Analyzing two hand interaction with a mobile phone to modify KLM method: Effect of hand size

SIE 511 Human-Machine Interaction

Instructor: Dr. Changxu Wu

Student Name: I-Chun Lien(23483334), Yang-Sheng Chang (23424239)

1. Abstract

Keystroke model level helps build a model of keystroke device's execution time widely used in user analysis. With considering the pointing device like PDA, mobile phone , and tablet, the prediction of the execution time is unable to apply in the different UI model. Keystroke model level is not precise to describe the circumstance of those pointing device. To improve the model of prediction, modifying the execution time model and adjusting the operators are the enhancement of KLM's modification called fingerstroke model level. For the movement of pointing, the model is based on Fitts's Law, which is a predicting execution time of human movement in ergonomics. One of the main difference between keystroke device and the touchscreen device is replacing the operating methods from typing the keyboard and clicking the mouse to typing the virtual keyboard and moving hand to the target. Fitts's Law considers the distance of the movement and the target area. However, the effect of the hand size is not consider in the model which may be an effective factor of predicting the time. The original fingerstroke level model is constructed on a PDA device. The effect of the hand size is more obvious in the pure touchscreen device like mobile and tablet than PDA. The result is that we constructed a new model to predict the execution time of touchscreen device effecting by hand size.

Keywords: Keystroke-level model, Fingerstroke-level model, Execution time, User analysis, User interface, Pointing device

2. Introduction

2.1 Introduction of KLM

KLM is short for Keystroke level model applying in keystroke model to predict users' execution time. KLM helps build the model of how long it takes a user to accomplish a given task. It disassembles the process of users' execution into several operators. The operators include keystroking(K), pointing(P), drawing(D), homing(H), response time(R), and mental preparation(M). Keystroking operator represents the action of clicking the target by mouse or typing the keyboard in fixed position. Pointing operator represents the movement from target to target using the mouse. Drawing operator represents the action of dragging the target like moving the folder on the desktop to the trash. Homing operator represents the movement between the keyboard and the mouse when the users place their hands to the ideal position to start executing the tasks. Response time means that the response time of computer system or the waiting time of the internet transmitting the data from the server. Mental preparation is a psychological operator like thinking and human response time which is dynamic in a large range.

Table1. Step termninology

Operator		Description	Time (s)
K	Press a Key or button (including shift, control), time varies with user skill	Best typist (135 wpm)	0.08
		Good typist (90 wpm)	0.12
		Avg skilled typist (40 wpm)	0.20
		Avg no-secretary typist (40 wpm)	0.28 or...
		Typing random letters	0.50
		Typing complex codes	0.75
		Worst typist (unfamiliar with keyboard)	1.20
P	Point with a mouse	Point with a mouse (range is 0.8 to 1.6 sec, not including button press)	1.10
H	Home to/from keyboard or other device		0.40
D	Draw nd straight lines of total length ld		9 nd + 0.16 ld
M	Mentally prepare		1.35
S	Scan	e.g., find coordinates of spreadsheet, not in original KLM	2.29

R(t)	Response by system	Varies with command, including wait if required	t

For example, the task is to rename the folder “old_name” to “new”.

Solution of the task: Preparation(M)➔moving the mouse to the target folder(P)➔selecting the target(K)➔right click the mouse(K)➔moving down to the “Rename” option(P)➔clicking on the option(K)➔moving hand back to the keyboard(H)➔pressing DELETE(K)➔thinking the new name(M)➔entering character “NEW”(3K)➔pressing ENTER(K)

Execution time: $K \times 8 + P \times 1 + H \times 1 + M \times 2 = 5.8(s)$

2.2 Disadvantage of KLM

The original KML has some disadvantages. One of them is that it is precise to predict the execution time on the pointer device because the pointer device can fast and accurately move to the target place. However, for the touch screen, it is not suitable to use original KLM to predict the execution time. The reason for this problem is that our finger and hand size is different from pointing device. Also, everyone’s hand and finger size vary. Different finger size, hand size, and target size will affect the result of predicting execution time. When using different sizes of touch screen, the users require different models to predict the execution time. For instance, the users could use one hand to handle the touch screen such as iphone or smaller size of touch screen device such as smart watch. When people use the mobile phone, they could tell that some region in the touch screen is more comfortable than other region in the touch screen. The other limitation of original KLM concludes several small number of operators. In the pointing device, it can be easily to categorized to five operators such as pointing, keystroke, home, draw and mental. Multiple gestures on the touch screen can provide useful functions that can make the customers easily use and the execution time is shorter. Therefore, it should have more operators to describe the different gesture and accurately predict the execution time.

