International Journal of Industrial Ergonomics 65 (2018) 122–138

Contents lists available at ScienceDirect

International Journal of Industrial Ergonomics

journal homepage: www.elsevier.com/locate/ergon

Predicting user performance time for hand gesture interfaces

Orlando Erazo [a], [b], [*], José A. Pino [a]

[a] Department of Computer Science, Universidad de Chile, Beauchef 851, Santiago, Chile
[b] Universidad Técnica Estatal de Quevedo, Av. Quito km. 1 1/2, Quevedo, Ecuador

ARTICLE INFO

Article history:
Received 31 May 2016
Received in revised form
9 June 2017
Accepted 17 July 2017
Available online 29 July 2017

Keywords:
Touchless hand gestures
Mid-air gestures
Natural user interfaces
Predictive evaluation
User models

ABSTRACT

User interfaces based on touchless hand gestures have advantages over conventional user interfaces in a variety of scenarios. However, they still have challenging problems to be researched, such as the design and evaluation of them in order to obtain satisfactory results. The classical approach of involving users to choose gestures or analyze interface designs needs to be complemented with predictive evaluations for cases in which those user-based methods are inapplicable or expensive to do. Thus quantitative user models are needed to perform those evaluations. THGLM is a model based on KLM and gesture units, but its first formulation needs to be improved. This paper completes the model by analyzing its performance in several user studies. In particular, we found out that THGLM forecasts performance time in doing tasks on UIs based on touchless hand gestures (THG) in an acceptable way (prediction error = 12%, $R^2 > 0.9$). The paper also reports a study concerning the model utility to analyze and compare interface designs. Moreover, the model utility was confirmed by independent designers who were invited to participate in a study. Finally, the initial model was extended by introducing several new operators. As a conclusion, the present model has some intrinsic limitations which are discussed, but the results confirm the general hypothesis that it can be used to analyze UIs based on touchless hand gestures.

Relevance to industry: THGLM should become a useful tool for UI designers to perform usability assessments, improve interface designs, and develop good software applications using THG. This is especially useful in situations where it is difficult to conduct tests with users or as a preliminary step in the process of developing software.

1. Introduction

Current availability of both new body-tracking devices and high-resolution displays has contributed significantly to develop applications that go beyond only entertainment. Inexpensive devices like Kinect, Intel RealSense, or Leap Motion can capture gestures people make with their hands without haptical contact. Therefore, instead of a traditional user interface (UI) based on mice and keyboards we could use an interface based on touchless hand gestures (THG) captured and interpreted by the computer system. Interfaces of this type may be considered as a new class of Natural User Interfaces (NUI) (Wigdor and Wixon, 2011; Webb and Ashley, 2012), and can be advantageous (de la Barré et al., 2009; Walter et al., 2013) in various scenarios. A few examples are sterile environments (e.g., operating rooms (Gallo et al., 2011)), public places where it is not possible or advisable to touch a display (Hinrichs et al., 2013), and classrooms to enhance the quality of education (Jagodziński and Wolski, 2015). Despite interfaces of this type may have great advantages, there are still challenges to address (Norman, 2010). One of these challenges is the design and evaluation of interfaces based on THG in order to develop successful software products; this is the subject of this article.

Designers typically involve users to design and evaluate UIs based on THG. Some of the currently available methods allow choosing gesture sets (Wobbrock et al., 2009; Vatavu, 2012; Nielsen et al., 2004) and understanding interactions with systems (Barclay et al., 2011; Hincapié-Ramos et al., 2014) by considering features like user preferences, memory, or fatigue. These methods allow quantitative evaluation of interfaces based on THG, but this approach requires dealing with the logistic difficulties of doing tests with real users, regarding planning, timing, laboratory setup, recruiting subjects, and conducting experiments. Consequently, a reasonable assumption is that UI designers may find value by adopting predictive evaluations instead of recruiting users,

[*] Corresponding author. Department of Computer Science, Universidad de Chile, Beauchef 851, Santiago, Chile.
E-mail address: oerazo@uteq.edu.ec (O. Erazo).

http://dx.doi.org/10.1016/j.ergon.2017.07.010
0169-8141/© 2017 Elsevier B.V. All rights reserved.

especially at early design stages (Erazo et al., 2015) due to the cost of collecting and analyzing data (MacKenzie, 2003; Kieras, 2003).

Given this problem with user testing, designers can use predictive evaluation instead of assessing interfaces by quantifying human performance. Simulation of the biomechanics of human motion (Nunes et al., 2015) is a method which is gaining attention for measuring performance of UI (Bachynskyi et al., 2014). It allows obtaining rich descriptions of users' movements with low cost (Bachynskyi et al., 2014). Moreover, biomechanical simulation has been made more accessible by clustering user muscle activations in interactive tasks (Bachynskyi et al., 2015). These clusters can help designers to estimate muscle loads and user performance in pointing with the arm (Bachynskyi et al., 2015). Another approach that may allow studying more than pointing tasks is the use of predictive models (MacKenzie, 2013) to quantify human performance particularly in terms of time (the period a user takes to accomplish a set of tasks (Card et al., 1980)), which is the focus of this work.

Model-based evaluation has been widely used to analyze interaction problems in HCI especially due to its advantages such as analyzing interface designs and making changes without implementing a real system (MacKenzie, 2003, 2013; Kieras, 2003). Nevertheless, previous models are insufficient to evaluate UIs based on THG due to any of the following causes: they were formulated for other interaction styles (e.g (Card et al., 1980; Cao and Zhai, 2007; Isokoski, 2001)); the extended versions of these models are not applicable to THG (e.g (Holleis et al., 2007; Luo and John, 2005; Lee et al., 2015)), or the feasibility of applying them has not been verified yet (as in the case of (Card et al., 1980; Cao and Zhai, 2007; Isokoski, 2001)); they are constrained to certain type of tasks (e.g. the main use of Fitts' Law (Fitts, 1954) is to analyze tasks of pointing and selecting in the air using a hand (Schwaller and Lalanne, 2013; Pino et al., 2013; Polacek et al., 2012; Zeng et al., 2012) and to compare devices such as Kinect and Wii (Sambrooks and Wilkinson, 2013; Pino et al., 2013; Polacek et al., 2012)). Therefore, new models to evaluate UIs based on THG are necessary. To tackle this problem, some authors have adapted previous models for drawing gestures (Erazo et al., 2015) and derived new ones for optimizing gesture sets based on multi-finger gestures (Sridhar et al., 2015). However, taking into account we are not considering fingers as a first step, Touchless Hand Gesture Level Model (THGLM) (Erazo and Pino, 2015) is an alternative. THGLM is a predictive model based on the assumptions of KLM (Keystroke-Level Model) (Card et al., 1980)—which is a well-known, well validated and relatively easy to use model. THGLM allows forecasting the time to execute a task given a method (expressed using gesture-units (McNeill, 1992; Kendon, 2004) and THG-level actions as illustrated in Fig. 1) and computed using the corresponding formulas. The current state of THGLM is that of a promising model but with incomplete results. Its authors noted that further validation was needed and other operators should be included in order to complete the model (Erazo and Pino, 2015).

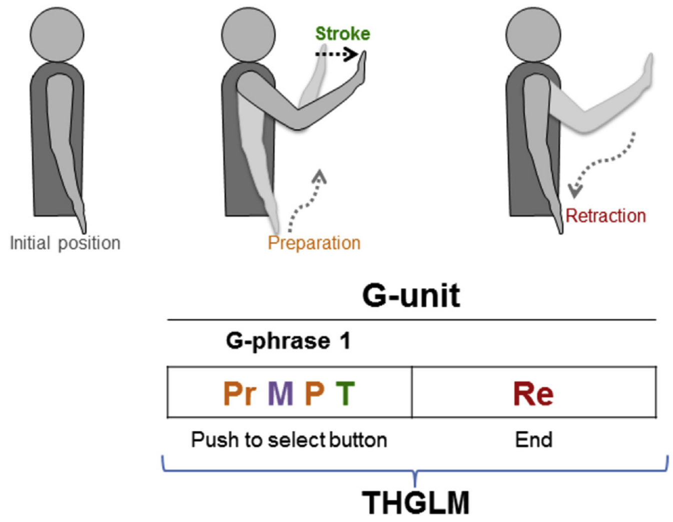


Fig. 1. Example of using THGLM (Erazo and Pino, 2015) which is a model based on KLM (Card et al., 1980) and gesture units (McNeill, 1992; Kendon, 2004) to analyze NUIs based on gestures. The task consists of selecting a button performing a push gesture, departing from a resting position and returning to it. The upper part illustrates the task execution, and the lower part shows the task modeled using THGLM.

