double tap, and dragging the tapped state is to press and drag. If drag is carried out beyond the two fingers, it becomes a multi-finger drag, a gesture to turn over the screen. Due to such reasons, this study selected basic gestures by removing repetition and utilization factors. The basic gestures selected in such a manner are tap, flick, rotate, drag, zoom-out and zoom-in. The explanation of each gesture is as follows: Tap means a motion to lightly touch the screen with a finger, and then take if off the screen. Flick is a gesture quickly flicking the screen. Rotate is a motion rotating clockwise or anticlockwise, making a circle with fingers, and is used to cancel execution, or control photo rotation and moving image speed. Drag is a motion to draw a straight line with a finger, and then to take the finger off the screen. Zoom-out and Zoom-in are the gestures used for photo enlargement and reduction, and they mean maintaining contact with the screen using two fingers, and then pinching or spreading the two fingers. Figure 2 show the selected gestures, and Table 2 explains those gestures.

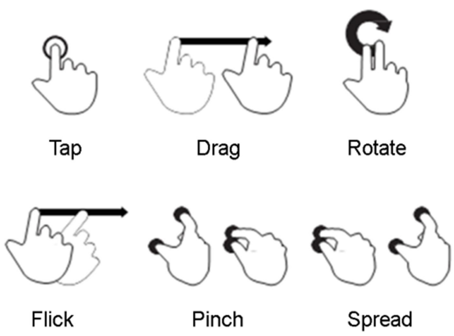

Figure 2. Representative touch gestures

Table 2. Definition and explanation of touch gestures

Gesture	Description
Tap	A motion to touch the screen with a finger simply
Drag	Move, while maintaining a contact with the screen with fingers
Rotate	Contact moving clockwise or anticlockwise with two fingers
Flick	A motion touching the screening as if one flicks the screen quickly with fingers
Zoom-out	A motion to be gradually closer, while maintaining a contact with the screen with two fingers
Zoom-in	A motion to be gradually farther, while maintaining a contact with the screen with two fingers

2.2 Video analysis

This study carried out a video analysis to extract a unit touch gesture comprising one touch gesture.

2.2.1 Subjects

The subjects consisted of ten undergraduate and postgraduate students aged 26.4 (±2.02): six males and four females. The ten subjects were those who were accustomed to using touch screen-based mobile devices, and their average use period was 5.6 (±1.8) years.

2.2.2 Experimental environment

This study video-taped subjects' gesture performance aspect with a speed of 120 frames per sec to classify motions in detail. Concerning the gestures performed by the subjects, this study developed a Web programing (html5, javascript)-based program to record input coordinates, coordinates on movement path, terminating coordinates and performance time in real time. The mobile device used for the experiment was ios 9.3-based iPhone 5S.

2.2.3 Experiment procedure

This study have the subjects perform gestures by selecting representative mobile jobs. They performed a representative job twice, respectively, using touch gestures as usual.

• Representative mobile task
- Text messaging ("사랑해": Test messaging "Love you" to the person him/herself)
- Path finding using the App for map ("운중동 - 안암역": Woonjung-dong - Anam Station)
- Web search ("인터페이스": Interface search)
- App downloading and deletion ("카카오톡": Kakao Talk downloading and deletion)
- Photo enlargement and reduction (Enlargement->rotation->Reduction in the photo album)

This study selected work mainly performed by using mobile touch devices at usual, and the representative mobile jobs are mentioned above. Starting from locked screen till each task is terminated, text messaging is performed through Flick x 1, Tap x 12; path finding through Tap x 28 , Zoom-in x 2, Zoom-out x 2, Flick x 5, Drag x 3; Web search through Flick, Tap x 15; Web downloading and deletion through Tap x 14; photo reduction/enlargement/rotation through Tap x 4, Flick x 3, Zoom-in x 2, Zoom-out x 2, Rotate x 2, respectively.

Therefore, each subject carried out the following gestures per performance of the task: Tap x 74 times, Flick x 9 times, Zoom-in x 4 times, Zoom-out x 4 times, Long tap x 1 times, Rotate x 2 times, Drag x 3 times in total.