2.2 Applying at touch screen (Fingerstroke-level model)

The traditional KLM operators is based on the execution time on keystroke. Due to the development of digital touch devices, fingerstorke-level model was emphasized four input operators. In 2014, Rice et al. mentioned new additional operators to analyze human task performance on touchscreen. Lee et al.(2015) first defined four fingerstroke operators, which are Tapping (T), Pointing (P), Dragging (D), and Flicking (F).

The original keystroke-level model is composed of K(pressing the keyboard or mouse), P(pointing to the target), D(moving the mouse from original position to next target), H(moving from keyboard to mouse or from mouse to keyboard), R(system response time), M(mental action)(Card et al. 1980). The R and M operators are uncontrollable parameters related to different people and scenario. Besides, H operator is ignored in FLM because touchscreen integrates mouse and keyboard into a single screen. The other operators correspond to the newly defined operators: K, the input operator of pressing keyboard or mouse, in KLM is related to T(tapping) in FLM, P(pointing) is the similar operator in KLM and FLM that represents lifting the finger and moving to the next target, D(drawing) in KLM is related to D(dragging the object from one place to another). And a new operator is called F(flicking)(Lee et al. 2015)

Table2. Comparison between the keystroke-level model(Card et al. 1980) and fingerstroke-level model(Lee et al. 2015)[3]

Keystroke operator	Time(s)	Fingerstroke operator	The average time(s) of 4,8,11mm target size
K(ketstroking)	0.2	T(tapping)	0.31
P(pointing)	1.1	P(pointing)	0.43
D(drawing)	0.9n+0.16d	D(dragging)	0.17
		F(flicking)	Left to right: 0.11 Right to left: 0.12
H(homing)	0.4		
R(response time)	Variable	R(response time)	Variable
M(mental preparation)	1.35	M(mental preparation)	1.35

2.3 Fitts's Law

Fitts's law (MacKenzie, 1989, 1992) is used to predict the movement time for a given task on a pointing device. . (Mackenzie, 1989; Wright & Lee, 2013) shows lots of detail about Fitts's law to measure the execution time. The original Fitts's law is equation (1). It expresses Movement

Time(MT), which is the time to finish the task. For the equation (1), A is in term of amplitude or the distance of the move. The W is the width of the region of the move.

$$\mathrm{MT} = \mathrm{a} + \mathrm{b} * \log_2\left(\frac{2A}{W}\right) \qquad (1)$$

Index of difficulty (ID) of a movement task is expressed as

$$\mathrm{ID} = \log_2(\frac{2A}{W}) \qquad (2)$$

The fitt's law can also expressed as

$$\mathrm{MT} = \mathrm{a} + \mathrm{b} * \mathrm{ID} \qquad (3)$$

The equation (1) and (3) are linear. Constant a is intercept and constant b is slope. For achieving higher accuracy, it can be expressed as equation (4). Equation (4) is more accurate because it separates A and W and offer extra degree of fine tuning for the prediction model by using three empirically determined constants.

$$\mathrm{MT} = \mathrm{a} + \mathrm{b1} * \log_2 2A - \mathrm{b} * \log_2 W \qquad (4)$$

The index of difficulty (ID) of movement task from the Shannon's theorem 17 to derive the equation (5). A is the distance between targets; W is the target size; MT is the execution time.

$$\mathrm{MT} = \mathrm{a} + \mathrm{b} * \log_2(\frac{A}{W} + 1) \qquad (5)$$

Fitts's law equation could reconstruct the model of touchscreen devices to certain constrained situation. The formulation used heavily in current study is $T = a + blog_2\left(\frac{A}{W} + 1\right)$, which T is the predicted movement time, A is the distance of movement, and W is the target size, a and b terms are derived regression coefficients specific to certain task condition.

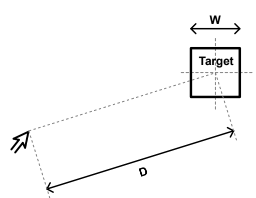

Figure1. Fitts's Law: Draft of target size W and distance to target D [2]

2.4 Possible Variables

Park et al. discussed the effects of touch key size and touch location for one-hand thumb to interact with a smartphone. Hsinfu Huang(2017) concluded some factors affecting interface accessibility of smartphone. By analyzing multiple reasons causing the execution time, we think that three parts of factors may affect the fingerstroke-level model.