Given this landscape, the generic goal of this paper is to complete the initial THGLM proposal (Erazo and Pino, 2015) and study its performance. Our basic research hypothesis to reach this goal is that the final model is a practical tool to help designers in the analysis and design of UIs based on THG. The verification of this general hypothesis relies on the validity of three specific ones: (*H1*) THGLM predicts performance time with acceptable quality, i.e., the values of the used metrics are consistent with those ones reported in the field for similar models; (*H2*) the model allows analyzing UI designs and comparing two or more design options; (*H3*) model predictions remain stable when they are computed by independent designers. If these hypotheses are validated, then THGLM should allow designers of UIs based on THG to predict the performance time required to complete tasks without users' participation, and next, use that value as a metric to assess a user interface. This approach is especially useful in interface evaluations where it is difficult to conduct tests with users or as a preliminary step. Therefore, we expect the model becomes a useful tool for software designers to carry out usability assessments, improve interface designs, and develop better software applications using interfaces based on THG. The main contribution of the paper is then to present a valid and usable model for predicting execution time of hand gestures by adult novice users, provided the hypotheses are confirmed.

The article starts providing the background (section 2) and summarizing the model (section 3). We give further details about THGLM in section 4, especially those related to the use of the mental operator. Then, the model validity is studied in further detail to confirm it makes good predictions (section 5). The use of the model as a tool to analyze interface designs is also included as part of the validation. Section 6 studies the stability of THGLM predictions when independent designers use the model. Next, several operators to be incorporated in the model with their estimated values are described in section 7; also, other operators that may be included in the future are suggested. The article ends with a discussion and the conclusions.

2. Background

2.1. Model-based evaluation

One way to support design and evaluation of UIs is to use model-based evaluation (MacKenzie, 2003, 2013; Kieras, 2003). It is a valuable supplement to conventional usability evaluation that is especially useful for designing, evaluating, or providing a basis to understand interfaces (MacKenzie, 2003), especially at early design stages, before starting to develop the real UI or testing with humans. Model-based evaluation implies using models of how humans would interact with applications. Models can be either descriptive or predictive depending on detail and complexity (MacKenzie, 2013). Descriptive models give designers a framework to describe and reflect on problems qualitatively, whereas predictive models are based on mathematical expressions for predicting user performance (MacKenzie, 2003).

Performance can be measured using various dimensions (Card et al., 1980), but we are just interested in time. Performance time is the required period to accomplish a set of tasks using a system (Card et al., 1980).

Some notable models have been proposed in the research literature to design and assess UIs by producing numerical predictions of users' performance time. One of the most influential contributions on modeling is the Card, Moran and Newell's book (Card et al., 1983), in which they described several models that have been widely used afterwards; they also proposed a theoretical basis to build new models. MHP (Model Human Processor) and GOMS are two of those models. The main assumption of MHP is that the human mind is an information processing system described by three processors (perceptual, cognitive, and motor), two memories (working memory and long-term memory), and operation principles (Card et al., 1983). Processors operate in series and have cycle times, and operating principles are provided to describe and forecast performance. On the other hand, GOMS uses goals, operators, methods and selection rules to model and analyze user's behavior while interacting with a system (Card et al., 1983). GOMS is in fact a general term used to refer to a family of models, in which KLM is a variant.

KLM (Card et al., 1980) is one of the most comprehensive models in the area (MacKenzie, 2013). Despite it was introduced in 1980, it is still one of the most useful models in the field. KLM predicts the time to execute a task by expert users given the method, which must be specified in detail at the level of keystrokes, and the performance must be error-free. KLM states that the *execution part* of a task is described in terms of operators. (A unit task has two parts: *acquisition* and *execution* of the task (Card et al., 1980). Acquisition time is beyond the scope of the proposed model.) Operators, which in most cases are assumed to take constant times, are actions (e.g., pressing keys/buttons or pointing to targets) that users should follow to perform tasks. Then, the time to accomplish a task is computed by summing up the times required by such actions. Furthermore, many authors have used the KLM method over the years to propose extensions and/or new models to analyze other interaction styles, e.g., mobile phones (Holleis et al., 2007), handheld devices (Luo and John, 2005), and touchscreens (Lee et al., 2015). This fact demonstrates the strength and usefulness of KLM, but both the original KLM and its variants do not provide operators applicable to THG (Erazo and Pino, 2015).

Fitts' Law (Fitts, 1954) is another of the most widely cited and used models in HCI. Despite the initial Fitts' work was published in 1954, it is still used to make predictions with various types of interfaces. UI designers normally use Fitts' Law to compute the time it takes to click or select objects on a screen, e.g., using an input device such as a mouse. Mathematically, the time to rapidly move the cursor to a target is a logarithmic function of the corresponding distance and the target size; i.e., if a target gets smaller and/or further away, it takes longer to reach that target. Although fairly accurate values for optimal results can be calculated by applying Fitts' Law, it is necessary to know its applicability to the intended interaction style. For instance, the utility of this model has been verified with new interaction styles (e.g., those based on touchscreens (MacKenzie and Teather, 2012), and head mounted displays (Lubos et al., 2014)), also including touchless hand gestures (Schwaller and Lalanne, 2013; Pino et al., 2013; Polacek et al., 2012; Zeng et al., 2012; Sambrooks and Wilkinson, 2013; Jude et al., 2014). Consequently, Fitts' Law can be used to analyze interfaces based on THG, but its usefulness is constrained mainly to pointing tasks.

Since models must be validated, several metrics can be used to confirm whether they make good predictions or not. The R^2 value is a common validity metric (Cao and Zhai, 2007) reflecting the strength of the relationship between predicted and observed times. The percentage root mean square error (%RMSE) is another metric showing the percentage difference between predicted and observed values. Both metrics can be used to consider THGLM has acceptable quality as long as their values are similar to typical values in the field. As a consequence, there should be strong positive correlation between predicted and observed times (i.e., high R^2), and error percentages should be close to those reported in previous works (e.g., (Holleis et al., 2007; Luo and John, 2005; Lee et al., 2015), and particularly the original KLM (Card et al., 1980)) in order to verify our hypothesis *H1.*

2.2. Gestures

Although gestures generally refer to movements performed with the hands or other body parts to convey some meaning, their definition may change depending on the study field. One of the most accepted definitions in HCI (Webb and Ashley, 2012) was proposed by Kurtenbach and Hulteen (Kurtenbach and Hulteen, 1990):

“A gesture is a motion of the body that contains information. Waving goodbye is a gesture. Pressing a key on a keyboard is not a gesture because the motion of a finger on its way to hitting a key is neither observed nor significant. All that matters is which key was pressed.”

This definition could be applied to gestures executed with various body parts, for example, nodding the head, forming a “T” with the whole body, or pushing with a hand (Fig. 1, top). Nevertheless, this work only studies “hand gestures” without considering those ones performed with one or more fingers (e.g., fingerspelling).

Previous studies about human hand gestures have proposed to analyze gestures by considering their temporal structure. Gestures are defined in terms of *gesture units* (*G-units*), *gesture phrases* (*G-phrases*) and *phases* according to this approach (upper part of Figs. 1 and 2) (McNeill, 1992; Kendon, 2004). A G-unit is the entire “excursion” between successive rests of the limbs from the moment the limbs begin to move until reaching a resting position again. We may distinguish one or more G-phrases within the course of a G-unit. Similarly, a G-phrase (or a gesture) is comprised of one or more phases. *Preparation* is the first—and optional—phase in which the hand is moved away from its resting position to the position where a *stroke* starts. The gesture meaning is expressed in the stroke phase which is the only obligatory phase. Also, a G-phrase may have two optional hold phases preceding or following a stroke (called *pre-stroke hold* and *post-stroke hold* (Kita et al., 1998) respectively), but these phases are not used in THGLM (Erazo and Pino, 2015). The final phase, *retraction*, is not considered to be part of any G-phrase. It may happen when the hand relaxes and returns to some resting position or to the original one. In addition, two types of phrases are distinguished—*hold-phrase* (*H-phrase*) and *stroke-phrase* (*S-phrase*) (Erazo and Pino, 2015; Neff et al., 2008)—because some gestures can have a single meaningful still phase

G-unit = {G-phrases} + [Retraction]
G-phrase = S-phrase | H-phrase
S-phrase = [preparation] + **Stroke**
H-phrase = [preparation] + **Hold**

Fig. 2. G-units, G-phrases and phases (based on (McNeill, 1992; Kendon, 2004; Kita et al., 1998; Erazo and Pino, 2015)).