2.2.4 Video analysis experiment results

This study analyzed the subjects' gesture movement coordinates and gesture performance aspects. The subjects performed gestures as shown in Figure 3 in relation with representative gestures. Tap touch gesture has no movement on the coordinates, but was performed with a motion touching the screen with a finger, and then taking the finger off the screen. This gesture was mainly used to select a menu or input text. Drag was carried out as a motion to touch the screen with a finger, and then take it off the screen. This gesture was used to slowly move on the screen or move a specific object. Rotate was performed as a motion to touch the screen with fingers, and take them off the screen, after moving clockwise or anticlockwise. This gesture was used to perform the cancellation of execution and rotate a photo. Flick was different from other gestures, and the motion to take fingers off the screen was not revealed. Flick is a motion to turn over the screen fast by definition, and was performed as a motion flying fingers in the air fast, unlike other gestures that take fingers off the screen, while fingers move. Therefore this study defined slip, a unit touch gesture, differentiated from P-stroke. Flick was used to cancel screen lock or turn over the screen fast. Zoom-out and Zoom-in are the gestures used to reduce or enlarge the screen. They were performed as the motions to take fingers off the screen, after touching the screen and moving the fingers to the desirable ratio in the straight line direction.

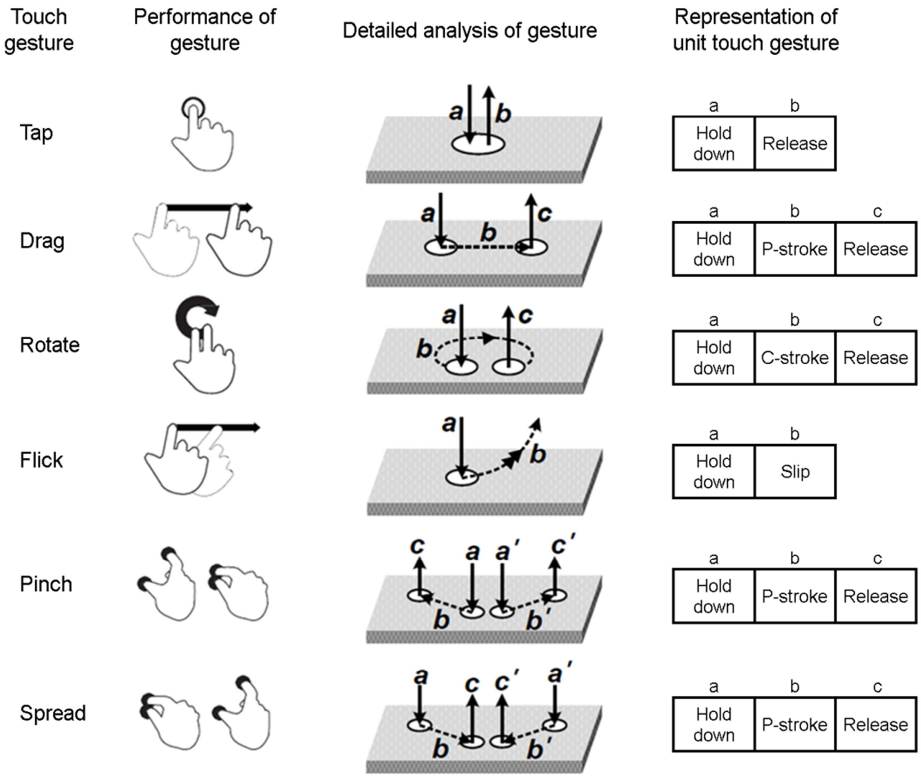

Figure 3. Video analysis result of tough gestures

Figure 4 shows the result of the subjects who carried out a task looking for desired path by inputting the starting point and destination on the App of map.

As a result of performing the video analysis of all the subjects' gesture performance, the gestures could be classified as follows, when they perform the gestures: 1) Gesture until inputting, 2) Gesture by which movements occur, 3) Gesture that is terminated, and that take fingers off the screen, 4) Gesture moving fingers to the next gesture from the outside of the touch screen. This was consistent with a study of Wigdor and Wixon (2011) reporting one gesture consists of beginning motion, continuing motion and terminating motion.

2.3 Extraction of unit touch gesture

The unit touch gesture's performance time was extracted on the basis of the video analysis result. Table 3 shows the performance time of unit touch gestures.