2.4.1 Effect of Target Size

The design of the target size affects users' convenience and accuracy and then indirectly influences the execution time. In 2006, Parhi et al. studied for target size for one-hand thumb. They analyzed five kinds of target sizes and discovered that the smaller size of the target required longer transition time and total execution time. When the target size grew, the speed of execution the tasks would have progression.

Park et al.(2009) proposed the effect of target size was related to the pressing convenience and number of errors. The smaller target size would consume more time to complete the task. Kim et al.(2018) and Quezada et al.(2018) found out target size affected the execution time and did further analysis. The target size influences both precision and the execution time when performing to achieve the given tasks.

2.4.2 Effect of Target location

In FLM, D(dragging), P(pointing), and F(flicking) are operators moving the finger to the different position. The touch key location affects the total execution time. Parhi et al(2006) separated the touch screen into 9 regions and observed the result of the usability of each region. The users would have preference in certain regions and it would affect the performance.

The movement of execution time can be estimated by Fitts's Law equation. The Fitts's Law can be applied in process of moving finger in pointing device.

2.4.3 Effect of Hand length

Another factor may influence predictive time is users' hand length such as palm size and fingers' size. Wobbrock et al.(2008) discussed the performance of hand postures. When writing the letter, the users spent more time using thumb nail than using index finger and wasted more time using single hand than both hands. The conducting time is different when the users using the index finger or thumb. Besides, users applying single hand with thumb nail interaction with mobile phone, the length of thumb affected task completion time.(Kim et al. 2018)

3. Research paper’s relation

3.1 Keystroke-Level model

Keystroke level model was proposed by Stuart K. Card, Thomas P. Moran and Allen Newell in 1980 and widely used in human-computer interaction observation. They mentioned that the execution time could be separated into several processes which were called operators.

In the following development of digital devices, the researchers use the model of KLM to estimate the execution time and apply in the designation of user interface. Kieras (2001) used standard keystroke model’s operators to analyze execution time. For example, KLM is applied by designing user interface on the middle size touch screen by Abdulin (2011). He analyzed three different graphical user interfaces for choosing the best arrangement for the panel of the integrated control systems.

Kieras (2001) discussed the mental operator in KLM existing a lot of judgement because mental operator is different from the physical operators (tapping, pointing, drawing, homing). The reason why the mental operator is inconsistent in each circumstance due to several detailed possibilities, users’ experience, and so on. Otherwise, experienced users can overlap mental operator with physical operator.

3.2 Fingerstroke-Level model

Park et al. discussed the effects of touch key size and touch location to interact with a smartphone with personal digital assistance(PDA) which is different from smartphone nowadays.

4. Experiment design

In observation of solving the limitation, the researchers try to develop a revised KLM to predict the execution time. Several papers talked about how to accurately predict the execution time. Paper shows lots of details to measure the data and use the nonlinear regression to know the constant by the equation based on Fitts’s Law.

$$\mathrm{T} = \mathrm{a} + \mathrm{b} * \log_2(\frac{A}{W} + 1)$$

The equation is to predict the movement of human-machine interaction (A is the distance between targets, W is the target size, T is the execution time, a and b are constants). Each movement has different the a and b. For instance, right hand people will fast and comfortable move from left to right.

For the main propose, the experiment look forward to the influence of hand size affecting the execution time. At first, we will tell the procedure of the experiments to the participants and they need to practice to be familiar with the experiments. Second, the participants will do the tasks steps by steps. In the meantime, the data will be collected through the android on-device developer option. Then, through MATLAB to classify the data and analyze, we will conclude the modification of the model. The result will show at next section. Besides, the participants will need to do another test to verify the result.

4.1 Condition

The experiment conducts at an android system's mobile phone which has screen size 12.1mm* 6.9mm. The target size that is 8mm and categorized the different length of thumb and other fingers' length. Therefore, the result will conclude the relation between these elements and execution time.

4.2 Experiments

Due to uncertainty of the mental operator which was mentioned in Kieras (2001) and inconsistency of system response time, we discussed the four physical operators relatively more predictive. The design of experiment separated into four parts for each physical operators in fingerstroke-level model: Tapping, Pointing, Dragging, Flicking.