(called *independent hold* (Kita et al., 1998)) instead of a stroke for static gestures (or *hold gestures* (Neff et al., 2008)).

Although the main application of G-units is the analysis of gestures performed as part of humans' conversations or speeches, they have also been used in our area (e.g. to produce gestures of animated characters (Neff et al., 2008)). The advantage of using G-units is that they allow analyzing continuous production of gestures (Kita et al., 1998).

3. Touchless hand gesture level model (THGLM)

THGLM is a model based on KLM (Card et al., 1980) and gesture units (McNeill, 1992) whose goal is to estimate the time to accomplish a task using a NUI based on THG (Erazo and Pino, 2015). It assumes that the method to execute the task is known, executed without errors, and completely specified at the level of THG using a set of operators. The set of operators used by THGLM is different than the corresponding ones used by KLM because gestures are more complex than keystrokes, making it necessary to analyze them in a different way (i.e., using gesture units) (Erazo and Pino, 2015). Besides, it is necessary to make some assumptions so that the model is restricted to gestures performed by young adults in normal health conditions, with basic or no experience with touchless interactions, and using the whole hand (fingers are not considered). Though other simplifications may be needed to build or extend the model, it has an acceptable quality making these assumptions (Erazo and Pino, 2015).

THGLM is an additive model which is an important characteristic and one of the reasons to be considered relatively easy to use. In other words, the time to execute a task is equal to the sum of all G-units needed to describe that task (formula 1 below (Erazo and Pino, 2015)). The number of G-units depends on the times the user's hand begins to move and reaches a position of relaxation again, i.e., a G-unit is counted each time the hand departs from a resting position until the moment it returns to a resting position or the initial position (Erazo and Pino, 2015). For example, the period of time a user takes to raise the hand toward the interaction space, navigate through some pictures (e.g., using swipes or selecting buttons), and return to a resting position, is equivalent to a gesture unit. A G-unit time in turn is computed by summing up all G-phrases plus an optional retraction time because a G-unit can have one or more G-phrases (formula 2 (Erazo and Pino, 2015)). For example, a drag and drop task could consist of one G-unit with three G-phrases: grip, move and release an object (Erazo and Pino, 2015). The number of G-units and G-phrases also depends on the complexity of tasks and interface designs.

A G-phrase time is computed adding an optional preparation time with H-phrase time or S-phrase time as appropriate (formula 3; bear in mind two G-phrases are distinguished) (Erazo and Pino, 2015). H-phrase time is equal to the sum of *feedback time* plus *exit time* (formula 4 and according to (Müller-Tomfelde, 2007; Erazo and Pino, 2015)). (Feedback time is the time established by designers that users must hold the hand to consider the action valid; exit time is the time the user's hand remains in the same position or pose after feedback time is completed and the hand moves away (Müller-Tomfelde, 2007).) S-phrase time is computed in a way similar to KLM as proposed in the original version of THGLM (formula 5); i.e., summing up the times of each of the needed stroke-phrase operators (Erazo and Pino, 2015).

$$T_{execute} = \sum_{i=1}^{m} T_{\mathrm{Gunit}_i} \tag{1}$$

$$T_{\mathrm{Gunit}} = \left(\sum_{j=1}^{n} T_{\mathrm{Gphrase}_j} \right) + [T_r] \tag{2}$$

$$T_{\mathrm{Gphrase}} = [T_p] + \{T_{stroke} \mid T_{hold}\} \tag{3}$$

$$T_{\mathrm{hold}} = feedback\_time + exit\_time \tag{4}$$

$$T_{\mathrm{stroke}} = \sum_{op \in OP} n*op \tag{5}$$

Three groups of operators derive from the previous model formulation: movement operators, expressive operators and general operators (Erazo and Pino, 2015) (Table 1). The first group is composed of four operators. There are two optional operators for preparation (***Pr***) and retraction (***Re***). ***Pr*** (moving the hand from a resting position to the position where a stroke begins) should be used each time the user needs to physically prepare the hand to perform a gesture, which is different than mental preparation. ***Re*** should be used to represent the movement of the hand from the position where a stroke or hold finishes to a resting position. The third operator, ***P***, is for pointing to targets on a display with the hand in the air. This operator can take a constant value or its value can be computed using Fitts' Law. Additionally, the THGLM authors suggest being careful when placing ***P*** and ***Pr*** operators together to avoid redundancy. Furthermore, ***P*** only provides the time to point a target (Card et al., 1980), and hence, an expressive operator must be used for the following sub-action. Finally, an alternative operator for preparing the hand when performing swipes is included in THGLM. Actually, two other options to distinguish between horizontal and vertical swipes were proposed.

As suggested above, there are two kinds of expressive operators: *H-phrase* and *S-phrase* operators (Erazo and Pino, 2015). There are various operators that could be included in each category. Thus, Erazo and Pino performed a systematic literature review with the aim of finding "the most used (and/or suitable to use) gestures in NUIs based on THG or touchless interaction". As a result, four operators were initially selected as S-phrase operators, and one as H-phrase. The H-phrase operator (***H***, Holding) is used with static gestures or holds a hand on a target, position or pose a pre-set time. The selected S-phrase operators are (1) push the hand toward the front (***T***, Tapping); (2) move a hand from right to left or vice versa (horizontal swipe, ***Sh***), from top to bottom or vice versa (vertical swipe, ***Sv***), or swipe in general (***S***, swiping); (3) close the hand (***G***, gripping); (4) open the hand (***R***, releasing). The values for these operators were estimated by conducting a user study, which also allowed estimating the values for ***Pr*** and ***Re*** operators.

Drawing gestures (air figures of letters, numbers, and shapes) can also be included in THGLM as S-phrase operators, but no formula (or value) was provided for this operator (Erazo and Pino, 2015). However, this type of gestures was studied in detail in another work. Specifically, three models—developed for other interface types—were assessed in that study with the aim of extending them to estimate the production time of touchless hand drawing gestures (THDG) (Erazo et al., 2015). The authors concluded that the three models can be used with THDG. Furthermore, they provided new or updated formulas and empirical constants needed to use the models. Therefore, we select the model (formula 6) that corresponds to the best evaluated one to be included in THGLM as drawing operator, which is a variant of the ***D*** operator of KLM.

Table 1
Overview of the proposed operators with the corresponding values (based on (Erazo and Pino, 2015)). [a] This value corresponds to the total time of holding (i.e., 1 s). [b] According to (Erazo et al., 2015) and as discussed in the text.

Operators				Description	Time (s)	SD (s)
Expressive	*H-phrase*	*H,* Holding		Perform static gestures or holding a hand on a target, position, or pose, a pre-set time.	0.500 + feedback_time	0.103[a]
	S-phrase	*T,* Tapping		Pushing the hand toward the front.	1.108	0.370
		S, Swiping	Mean	Moving the hand from right to left or vice versa (horizontal swipe), from top to bottom or vice versa (vertical swipe), one time and returning to the starting position.	0.553	0.211
			Horizontal (***Sh***)		0.613	0.208
			Vertical (***Sv***)		0.493	0.198
		G, Gripping		Closing the hand.	0.586	0.152
		R, Releasing		Opening the hand.	0.520	0.172
		D, Drawing [b], $D_c(n_D, l_D,\ n_C)$		"Drawing" shapes, numbers, etc. in the air.	$a\,n_D + b\,l_D + c\,n_C$	N/A
Movement		***Pr,*** Preparation		(Optional) Moving the hand from a resting position to the position where a stroke begins. It should be used each time the user needs to physically prepare the hand to perform a gesture, which is different than mental preparation.	0.452	0.103
		Re, Retraction		(Optional) Moving the hand from the position where a stroke or hold finishes to a resting position.	0.746	0.106
		Sp, Swipe preparation	Mean	Preparing the hand for next swipe.	0.624	0.325
			Horizontal		0.562	0.361
			Vertical		0.685	0.274
		P, Pointing		Pointing to a target on a display with the hand in the air. A constant time is proposed, but Fitts' Law may be used instead. Be careful when placing the ***P*** and ***Pr*** operators together to avoid redundancy.	1.046	N/A
General		***M,*** Mentally prepare		Mentally preparing to execute subsequent physical operations. (See next section.)	0.927	0.116
		SR(t), Response Time		The time the system needs to respond to user input.	*t*	N/A

$$D_c(n_D, l_D,\ n_C) =\ a\,n_D + b\,l_D + c\,n_C \tag{6}$$

where n_D = number of segments, l_D = total length of all segments, n_C = number of corners, $a = 0.223$, $b = 0.297$, $c = 0.173$.