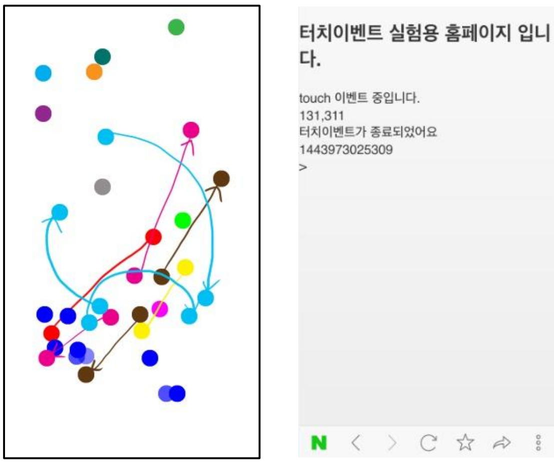

Figure 4. Example of video analysis result

Table 3. Unit touch gesture model

Unit touch gesture	Execution time (msec)	Std (msec)
Hold down (H)	54	6.81
Release (R)	54	9.39
Slip (S)	123	49.9
C-stroke (Cs)	620	75.8
P-Stroke (Ps)	544	26.3
Out of range (Or)	221	51.4

Hold down (**H**)

This is a unit touch gesture making a finger perceived on the screen. From the moment a subject's finger touches a touch-based device until the moment before a finger movement occurs, namely, performance time immediately before coordinates change occurs was extracted in this study.

Release (**R**)

As a motion taking a finger off the screen, "Release" is a unit touch gesture occurring when a gesture is terminated. Therefore, this study extracted performance time from the moment when there is no change of coordinates to the moment when a finger is taken off the screen, as a touch gesture from the movement of coordinates to immediately before a finger is taken off the screen.

Slip (**S**)

Slip is a unit touch gesture sweeping up a finger quickly from the state that a finger is on the screen. This study extracted the performance time from the movement to the moment immediately before taking a finger off the screen. Unlike P-stroke, the characteristic of "Slip" is to fly the finger into air. Therefore, this study extracted the execution time till the moment of interface's slip response with a subject's slip gesture through video-taping additionally.

C-stroke (**Cs**)
A motion demonstrated when rotation is conducted like "Rotate", and C-stroke is a unit touch gesture in which rotation continues until the motion is terminated in the finger-held down state. Therefore this study extracted performance time during the curved movement performance between hold down to release.

P-stroke (**Ps**)
A unit touch gesture in which straight line movement continues from the state of finger-held down before the motion is terminated. This study extracted performance time during the straight line movement performance between hold down and release.

Out of range (**Or**)
A gesture moving fingers outside of a mobile device to perform a touch gesture, although "Out of range" is not a motion to touch the mobile device. This motion is found, when several touch gestures are performed to carry out a task. This motion was also a confirmed motion, when one gesture is performed twice repeatedly like "Double Tap". Out of range was defined as the time when a finger stays in the air from the first tap terminating point in time, if a terminating gesture is performed twice like "Double Tap", and this study extracted the time.

$$T_{total} = n_1 \cdot T_h + n_2 \cdot T_r + n_3 \cdot T_p + n_4 \cdot T_c + n_5 \cdot T_s + n_6 \cdot T_o \qquad (2)$$

(Th: Hold down performance time, Tr: Release performance time, Tp: P-stroke performance time, Tc: C-stroke performance time, Ts: Slip performance time, To: Out of order performance time, n: The used number of unit touch gesture operator).

Therefore, the unit touch gesture model like equation (2) can be defined. The touch gesture's total performance time (Ttotal) can be predicted with the sum of multiplying each unit touch gesture's performance time by the number of performance of each unit touch gesture. For example, because Tab is Hold down + Release, and Double tap is Hold down + Release +Hold down + Release, the total performance time is 2xHold down + 2xRlease.

3. Experiment

The unit touch gestures and performance time could be extracted through video analysis results in this study. Based on this, this study carried out a validation experiment to validate whether the unit touch gesture model can precisely predict actual gesture performance time and new gesture performance time. The actual gesture performance time was extracted with the same experiment environment and task as the experiment environment. For performance time prediction on the gestures not defined yet, this study selected user-defined gestures that a user defines and uses them.

3.1 Subjects

The subjects consisted of ten undergraduate and postgraduate students aged 24.5 (±2.9) on average, and the subjects were composed of five males and five females. All the ten subjects were those who were accustomed to the use of touch screen-based mobile devices, and their average use period was 4.7 (±2.2) years.

3.2 Experiment Environment

The mobile device used for the experiment was is 9.3-based iPhone 5S. The subjects' gesture performance was video-taped with the unit of msec using a mini iPad 2.

3.3 Experiment procedure

This study made the subjects perform a task with the same number of repetition in the same unit touch gesture extraction experiment environment to validate the already existing gesture performance time (SMS, finding path using the App for map, photo zoom-out and zoom-in, and Web search). Based on the research on the gestures to be used in the future, and user-defined gestures (Figure 5), the subjects performed six gestures on the mobile device twice, respectively.