4.2.1 Experiments of Tapping

For tapping, like the figure2 below, the participants tapped each target for three times clockwise with each thumb. The target was close enough to simulate tapping the target in the same position. Both hands conducted three times.

4.2.1 Experiments of Pointing

For pointing, like the figure3 below, the participants implemented the tasks from left to right and from right to left. The participants required to press the first target and lift hand to the next target. Both hands for three times.

4.2.1 Experiments of Dragging

For dragging, like the figure3 below, the participants implemented the tasks from left to right and from right to left. The participants required to press the first target and drag touch the next target. Both hands for three times.

4.2.1 Experiments of Flicking

For flicking, using the main menu with 5 pages, the participants flicked the pages up for 5 times, press the home button and down for 5 times. Both hands for three times.

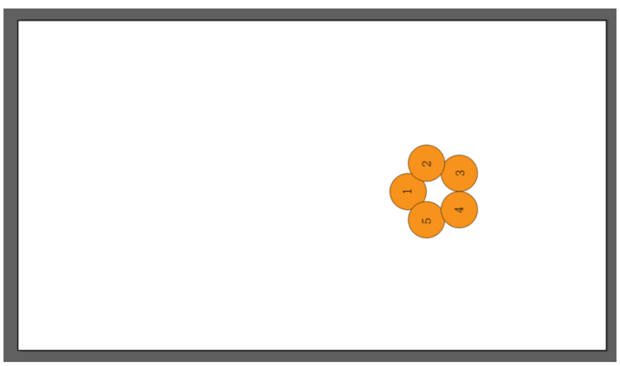

Figure2. Tapping test design

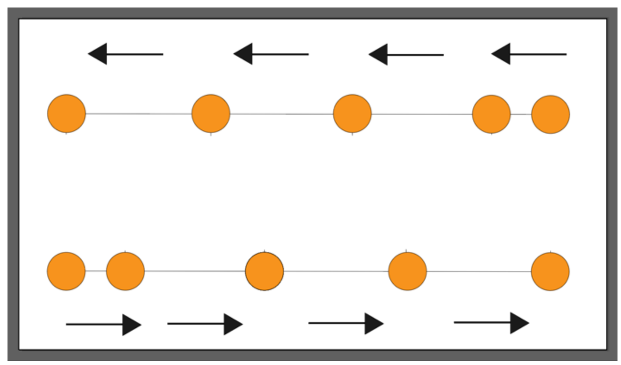

Figure3. Pointing, Dragging test design

4.3 Collecting the data

To precisely capture the data of the position that the users touch at, the data collected from the debugging report in android configure system behaviors. First, turn on the developer options in android phone (in Setting App). Second, enable the option of "debugging over the USB". Third, capture the information of the input from the mobile phone. Forth, use android debug bridge(adb), which is a kind of command line tool to intercept the debugging report.

Type the command to show all the input events,

> cd ~/to_the_adb_shell_folder

>./adb shell getevent -lt

The information could be implemented in analysis of the touch screen interaction is dev/input/event7.

4.4 Testing experiment

To verify, the experiment of testing was designed to compare with the modified FLM equation and the standard prediction.

The given task: Open the dictionary, search keyword “mac”, and copy the meaning of the keyword like figure3.

Each step of the task	Operators
1. Click the Dictionary APP	T
2. Tap the textbox of searching	T
3. Pointing to the virtual keyboard	P
4. Type the character “MAC”	3T
5. Type enter	T
6. Tap twice to highlight the words	2T
7. Drag to highlight ten sentence	D
8. Flick up and flick down	2F

Figure3. The testing tasks

Figure4. Testing experiment

5. Analyze Data

Before analyzing the data, it is necessary to preprocess the raw data. There are some steps to get the result of data. First step, it is better to convert the txt file to excel file because it is much easier for people to realize the original data. Second step is to find the data that can let the analyzing use to know when the participant touch screen and leave the screen. There is an item called BTN_TOUCH which can show when the participant touch screen and leave the screen. Third step is to use BTN_TOUCH to extract the data to know the time. Fourth step is to find the average of each operator. Fifth step is to develop the model to fit the data. It is great for analyst to use matlab to develop the model. The model's equation is

$$\mathrm{T} = \mathrm{a} + \mathrm{b} * \log_2\left(\frac{A}{W} + 1\right) + c * finger's\ length$$

It is a linear model. In the processing of analyzing, there is an interesting thing that it is no big difference between linear model and nonlinear model. Therefore, using linear model is quite great for the prediction. Sixth step, using the data from previous preprocessing is enough to describe the relationship between time and finger. Seventh Step, using the MSE data to know that which factor to develop the general equation. For example, in the tapping operator, the little finger's MSE is smallest. Therefore, the tapping operator's equation is developed from little finger model to get the variable.