The operators that belong to the third category, general operators, are the same ones introduced in the original KLM (Card et al., 1980) and used in other KLM extensions (Holleis et al., 2007; Luo and John, 2005; Lee et al., 2015). The response time operator (***SR(t)***, the time the system needs to respond to user input) remains unchanged, and the ***M*** operator in turn is defined similarly but with a value different than the original one (i.e., 0.927 s instead of 1.35 s) (Erazo and Pino, 2015). This value was also estimated in the user study to compute the times for the other THGLM operators. It is lower than the one proposed by Card et al. (Card et al., 1980), but it is in the range of 0.6–1.35 s suggested by Kieras (2001). Furthermore, although some authors have used the original value (Holleis et al., 2007; Luo and John, 2005; Lee et al., 2015), other authors advocate to update this operator (MacKenzie, 2013)(p. 272). Consequently, the THGLM authors computed their own value, and emphasized the need of studying this operator in further detail (Erazo and Pino, 2015).

4. Using THGLM

4.1. Including mental operators

Mentally Prepare is an operator that needs special attention. As noted by Kieras (2001), including mental operators is tricky; it requires a lot of judgment, and it is necessary to hypothesize on how users think about tasks rather than only which movements they have to perform. Moreover, the set of heuristic rules, which was provided with the original KLM and should be followed to use this operator, must be revised to make the necessary interpretations. Thus, this section contains that revision and some recommendations.

4.1.1. Heuristic rules for placing M operators

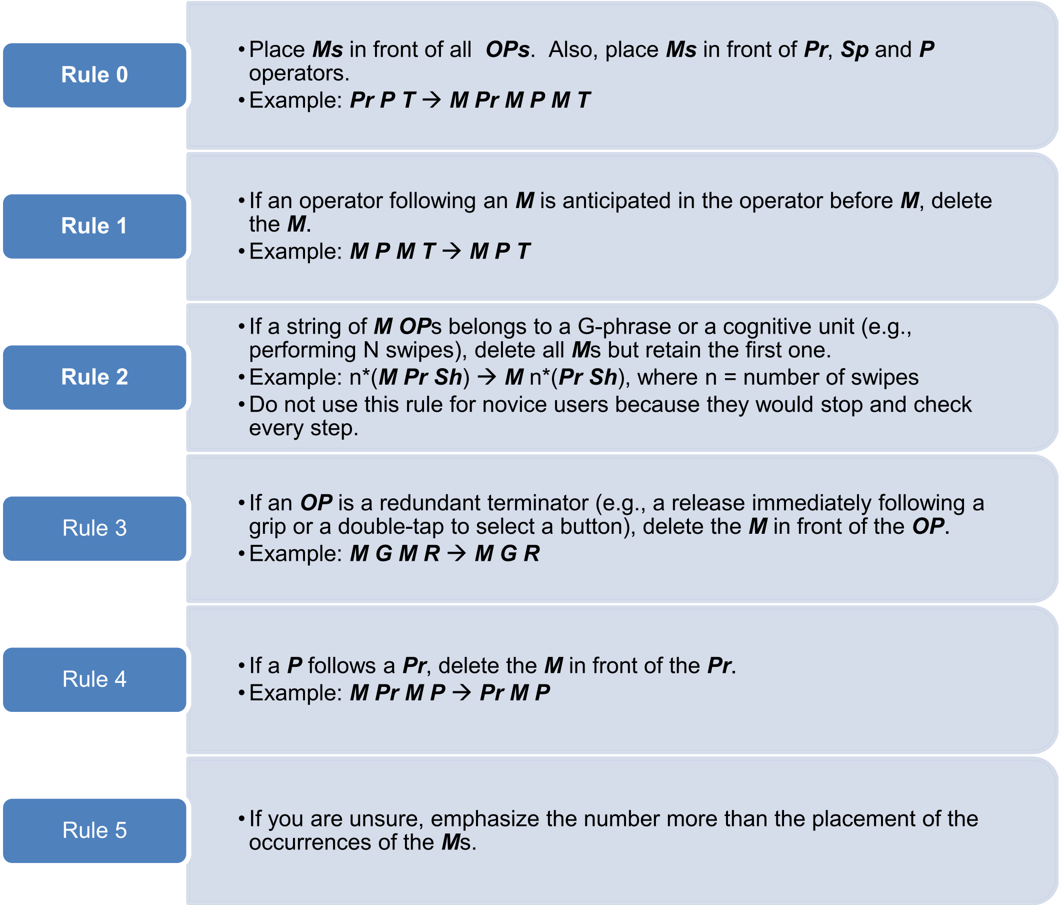
Fig. 3 shows the THGLM heuristics with the corresponding examples. These heuristics have been revised and/or adapted from the original KLM heuristics (Card et al., 1980) and taking into account Kieras' suggestions (Kieras, 2001). Bear in mind that *OPs* refer to both S-phrase and H-phrase operators in this section.

4.1.2. Other recommendations for placing M operators

As we mentioned above, Kieras (2001) provided some recommendations to use the ***M*** operator. The following is a summary of some of those recommendations —with the needed adaptations and examples— for activities that take an ***M***.

1. Pausing before initiating a task or performing a sequence of actions.
2. Stopping and thinking to make a strategy decision; e.g., choosing one from two or more options.
3. Retrieving a cognitive unit from memory; e.g., remembering the gesture to execute a command.
4. Pausing to scan and find something on the screen; e.g., a button that should be pressed to perform the next step.
5. Pausing to check an action or entry; e.g., verifying the actual element after performing a swipe.
6. Pausing to check the result when the screen changes in response to user input; e.g., performing a swipe when browsing a map.

Additionally, it is necessary to make distinctions of using ***M*** operators between novice and expert users. It is expected new NUI users would become experts with little to no training (Wigdor and



Rule 0

- Place ***Ms*** in front of all ***OPs***. Also, place ***Ms*** in front of ***Pr***, ***Sp*** and ***P*** operators.
- Example: ***Pr P T*** → ***M Pr M P M T***

Rule 1

- If an operator following an ***M*** is anticipated in the operator before ***M***, delete the ***M***.
- Example: ***M P M T*** → ***M P T***

Rule 2

- If a string of ***M OP***s belongs to a G-phrase or a cognitive unit (e.g., performing N swipes), delete all ***M***s but retain the first one.
- Example: n*(***M Pr Sh***) → ***M*** n*(***Pr Sh***), where n = number of swipes
- Do not use this rule for novice users because they would stop and check every step.

Rule 3

- If an ***OP*** is a redundant terminator (e.g., a release immediately following a grip or a double-tap to select a button), delete the ***M*** in front of the ***OP***.
- Example: ***M G M R*** → ***M G R***

Rule 4

- If a ***P*** follows a ***Pr***, delete the ***M*** in front of the ***Pr***.
- Example: ***M Pr M P*** → ***Pr M P***

Rule 5

- If you are unsure, emphasize the number more than the placement of the occurrences of the ***M***s.

Fig. 3. Set of updated heuristics for placing M operators (based on (Card et al., 1983; Kieras, 2001)).

Wixon, 2011), but designers could want/need to consider both options especially because today there are still few users with extensive experience with UIs based on THG. The following recommendations (again, adapted from (Kieras, 2001)) may be applied in this case.

- ➢ New users will stop to verify every step or check feedback. Consequently, the recommendations (4), (5) and/or (6) would not be applicable to experienced users.
- ➢ New users have small cognitive units, whereas expert users have large cognitive units. Therefore, an experienced user could perform a task requiring one G-phrase and the same task could require several G-phrases for a novice user.
- ➢ Experienced users may overlap ***M***s with physical operators. For instance, a user may think about the next step or locate a button on the screen while s/he is performing a stroke.

Finally, consistency is important in placement of ***M*** operators (i.e., apply the same rules to all designs).