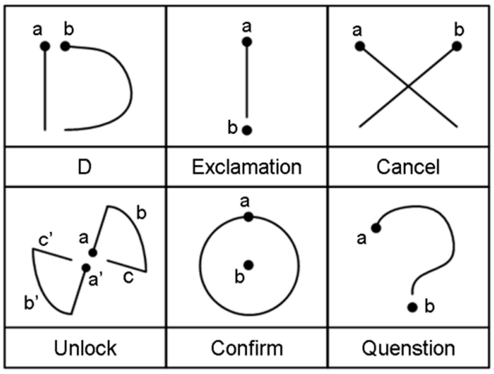

Figure 5. User-defined gestures

3.4 Experiment results

This study provided a unit touch gesture model so that GOMS operators can utilize it in the current touch gesture environment. To validate the statistical differences of the actual touch gesture performance time and prediction results by the unit touch gesture model, the *t*-test (degree of freedom: 9) was conducted. Table 4 shows the validation expe4riment results. As a result, it was confirmed that no significant difference was revealed between the subjects' gesture performance time and model's predicted performance time on the currently existing gestures (drag, *p*-value = 0.79; flick, *p*-value = 0.164; tap, *p*-value = 0.603; double tap,

Table 4. The result of validation experiment

	Unit touch gesture									
	Hold down	C-stroke	P-stroke	Slip	Release	OOR	Predicative value (msec)	Gesture execution time	*t*	*p*-value
Drag	×1		×1		×1		652	658.9	0.26	0.79
Flick	×1			×1			177	203.14	-1.46	0.164
Rotate	×1	×1			×1		728	743.99	-0.76	0.45
Tap	×1				×1		108	105.6166	0.53	0.603
Double tap	×2				×2		216	217.0476	-0.133	0.89
Zoom-in	×1		×1		×1		652	694.273	0.22	0.82
Zoom-out	×1		×1		×1		652	646.45	0.14	0.88

Table 4. The result of validation experiment (Continued)

	User-defined gesture									
	Hold down	C-stroke	P-stroke	Slip	Release	OOR	Predicative value (msec)	Gesture execution time	*t*	*p*-value
Exclamation	×2		×1		×2	×1	921	911.625	0.33	0.749
"D"	×1	×1			×2	×1	1385	1156.857	3.25	0.014
Cancel	×2			×2	×2	×1	683	691	-0.36	0.72
Question	×2	×1			×2	×1	1057	1027.5	0.44	0.66
Confirm	×2	×1			×2	×1	1057	1019.04	0.87	0.41
Unlock	×1	×1	×1	×1			1341	1396	-1.32	0.22

p-value = 0.89; zoom-in, *p*-value = 0.82; zoom-out, *p*-value = 0.88). This means that the model predicts users' performance time of gestures well.

Whether the model can predict well the gestures not currently used by all users was validated through user-defined gestures. It was also confirmed that no statistically significant differences were found between subjects' performance time and the performance time of user-defined gestures predicted through the model (exclamation, *p*-value = 0.749; cancel, *p*-value = 0.72; question, *p*-value = 0.66; confirm, *p*-value = 0.41; unlock, *p*-value = 0.22). However, concerning "D" gesture, the subjects' gesture performance time and the performance time predicted by the model were not consistent (*p*-value = 0.014).

4. Discussion

This study proposed a unit touch gesture model for the prediction of touch gesture performance time. The newly proposed unit touch gesture model can predict not only the performance time of existing gestures, but the performance time of the gestures defined by users. Table 5 shows the error rate of the unit touch gesture model. According to this, less than 20% of error rate was revealed between the actual subjects' gesture performance time and the predicative values of the unit touch gesture model. However, the *p*-value of the gesture "D" was 0.014, and the model's predicative value and experiment-measured value were different. According to a study of Olson and Olson (1990) reporting that a model is effective, if less than 20% of error rate is shown between the predicative and measured values, the measured value can also be effectively used for the purpose of performance time prediction in the initial design stage.

Error rate was represented as percentage through dividing difference between the model value predicted with unit touch gesture and subjects' experiment value by the model value. By referring to the gesture definition of Wigdor and Wixon (2011), this study drew the error rate by applying the operator suitable for the gesture definition. For example, this study applied the performance time not to include terminating motion in Flick, although Drag and Flick are the same straight line motions. The reason why a difference between "Drag" and "Flick" exists is concerned with the existence or nonexistence of a terminating motion. Actually, the gesture performance time was drawn through the combination mode as shown in Table 4.