5.1 Participant's finger data

In the Talbe1, it shows the participants' finger information. The special thing is that it shows the average of the palm size because it might affect the result of the prediction. This project is to show which elements will influence the prediction of four operators (tapping, flicking, pointing and dragging). The total participants are four people (2 males and 2 females). It will reduce the gender element's importance. In the Table2, it shows the finger's deviation. The deviation is quite large because this experiment only has four participants. In the detail, the index finger and palm's deviation are especially large. It can tell that human's palm can be huge different.

Table1.

The average of each finger size

	Thumb	Index	Middle	Ring	Little	Palm
Left	7.175	7.825	8.125	7.75	6.025	10.9
Right	7.05	7.85	8.075	7.375	6	10.7
Both	7.1125	7.8375	8.1	7.5625	6.0125	10.8

unit: centimeter

Table2.

The finger's deviation

	Thumb	Index	Middle	Ring	Small	Palm
Left	0.4573	0.8732	0.5377	0.238	0.4991	0.6164
Right	0.7325	1.0969	0.25	0.236	0.4966	0.9055
Both	0.5693	0.9179	0.3891	0.297	0.4611	0.7250

unit: centimeter

5.2 Each operator's data

In this part, it will show each operator's analyzing description. First one is tapping operator, second one is flicking operator, third one is dragging operator, and the last one is pointing operator.

5.2.1 Tapping operator

In the tapping part,the Table3 is derived from the linear model and to predict the execution time.. It can tell the right hand is faster than left hand. It might be that the participants are right-handed. It will be more comfortable and faster to type the word with right hand.

The average of the test tapping is 0.078185714 seconds. About the kind of great difference between prediction execution time and testing time. The possible reason is that when the participants type the word or touch screen in the experiment, they will not touch and leave as soon as posible becuase people will not tap quickly in the unhurry time. In the tesing stage, the participant will like to quickly finish the task we give. And also, the participant tap in concecutively. It will be faster to tap in the testing stage.

About how to evaluate the relationship between finger and the exectuion time. The mean square error(MSE) can tell the relationship between finger and exectuion time. When the MSE is really small, it means that the factor relatively affect the prediction model. Therefore, the ring, small and palm is highly relate to the tapping operator. It means that these factors can help to enhance the model to predict.

Table3.

The prediction time of tapping

	Thumb	Index	Middle	Ring	Little	Palm
Left	0.1226	0.1228	0.1229	0.1228	0.1234	0.1239
Right	0.1081	0.1086	0.1088	0.109	0.1088	0.1085
Both	0.123	0.1157	0.1159	0.1157	0.1161	0.1177

unit: seconds

Table4.

The mean square error of tapping predict model

	Thumb	Index	Middle	Ring	Little	Palm
Left	0.001281	0.001218	0.001158	0.000709	0.000589	0.00044
Right	0.001074	0.000817	0.000426	0.000928	0.000563	0.000296
Both	0.001065	0.00047	0.000303	0.000519	0.000001	0.00095

unit: seconds

The tapping predicted execution time's equation will be

$$\mathrm{T} = 1.5 + 3.2 * \log_2\left(\frac{A}{W} + 1\right) - 0.007 * little\ finger's\ length$$

5.2.2 Dragging operator

In this part, the dragging operator's execution time is greater than tapping. There maybe two reasons. First is that the distance between two targets is 6 centimeters in the experiment stage and testing stage. Second, there is friction between hand and screen. If the participants want to drag the icon or select words, they will touch the screen in the execution period. Therefore, it will slow down the movement. For the Table5, it can tell that the index finger is more effective to the dragging operator because the MSE is only 0.0005583.