4.2. A procedure to apply THGLM

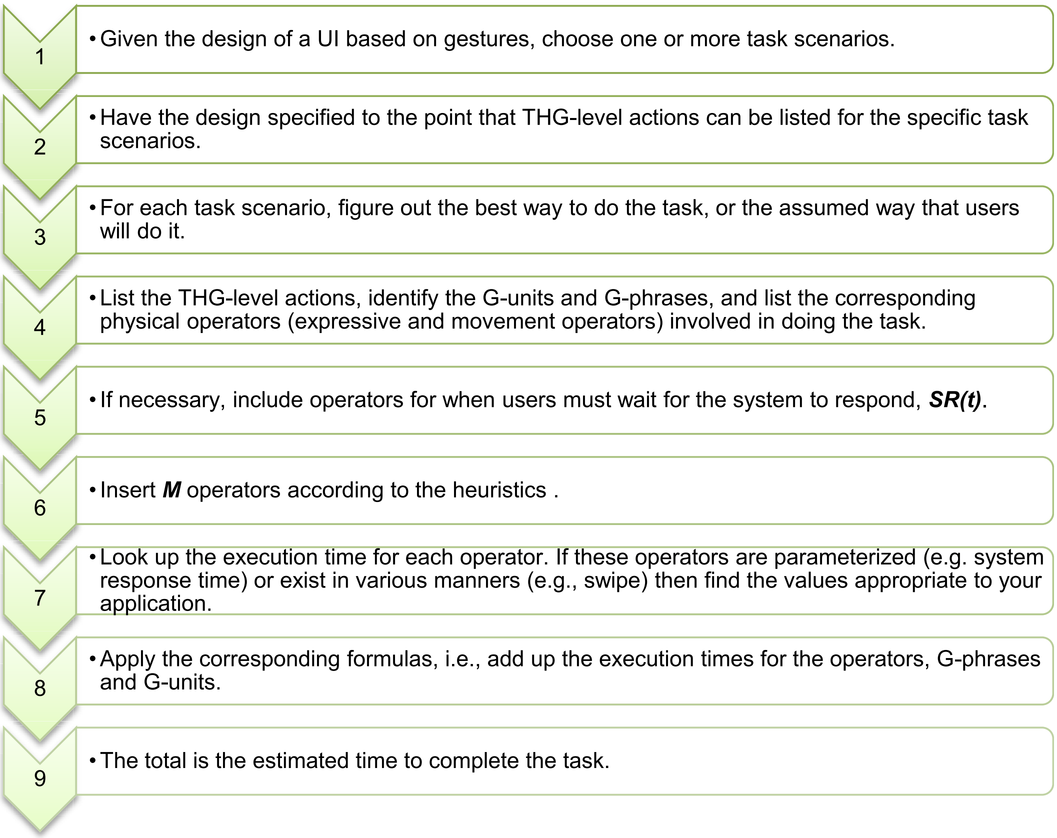
In addition to the model formulation, a procedure to apply it would be useful to achieve good predictions. Fig. 4 describes the steps needed to be taken to apply THGLM. In general, this procedure is similar to the one to apply KLM taking into account that THGLM is based on the first one. Thus, we reproduce the procedure from (Kieras, 2001) with the corresponding changes or additions, taking also into account the modifications made in (Holleis, 2009).

5. Validation of THGLM

An inevitable question arises after building the model: Are the predictions made using THGLM acceptable? In other words, we want to know whether predicted times to perform tasks on UIs based on THG are close to the corresponding ones observed when users perform such tasks. Therefore, an *empirical validation* (Card et al., 1980) is needed to know it.

THGLM has been validated using three applications (Erazo and Pino, 2015). The first one (called NUIPy) allows solving a kind of puzzles that represent basic programs written in Python. If a user selects a statement in the right order, it is executed in Python IDLE. The second application (called OctaNUI) allows interacting with Octave—which is a high level interpreted language for numerical computations, similar to Matlab—to run basic commands. Users have to select and execute a command with a dataset. Furthermore, InteractionGallery and KinectPaint were used by new participants as part of the validation, but the authors only processed the data collected with InteractionGallery. Despite the results achieved in this evaluation were acceptable, the authors noted there are some limitations being necessary to perform further evaluation (Erazo and Pino, 2015).

On the other hand, the experiments performed to validate the model were focused only on determining the model performance, and hence, they did not include other options that should be evaluated. These options are the use of the model to compare



1 • Given the design of a UI based on gestures, choose one or more task scenarios.

2 • Have the design specified to the point that THG-level actions can be listed for the specific task scenarios.

3 • For each task scenario, figure out the best way to do the task, or the assumed way that users will do it.

4 • List the THG-level actions, identify the G-units and G-phrases, and list the corresponding physical operators (expressive and movement operators) involved in doing the task.

5 • If necessary, include operators for when users must wait for the system to respond, ***SR(t)***.

6 • Insert ***M*** operators according to the heuristics .

7 • Look up the execution time for each operator. If these operators are parameterized (e.g. system response time) or exist in various manners (e.g., swipe) then find the values appropriate to your application.

8 • Apply the corresponding formulas, i.e., add up the execution times for the operators, G-phrases and G-units.

9 • The total is the estimated time to complete the task.

Fig. 4. Procedure to apply THGLM (based on (Kieras, 2001; Holleis, 2009)).

various designs and ask UI designers to use the model.

Given this fact, we decided to perform the missing evaluation. We started processing the data collected and not analyzed in the initial evaluation and performing the corresponding analysis. Additionally, two new applications were used to confirm whether the model makes good predictions or not. Moreover, another experiment was conducted with the aim of using the model to compare several interface designs. All of these experiments are described in this section, except the experiment involving designers which is described in section 6.

5.1. Further model validation

5.1.1. Method

5.1.1.1. Apparatus. Two apparatuses were used in order to collect data of participants interacting with three applications. The first one was the apparatus employed in (Erazo and Pino, 2015) for the initial model validation. It consisted of a notebook, a projected display, and a Kinect sensor. The application used in this case was KinectPaint, which allows painting on a canvas and selecting buttons or options using holding gestures.

The hardware setup of the new apparatus (introduced in this work) consisted of a display (24 inches, 1920 × 1080 pixels of resolution), a computer (equipped with an Intel Core i7 processor, 16 GB of RAM), and a Leap Motion (LM). Both the display and the LM were placed on a desk at a height of 75 cm. The participants sat on a chair with armrests in front on the display (at about 1 m.); and both the height of the display and the chair were adjusted according to each participant needs until reaching a comfortable position. Moreover, two applications were used with this setup, which were named *Bmog* and *Gester*. Bmog allows browsing genre and movies projected on the display, and watching a trailer and/or getting information about the selected movie. It has four buttons to go forward and backward, select a movie, and cancel the process. Gester is a game in which users have to “catch” the letters of a word by performing a gesture. The used words were data types such as int, char, etc. The letters appeared in the same position for each participant, but in different positions for each word. After selecting the “start button”, the first letter of the word appears, and when the user selects it, it disappears and the next letter is shown until catching all the letters. Furthermore, we decided to develop these applications using Scratch with the aim of performing the tests with a different platform from previous experiments and encouraging participants to try developing their own applications in a near future. Also, Hover gestures were used in Bmog to make selections, whereas Grip gestures were used in Gester.

5.1.1.2. Participants. Seventeen volunteers performed the tasks using the three applications. Nine participants (mean age 19 years, $\sigma = 1$; 8 right-handed; 4 male) interacted with Bmog and Gester, and the remaining eight participants tried KinectPaint (according to (Erazo and Pino, 2015)). Also, six participants that used Bmog and Gester self-declared to have some previous experience on gaming using touchless interaction (e.g., Kinect for playing games).

5.1.1.3. Procedure and tasks. The experiment (using the new apparatus) consisted of a practice session and two repetitions of two tasks (one with each application). Participants performed the practice after receiving verbal instructions from the experimenter. They tried the corresponding application in a free way for a couple of minutes during the practice. Next, participants performed the task using that application. This process was repeated for the other task but alternating the order of applications. Participants had to “catch” the letters of the word “char” using Gester, and play the trailer of a movie (specifically, the fifth genre, and the third movie) with Bmog, using their dominant hands. The tasks performed with all applications were video recorded and segmented manually by using VirtualDub (a free video tool for basic editing, mainly geared to AVI files) to compute duration times.