The limitation of the unit touch gesture model is that all individual differences caused in the gesture performance aspect could not be considered. For example, in multi touch gesture, there were the subjects using a thumb or an index finger. 18 subjects out of 20 subjects in total showed a multi touch gesture performance aspect using a thumb. Consequently, this study drew the performance time reflecting most performance aspects.

Table 5. Error rate of unit touch gesture model

Gesture	Error rate of unit touch gesture model
Drag	1.05%
Flick	14.76%
Rotate	2.19%
Tap	2.21%
Double tap	0.48%
Zoom-in	6.44%
Zoom-out	0.84%
Exclamation	1.01%
"D"	16.47%
Cancel	1.25%
Question	2.79%
Confirm	3.59%
Unlock	4.101%

(Error rate = (Model value - Experiment value)/Model value x 100)

In the model, P-stroke was defined as the unit operator on the straight line movement, and this is the operator of which performance time increases in proportion to distance. The reason why it was defined as time is that the operator can be applied in proportion to distance through the number of performance like the unit touch gesture model, as performance time according to distance proportionally increases, but it increases in a certain rate. Therefore, the unit touch gestures can be utilized, so that the number of performance can be applied accordingly, as distance increases. Figure 6 shows the analysis results of performance time according to the subjects' drag, zoom-out and zoom-in gestures.

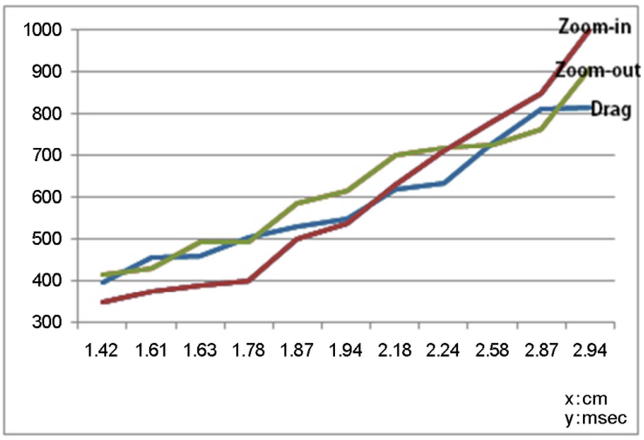

Figure 6. Performance time change analysis according to the distances of Drag, Zoom-out, Zoom-in gestures

5. Conclusion

The unit tough gesture model defined through this study makes it possible to predict the performance time of a new gesture in the initial stage of system design, and the model is meaningful in that the gestures not defined yet can be predicted in addition to the performance time of a touch screen currently used widely. Therefore, the model in this study can be usefully utilized by interface experts or in the system planning stage.

References

Anderson, J.R. and Lebiere, C., *Atomic Components of Thought*, Lawrence Erlbaum Associates, New Jersey, 1998.

Amant, R.S., Horton, T.E. and Ritter, F.E., Model-based evaluation of expert cell phone menu interaction, *ACM Transactions on Computer-Human Interaction*, 14(1), 1, 2007.

Back, J. and Myung, R., Applying CPM-GOMS to Two-handed Korean Text Entry Task on Mobile Phone, *Journal of the Ergonomics Society of Korea*, 30(2), 303-310, 2011.

Back, J., Kang, M., Lee, K., Lee, J., Kim, H., Kim, M. and Seo, J., Trend Analysis of Gesture Recognition, and NUX, *Review of Korean Society for Internet Information*, 12(4), 36-42, 2015.

Bennett, J., Butler, K. and Whiteside, J., Usability Engineering, A tutorial presented at CHI, *'89 Conference on Human Factors in Computing Systems, Austin*, TX, April 30-May 5, 1989.

Card, S.K., Moran, T.P. and Newell, A., Computer text-editing: An information-processing analysis of a routine cognitive skill, *Cognitive Psychology*, 12(1), 32-74, 1980.

Choi, M., Lee, B.G., Oh, H. and Myung, R., Extracting flick operator for predicting performance by GOMS model in small touch screen, *Journal of the Ergonomics Society of Korea*, 32(2), 179-187, 2013.

Ghasemifard, N., Shamsi, M., Kenari, A.R.R. and Ahmadi, V., A new view at usability test methods of interfaces for human computer interaction, *Global Journal of Computer Science and Technology*, 15(1), 2015.