The average of the test dragging time is 0.992208333 second. Compare between experiment stage and testing stage, the prediction execution time is not very accurate. The reasonable reason is that, the participants repeated to drag the icon in the experiment. Therefore, the speed is faster. This reason maybe opposite to the tapping. Also, in the test stage, the participant will not be familiar with our task in the beginning. Therefore, the actual execution time will be greater than predicted time.

Table5.

The prediction time of dragging

	Thumb	Index	Middle	Ring	Little	Palm
Left	0.8249	0.8238	0.8236	0.8258	0.8232	0.8234
Right	0.7059	0.7046	0.7029	0.7033	0.6986	0.7036
Both	0.7543	0.7645	0.752	0.7538	0.7506	0.7537

unit: seconds

Table6.

The mean square error of dragging predict model

	Thumb	Index	Middle	Ring	Small	Palm
Left	0.0043888	0.0043888	0.0004623	0.0092928	0.0018379	0.0024662
Right	0.0367809	0.0208165	0.065979	0.0558595	0.0368099	0.0509349
Both	0.0148748	0.0005583	0.0199396	0.0237321	0.0153688	0.0143474

unit: seconds

The dragging predicted execution time's equation will be

$$\mathrm{T} = -1.8 + 0.76 * \log_2\left(\frac{A}{W} + 1\right) + 0.02 * index\ finger's\ length$$

5.2.3 Pointing operator

The average of the test pointing is 0.456166667 second. The predicted time and actual time is close and the MSE is small. The pointing execution time is different from dragging execution time even the distance between two targets is the same. The reason is that there is no fraction between finger and screen when the participants pointing on the screen. In the Table7 and Table8, these can show that the difference execution time between right hand and left hand is not as big as dragging. It is interesting phenomenon. It probably tells that people use dragging operator with dominant hand, it will be faster. However, it does not happen in pointing operator. The reason mightbe when people point on the screen, they do not need to spend too much effort. It can spend less time.

Table7.

The prediction time of pointing

	Thumb	Index	Middle	Ring	Little	Palm
Left	0.4652	0.4628	0.4624	0.4668	0.4668	0.4669
Right	0.4118	0.4116	0.4098	0.4072	0.4074	0.4076
Both	0.4386	0.4823	0.4359	0.4384	0.4345	0.4397

unit: seconds

Table8.

The mean square error of pointing predict model

	Thumb	Index	Middle	Ring	Little	Palm
Left	0.0367915	0.0268421	0.0255611	0.0155704	0.0284872	0.0235584
Right	0.0356431	0.0355647	0.0273363	0.0132637	0.0001425	0.007816
Both	0.0271004	0.0223332	0.0188144	0.0309481	0.0090779	0.0306242

unit: seconds

The pointing predicted execution time's equation will be

$$\mathrm{T} = -2.33 + 0.59 * \log_2\left(\frac{A}{W} + 1\right) + 0.15 * little\ finger's\ length$$

5.2.4 Flicking operator

The average of the test flicking is 0.1523625 second. For the flicking part, this operator is to flick the touch screen quickly. It is much accurate to predict the flicking execution time because it does not have target. It will let the users flick quit the same time. Therefore, no matter people use the right hand or left hand to flick on the screen, the execution time might be quite the same.

Table7.

The prediction time of flicking

	Thumb	Index	Middle	Ring	Little	Palm
Left	0.1569	0.1582	0.1582	0.1565	0.1589	0.1684
Right	0.1391	0.1393	0.1393	0.1402	0.1397	0.1399
Both	0.1482	0.1486	0.1486	0.1482	0.1495	0.1489

unit: seconds

Table8.

The mean square error of flicking predict model

	Thumb	Index	Middle	Ring	Little	Palm
Left	0.0071363	0.0037882	0.0037882	0.0064628	0.0019143	0.0052179
Right	0.001149	0.000889	0.000889	0.0005098	0.0001561	0.0000077
Both	0.0033851	0.0023948	0.0023948	0.0034392	0.0002978	0.0034774

unit: seconds

The flicking predicted execution time's equation will be

$$\mathrm{T} = -1.47 + 0.72 * \log_2\left(\frac{A}{W} + 1\right) - 0.06 * little\ finger's\ length$$

Table9.