5.1.2. Results

Using the three aforementioned applications, the times to accomplish five tasks were observed and used to analyze and confirm how well the model works. The times for all tasks were estimated using the model with the proposed operator times (Table 1). These values were compared against the observed ones as shown in Fig. 5. About 14% of observed values, that corresponded to task instances with significant errors or in which participants did not follow the prescribed method, were discarded (according to (Card et al., 1980)). The reached RMSE is 16.3%, while the average of the absolute prediction error is 15.3% (min: 7.2%, max: 24%). The worst value corresponds to the task performed using Gester which may be due to the mental act to reach the next target is approximate to a simple reaction, and hence, the ***M*** operator should have a smaller value in this case. (This idea is discussed in further detail below).

5.2. Using the model to analyze interface designs

We conducted another user study in order to verify the usefulness of the model as a tool to analyze UI designs of applications. The chosen application should allow a user to take a photo after selecting the desired background/wallpaper picture. (Other functionalities could be included as part of this application, but we consider the proposed ones are enough for the goals of this work.) The study consisted on analyzing three design options for this application to select the best one. The first option (D1) uses buttons to interact with the application by holding the hand during one second over them (one button for going forward, one button for going backward, and another to take photos). Users hold the hand over the desired button for one second to select it. The application uses no buttons as a second design option (D2) because swipe gestures are used to navigate through pictures and a combination of grip and tap gestures should be performed to take photos. The final option (D3) is a variation of the second one, that is, the gesture to take photos is replaced by drawing a check.

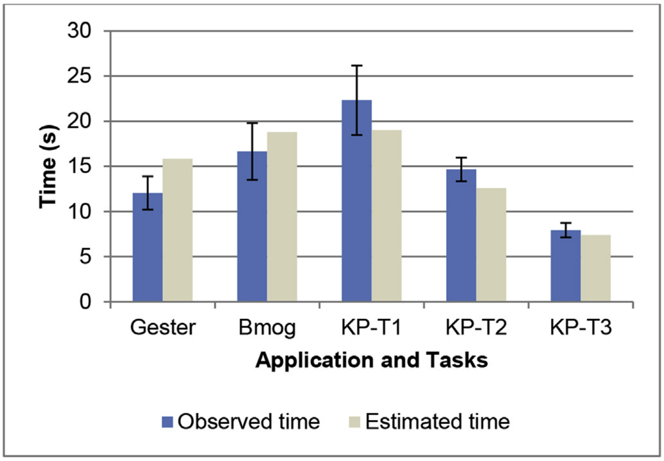


Fig. 5. Comparison of observed and predicted times for the five tasks. Error bars indicate 1 SD. KP = Kinect Paint.

5.2.1. Method

5.2.1.1. Apparatus. The three design options were implemented using Microsoft Visual C# and Kinect for Windows SDK V1.8 on Windows 7. Hand movements, joint positions, and $1 algorithm (Wobbrock et al., 2007) were used for gesture recognition. Moreover, the application logged the duration times of each task, and the researcher recorded entries in a log on the wrong tasks.

On the other hand, the hardware setup consisted of the same display and computer described in the previous experiment, but we used a Kinect sensor instead of the LM. The Kinect was placed at a height of 1 m and used with a refresh rate of 30 fps. Participants stood 2.5 meters away from the Kinect while performing the tasks.

5.2.1.2. Participants. Eight volunteers (mean age 28 years, $\sigma = 8$; all right-handed; 5 male) were recruited to take part in this experiment. Six of them had some previous experience on gesture interfaces such as playing games with Wii and/or Kinect.

5.2.1.3. Procedure and tasks. The study started with a verbal explanation about the application goal and general instructions. Next, each participant had to perform three tasks: one task using each design option. To do it, participants received the instructions concerning the gesture to use with each option, and then, they were allowed free practice for a couple of minutes. They accomplished each task for data collection after this period of practice. The task consisted on taking a photo of him/her with the fifth background executing the proper gestures. Moreover, the order of design options was determined by applying Latin squares as a within-subject design was used.

5.2.2. Results

The three design options were compared following the same procedure to analyze the model quality described above (Fig. 6), which included discarding about 25% of wrong task instances. According to the estimated vales, the order of design options from best to worst is: D2, D3, D1. Observed values confirmed this order. In other words, the comparison made to choose the best design option gave the same results using the model and observing users while interacting with the prototype. Furthermore, the prediction errors of the three designs remain below the baseline.

5.3. Discussion

We have described two studies in this section in order to verify both whether THGLM makes good predictions and whether it can be used as a design tool (hypotheses *H1* and *H2* respectively).

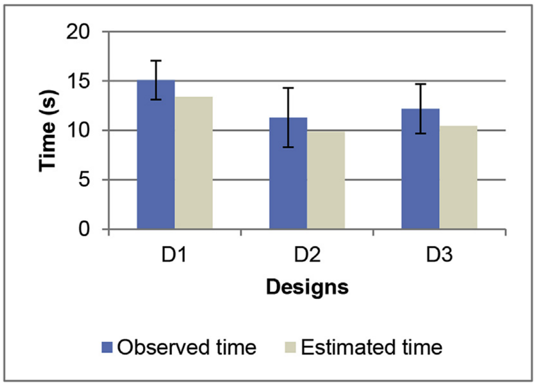


Fig. 6. Observed and predicted times for the three design options (D). Error bars indicate 1 SD.

Precisely, the results from the first study confirm the model has an acceptable quality. Though the reached RMSE is higher than the one reported in the initial validation of THGLM (RMSE = 10.1%) (Erazo and Pino, 2015), it is still lower than the baseline (21% Card et al.'s error (Card et al., 1980)). In addition, we calculated the global model performance based on the selected metrics. In other words, the %RMSE and the R^2 were computed using the data from the initial model evaluation reported in (Erazo and Pino, 2015) and the new validation reported here. The resulting metric values using six applications and nineteen tasks are as follows: %RMSE = 12.1%, $R^2 = 0.917$. Fig. 7 plots the correlation between estimated and observed execution times.

We took the next step based on those metric values, which is independent designers used the model to compare several design options. The results support the hypothesis (*H2*) that the model can be used to compare two or more UI designs and choose the best one. This fact allows suggesting THGLM can be used by designers as a tool to assess or analyze UIs based on THG.

6. Validation with designers

We have shown in the previous section that THGLM has an acceptable performance, though it is not enough to use the model with confidence. The model performance was determined using the estimations made by one of the researchers. This process is acceptable to evaluate the model, but it does not allow knowing whether estimated values are stable across UI designers. In other words, the values predicted by the researchers should be consistent with the ones predicted by one or more designers to conclude the model is valid or not (Stanton and Young, 2003).

Given that THGLM is based on KLM, it is arguable that its estimations are consistent across predictions. As mentioned above, a large body of research demonstrates the original KLM is a well validated model. Moreover, several researchers have verified the validity of KLM when numeric predictions are produced by independent designers including novices (Stanton and Young, 2003; John et al., 2004). However, THGLM adds not only new operators to allow using THG; it involves modeling at a gesture level instead of keystrokes and using other concepts (G-units and G-phrases). Furthermore, the heuristics to place ***M*** operators have been revised and adapted, making it necessary to evaluate whether designers can apply them consistently or not. Therefore, the THGLM predictions should be tested with UI designers.

An empirical study was conducted in order to assess the model predictions with designers' participation. Specifically, this study addressed the question:

Are THGLM predictions consistent when computed by independent designers to forecast performance time on UIs based on THG?

The study to answer the question had two parts. The first one was performed with the aim of gathering predicted values from UI designers using the model. Times to execute the corresponding tasks using the application prototype were collected from users in the second part of the study. Predictions made by both a researcher and designers using the model, and observed values, were compared to finally determine the model consistency. This study and results are explained in this section with further detail.

6.1. Method

6.1.1. Part 1: model predictions

The first part of the study was collecting data from UI designers. Thus, we asked participants to analyze a UI of a "hypothetical" application using the model. The goal of the proposed application was to do brainstorming using THG, using only the gestures included in THGLM as operators. Also, designers were provided with an initial UI design (Fig. 8) and the needed files (explained below).

6.1.1.1. Participants. Eight Computer Science undergraduate students participated in the study as UI designers. All of them were senior university students and had attended a course on HCI previously, but none had previous experience on model-based evaluation. An instructor invited them to take part in the study and gave them class credit points for the participation.

6.1.1.2. Procedure and tasks. After the designers accepted participation in the study, a researcher emailed them the study instructions and the needed files (five files in total) to accomplish the tasks. The students used the model according to the instructions to do the tasks and handed the required documents before the due date.