Holleis, P., Otto, F., Hussmann, H. and Schmidt, A., "Keystroke-level model for advanced mobile phone interaction", In *Proceedings of the 25th SIGCHI Conference on Human Factors in Computing Systems*, (pp. 1505-1514), San Jose, CA, 2007.

Hong, D. and Woo, W., Recent Research Trend of Gesture-based User Interfaces, *Telecommunications Review*, 18(3), 403-413, 2008.

Irving, S., Polson, P. and Irving, J.E., "A GOMS analysis of the advanced automated cockpit", In *Proceedings of the SIGCHI '94 Conference on Human Factors in Computing Systems*, (pp. 344-350), Boston, MA, 1994.

John, B.E., *Contributions to Engineering Models of Human-Computer Interaction*, Doctoral Dissertation, Pittsburgh, 1988.

John, B.E. and Kieras, D.E., Using GOMS for User Interface Design and Evaluation: Which Technique?, *ACM Transactions on Computer-Human Interaction*, 3(4), 287-319, 1996a.

John, B.E. and Kieras, D.E., The GOMS Family of User Interface. Analysis Techniques: Comparison and Contrast, *ACM Transactionson Computer-Human Interaction*, 3(4), 320-350, 1996b.

Jokisch, M., Bartoschek, T. and Schwering, A., "Usability testing of the interaction of novices with a multi-touch table in semi public space", *In Proceedings of the 14th international conference on human-computer interaction: interaction techniques and environments*, Vol. 2, (pp. 71-80), Heidelberg, Berlin, 2011.

Kieras, D.E., A Guide to GOMS Model Usability Evaluation using NGOMSL. In Helander, M. and Landauer, T.K. (Eds), *The handbook of human-computer interaction* (2nd ed). Elsevier, New York, 733-766, 1999.

Kieras, D.E. and Meyer, D.E., An Overview of the EPIC Architecture for Cognition and Performance with Application to Human-Computer Interaction, *Human-Computer Interaction*, 12(4), 391-438, 1997.

Kieras, D.E., Model-based evaluation. In Jacko, J.A. and Sears, A. (Eds), *The human-computer interaction handbook*, L. Erlbaum Associates Inc., Mahwah, NJ, 1139-1151, 2002.

Landauer, T.K., *The trouble with computers: Usefulness, usability, and productivity*, Vol. 21., MIT press, Cambridge, 1995.

Lee, S. and Myung, R., Modified GOMS-Model for Mobile Computing. *Journal of the Society of Korea Industrial and Systems Engineering*, 32, 85-93, 2009.

Lee, Y., Jeon, Y. and Myung, R., Revised computational-GOMS model for drag activity, *Journal of the Ergonomics Society of Korea*, 30(2), 365-373, 2011.

Lewis, C., Polson, P.G., Wharton, C. and Rieman, J., "Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces.", In *Proceedings of the '90 SIGCHI conference on Human factors in computing systems*, (pp. 235-242), Washington, 1990.

Moran, T.P., Card, S.K. and Newell, A., *The psychology of human-computer interaction*, CRC Press, New York, 1983.

Morris, M.R., Wobbrock, J.O. and Wilson, A.D., "Understanding users' preferences for surface gestures", In *Proceedings of Graphics Interface 2010*, (pp. 261-268), Toronto, 2010.

Olson, J.R. and Olson, G.M., The growth of cognitive modeling in human-computer interaction since GOMS, *Human-computer interaction*, 5(2), 221-265, 1990.

Sears, A. and Jacko, J.A., Human-computer interaction: Designing for diverse users and domains (2nd ed), CRC Press, New York 2009.

Villamor, C., Willis, D. and Wroblewsk, L., Touch Gesture: Reference Guide, Lukew, http://static.lukew.com/TouchGestureGuide.pdf (retrieved June 20, 2016), 2013.

Wigdor, D. and Wixon, D., *Brave NUI world: designing natural user interfaces for touch and gesture*, Elsevier, Burlington, 2011.

Author listings

Damee Kim: dameeing@korea.ac.kr
Highest degree: BE in Department of Interactive Media, Gachon University
Areas of interest: HCI, User interface, Cognitive Modeling

Rohae Myung: rmyung@korea.ac.kr
Highest degree: Ph.D., Industrial Engineering, Texas Tech University
Position title: Professor in Dept of Industrial Management Engineering, Korea University
Areas of interest: Cognitive Engineering, Cognitive Modeling