Operator	Equation
Tapping	$\mathrm{T} = 1.5 + 3.2 * \log_2\left(\frac{A}{W} + 1\right) - 0.007 * little\ finger\ length$
Dragging	$\mathrm{T} = -1.8 + 0.76 * \log_2\left(\frac{A}{W} + 1\right) + 0.02 * index\ finger\ length$

Pointing	$\mathrm{T} = -2.33 + 0.59 * \log_2\left(\frac{A}{W} + 1\right) + 0.15 * little\ finger\ length$
Flicking	$\mathrm{T} = -1.47 + 0.72 * \log_2\left(\frac{A}{W} + 1\right) - 0.06 * little\ finger\ length$

5.2.4 Evaluation

In this part, we will use the testing data to evaluate our model. In the previous, we mention that we will see each finger's MSE to the model. Therefore, we decide to choose the smallest MSE value and get the a,b and c variable. In the Table9, it is easier to see that our model's prediction time is close to the actual time. It can tell that our model is quite predictive in this task.

Table10.

Comparison of the each model to actual time

	KLM	Our Model	Actual Time
T	0.2	0.1161	0.067992
T	0.2	0.1161	0.08835
P	1.1	0.4345	0.456167
3T	0.2*3	0.1161*3	0.270883
T	0.2	0.1161	0.061558
2T	0.2*2	0.1161*2	0.126867
D	1.67	0.7645	0.992208
2F	0.2*2	0.1495*2	0.304725
Total Time	4.77	2.5429	2.36875
Divided by Actual Time	2.013720317	1.073519789	1

unit: seconds

6. Conclusion

There are two drawbacks of this project. One may be that in reality the users will not use the same size of screen. Therefore, the distance varies, and the design of touch screen's frame is different. Another one is that the number of participants are not enough. The data will not be consistent to every hand size. As result, the time estimates from our project may not be robust enough. Even we have great prediction result. It does not mean that the result of data can accurately predict to any size of touch screen. Also, when people repeat to do the same movement, the execution time will be smaller. Therefore, the practicality of this model we propose in this project does not guarantee to any size and any brand of touchscreen, but a great frame for applying time estimation to other design problems. Wright and Lee (2013) have pointed out this in detail. They proved that repetitive tasks are faster than discrete tasks, but they claim that this fact does not make the interpretation of Fitts' law on actual problems insignificant.

7. Reference

Card, S.K., Moran, T.P., and Newell, A. (1980). The Keystroke-Level Model for User Performance Time with Interactive Systems, Communications of the ACM, July, 23(7), 396-410.

[1]Heejin Jeong, Dongfang Lan, and Guofa Li. (2013). GOMS Model for Interface Design and Evalution. https://sites.google.com/site/436goms/gomsmodel/klm

Kieras, D. (2001). Using the keystroke-level model to estimate execution times. University of Michigan, 555.

[2] Foobar628 [CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0)], from Wikimedia Commons

Gustafsson, E., Coenen, P., Campbell, A., & Straker, L. (2018). Texting with touchscreen and keypad phones - A comparison of thumb kinematics, upper limb muscle activity, exertion, discomfort, and performance. *Applied Ergonomics, 70*, 232-239.

Huang, H. (2018). Blind users' expectations of touch interfaces: factors affecting interface accessibility of touchscreen-based smartphones for people with moderate visual impairment. *Universal Access in the Information Society*, *17*(2), 291-304.

Kim, H. K., Park, J., Park, K., & Choe, M. (2018). Analyzing thumb interaction on mobile touchpad devices. *International Journal of Industrial Ergonomics, 67*, 201-209.

[3]Lee, A., Song, K., Ryu, H. B., Kim, J., & Kwon, G. (2015). Fingerstroke time estimates for touchscreen-based mobile gaming interaction. *Human Movement Science, 44*, 211-224.

Mackenzie, I. S. (1989). A NOTE ON THE INFORMATION-THEORETIC BASIS FOR FITTS LAW. *Journal of Motor Behavior*, *21*(3), 323-330.

Park, Y. S., & Han, S. H. (2010). Touch key design for one-handed thumb interaction with a mobile phone: Effects of touch key size and touch key location. *International Journal of Industrial Ergonomics, 40*(1), 68-76.

Soukoreff, R. W., & MacKenzie, I. S. (2004). Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI. *International Journal of Human-Computer Studies, 61*(6), 751-789.

Shiroq Al-Megren. (2018). A Predictive Fingerstroke-Level Model for Smartwatch Interaction

Wright, C. E., & Lee, F. (2013). Issues Related to HCI Application of Fitts's Law. *Human-Computer Interaction*, *28*(6), 548-578.