The instructions, described in a pdf file and composed of six steps, started with a short introduction about models, gestures, and the proposed application. In the first step, we asked the designers to get acquainted with the model by reading the pdf file that contained the model description (a simplified version of section 3), a procedure to apply it (similar to section 4), and several examples of the use of the model. These examples were also provided in a

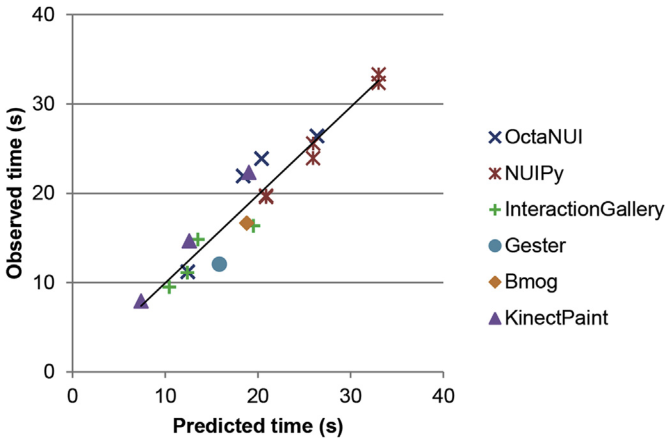


Fig. 7. THGLM prediction.

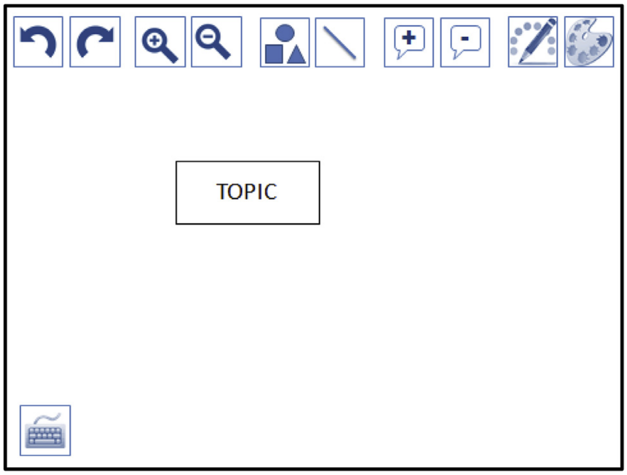


Fig. 8. User interface used in the study.

spreadsheet with the operator values and the needed formulas to make the estimations. When the designers considered they had learned to use the model, they had to register the spent time to do it using another file (a doc file).

We provided the initial UI (Fig. 8) and the corresponding explanation of its components in the second step. Moreover, in this step, we asked the participants to watch a Power Point presentation that contained a simulation of the application behavior.

Next, the designers applied the model in steps 3 and 4 to analyze three tasks using the initial design and a modified version respectively. “Add a topic” (add a topic, show the keyboard, type a text, and connect the topic to a previous one), “change topic colors” (select a topic, change border and fill colors) and “delete a topic” (select a topic and delete it) were the used tasks. Designers were instructed to use only hover gestures to analyze the tasks with the initial design (design D1) in step 3. In step 4, designers had to use tap gestures to make selections and grip & release to connect topics instead of using hover gestures (design D2), and the button to show the keyboard was discarded (i.e., the keyboard would appear automatically). After applying the model, the designers had to make a comparative analysis of both designs. As a final point, they had to record the required time to analyze each task. (In fact, we emphasized since the beginning of the instructions that designers had to record the required time for each part.)

In addition to the previous tasks, we requested designers to answer a questionnaire to evaluate the model and the procedure to apply it (step 5). They rated five questions (using a scale from 1 to 7, from low or totally disagree to high or totally agree respectively) regarding model explanation, heuristics for mental operations, procedure to apply the model, examples, and general evaluation.

The designers handed two files as a final step: the questionnaire as a doc file, and a spreadsheet with the analysis of all tasks. Thus, the results reported below are based on these files.

6.1.2. Part 2: observed values

6.1.2.1. Apparatus. The application prototype used the interface described in the previous section (Fig. 8) in order to ask users to perform the three aforementioned tasks. In fact, the application had two versions, one for each analyzed design. The application was developed using MS Visual C# and Leap Motion SDK 2.2.5 on Windows 7.

The hardware setup consisted of a desktop computer, a display, and a gesture input device, mounted in a laboratory in our university campus. The computer was equipped with an Intel Core i7 processor, 16 GB of RAM. A LM was connected to the computer to track participants' hand and recognize gestures. Also, the LM was placed on a desk (at a height of 75 cm), between the participant and the display. The display, which had a resolution of 1920 × 1080 pixels, was also placed on the desk at 0.9 m from the user. The participants sat in front of the display (with a white wall behind it) on a chair with armrests. All the display, the chair, and the armrests heights were adjusted according to each participant's height and preferences until she/he was in a comfortable position.

6.1.2.2. Participants. Ten healthy undergraduate students, five male and five female (mean age 21 years, $\sigma = 5$; nine participants were right-handed), took part in the study. Seven participants had some basic experience on touchless interaction, such as using Microsoft Kinect for playing games, whereas the other three had no prior touchless interaction experience. All participants self-declared their experience on THG and other demographic characteristics in a final questionnaire. Additionally, the University approved the study with students, and written informed consent was obtained from all participants.

6.1.2.3. Procedure and tasks. The experiment started with a researcher's explanation about the application and tasks. Initially, the participants carried out a practice session by using the application for a couple of minutes in the way they considered appropriate. When they learned how to use the application, the researcher explained the tasks. The performed tasks were the same ones analyzed by designers, that is, add a topic, change colors, and delete a topic, and using both design options. Each task was executed twice for each design using Latin squares to determine the order. Likewise, the order of design options was interchanged between participants. Also, the application logged the time of each task, and the researcher took notes about wrong tasks.

6.2. Results

Fig. 9 shows a comparison between observed times and the times calculated by both a researcher and independent designers. About 27% of observed values that correspond to task instances with significant errors were excluded from the analysis. In general, this analysis consists of three comparisons. The first one is the “classical” comparison we have described previously, i.e., observed values vs. values estimated by a researcher. The percentage difference between these values remains near to the ones reported above. The comparison of the estimations made by designers is very interesting. Fig. 9 reveals that the means of values predicted by designers are approximate to the values computed by the researcher. Likewise, designers' times are similar to the observed values.

These comparisons give a general idea of the consistency of THGLM, but it is also necessary to consider the individual designers' values to confirm it. With this aim, we computed the percentage difference between researcher's times and designers' times, and between observed times and designers' times. The average %RMSE in the first case is 12%, whereas 18.3% in the second case. Similarly, the strength of the relationship between values estimated by the researcher and designers is $R^2 = 0.929$; and $R^2 = 0.892$ for designers' values and observed values. Furthermore, 79.2% of the designers' estimations followed the same pattern than the researcher's ones doing either overpredictions or underpredictions.

Fig. 9 reveals another aspect that deserves attention as well. The study was designed in a way that allows comparing two interface designs (D1 and D2 in Fig. 9). D2 should be preferred to D1 according to the researcher's and designers' analysis, and the observed times.

Regarding the data self-reported by designers, they suggested they had no problems to learn to use the model. Fig. 10 shows the obtained scores of the five questions to evaluate the clearness and

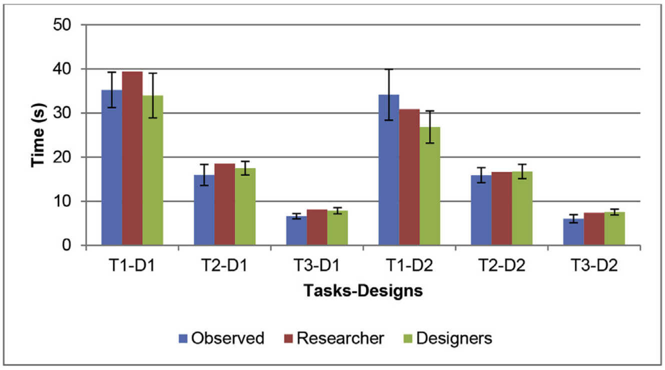


Fig. 9. Comparison of observed times and times predicted by a researcher and designers. T = Tasks, D = Designs. Error bars indicate 1 SD.

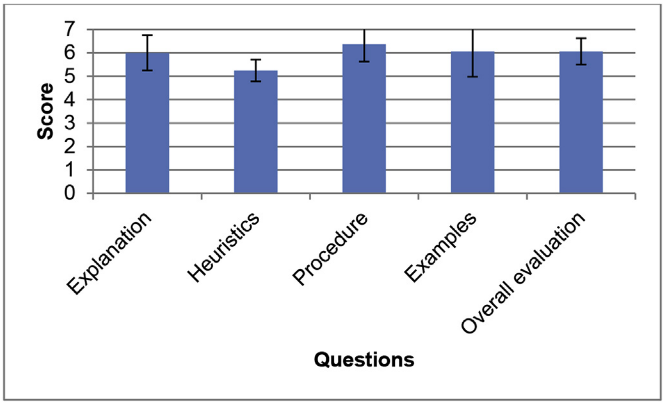


Fig. 10. Designers' opinions about the procedure to apply THGLM. Error bars indicate 1 SD.

ease of explanations and the procedure to understand and use the model. Although the heuristics use/explanation got the lowest value (5.3 out of 7), the remaining scores are above 6. It may be related to the fact that four designers mentioned they got slightly confused while reading the heuristics explanation. In addition, the mean time reported by designers to read the document, know/understand the model, and apply the model to produce the numeric predictions for all tasks was 65 min ($\sigma = 40$, min: 36, max: 140). Actually, the designer who reported the shortest time to do everything called our attention because he made the worst estimations. Consequently, we repeated the analysis after excluding this designer, and the metric values related to the model stability improved (%RMSE = 8.7%, $R^2 = 0.956$, between researcher's times and designers' times; %RMSE = 17.9%, $R^2 = 0.917$, between observed times and designers' times; 85.7% of cases followed the researcher's pattern).

6.3. Discussion

The results described in this section give a general idea of the consistency of THGLM when it is used by independent modelers. This study is not an exhaustive evaluation of the model predictions produced by designers because other aspects can be considered (e.g., involving designers with wide experience on model-based evaluation). However, this study constitutes another evaluation of the model validity that gives further evidence for the previous findings and supports hypothesis *H3*.

Beyond the possible limitations of this study, THGLM is a valid method to be used by UI designers according to the results. We reach this conclusion because the predictions were made by independent designers and compared against user trials performed by other subjects. The designers' predictions remain acceptable in both cases, i.e., comparing them with the researcher's predictions and with the observed values. Moreover, the results confirm the model validity as a tool to analyze or compare UI designs.

The designers also reported no problems on understanding and using the model despite they had no prior experience with HCI models. In fact, they only suggested a better explanation on the use of heuristics. However, it is expected novice designers have some problems applying heuristic rules (e.g., novice designers may include more ***M***s than experts (John et al., 2004)) because including mental operators may be tricky and it takes a lot of judgment as Kieras suggests (Kieras, 2001). Despite this difficulty, the designers provided high scores concerning the explanation and use of the model from which it is possible to infer the model is relatively easy to use. This ease of use may be confirmed by the relatively short times required to learn to use and apply the model. These times are also comparable with times reported in previous works such as (John et al., 2004). Finally, these small values support the economic benefits of THGLM. Numeric predictions can be produced in an easy and quick way in comparison with the logistic difficulties and costs of doing tests with real users (i.e., planning, timing, laboratory setup, recruiting subjects, and conducting experiments).

7. Extending THGLM

THGLM does not include all operators that may be used to analyze interface designs, but it is not a major limitation because the model is extensible following the inspiration of KLM (Erazo and Pino, 2015). For this aim, it is necessary to define the operator that will be included in the model, and then, estimate the value of that operator or find the equation(s) to compute its value. It is expected most of the new operators will correspond to gesture strokes, and hence, they will be included as S-phrase operators. The *Drawing* operator is an example of a new S-phrase operator that was added to the original model, as described in previous sections. Finding all candidate operators that may be incorporated to the model is beyond the scope of this work, but we introduce in this section several operators that should be useful.

7.1. The candidate operators

7.1.1. Mentally prepare

The first analyzed operator, to possibly be improved, is *Mentally Prepare* (***M***). It was introduced in the original KLM (Card et al., 1980), and the authors noted: “the use of a single mental operator is a deliberate simplification” (Card et al., 1980). For this and other reasons, MacKenzie (2013) advocates updating the ***M*** operator by replacing it with a set of operators, though this operator has been successfully used in other works using the original value (e.g., (Luo and John, 2005; Holleis et al., 2007; Lee et al., 2015)).

Based on (Card et al., 1983), MacKenzie (2013) (pp. 272–274) proposes to use five ***M*** operators depending on the required mental operation for simple decision tasks. These tasks are: simple reaction ($\boldsymbol{M_S}$, the user is attending to the application, and s/he reacts by doing an action when the stimulus appears), physical matching ($\boldsymbol{M_P}$, the user executes the action if the stimulus matches to a code stored in short-term memory), name matching ($\boldsymbol{M_N}$, similar to $\boldsymbol{M_P}$ but the user must abstract the stimulus in some way), class matching ($\boldsymbol{M_L}$, the user has to access the long-term memory before doing the action), choice reaction ($\boldsymbol{M_C}$, the user has to make a choice from several responses; Hick-Hyman Law (Hick, 1952; Hyman, 1953) is usually applied to analyze it), and visual search ($\boldsymbol{M_V}$, the user searches for a number of choices on the screen). Additionally, MacKenzie (2013) provides the estimated values for the operators, except for $\boldsymbol{M_C}$; and he also makes a comparison with the values computed by Card et al (Card et al., 1983) (who in turn did not provide a value for $\boldsymbol{M_V}$). However, these values were computed using an interaction style different than touchless interaction; in fact, MacKenzie's work (MacKenzie, 2013) used keystrokes to estimate those values. Consequently, it is necessary to verify what will happen when using THG instead of keystrokes in order to use more than one ***M*** operator.

7.1.2. Hand preference (Hp)

The next operator of interest refers to handedness (hand dominance or hand preference) since users may prefer to interact using either their dominant (DH) or non-dominant hand (NDH). In this sense, the following question arises: Is touchless interaction natural enough to be used with the preferred and non-preferred hand in a similar way? Answering this question should lead to determine whether there is a difference between hands, and in which level both hands differ.

In general, prior research suggests there is a difference between hands using a computer, but the NDH can be as good as the DH for some tasks such as pointing or motion (Kabbash et al., 1993). Actually, Peters and Ivanoff demonstrated the difference between hands is small using a mouse by analyzing several performance metrics (Peters and Ivanoff, 1999). Furthermore, though performance with the DH can be better than with the NDH in tasks that require visual control, there are occasions on which subjects may perform some tasks better with their NDHs, which could be due to cerebral organization (Hoffmann, 1997).

These previous findings are similar for touchless interaction in some degree as noted by Jude et al. (Jude et al., 2014). They calculated the increase in movement time (MT, the time to reach a target) for pointing tasks and found there is 11% degradation between hands. This work could be an initial step towards introducing an operator for hand preference, but there is a limitation: the authors only used hovering gestures to make selections. In other words, the computed degradation may apply only to pointing tasks based on Fitts' Law and hover gestures. Consequently, it is necessary to verify whether this difference is the same using other gesture strokes.

In addition, it is insufficient to analyze several strokes to introduce an $\boldsymbol{Hp}$ operator because other aspects should be considered. We refer to the relation between reaction time and handedness. Though it is expected "there is hardly any difference between the simple reaction times of the dominant and non-dominant hand" (Rosenbaum, 2009)(p. 280), Peters and Ivanoff reported shorter times for the dominant hand (Peters and Ivanoff, 1999) (in fact, the reported difference is, at most, in the order of 10%). Thus, reaction time should be taken into account in the handedness analysis, especially trying to make a relation with mental operations previously described.

7.1.3. *Other stroke operators*

THGLM includes a set of S-phrase operators that were selected by performing a literature search (Erazo and Pino, 2015), which is also consistent with the gestures set proposed by Walter et al (Walter et al., 2014) to select items on interactive public displays. However, Erazo and Pino (Erazo and Pino, 2015) only proposed the values for some S-phrase operators, and as they noted, these operators may be used to analyze some applications functionalities, though there are other options that have not been included in the model yet. Namely, two options are pulling (move the hand towards the back) and waving (wave the hand) gestures. Other S-phrase operators may be included, but it is beyond the scope of this work because the main goal is to illustrate how to extend THGLM by introducing several operators.

7.2. *User study for time measurements*

Given the aforementioned candidate operators, we conducted an experiment to estimate the values of the selected operators. First, we decided to use hand preference as a baseline to design the experiment. In other words, the experiment was designed to analyze hand preference but taking into account the operators of interest to make the needed measures. On the one hand, we analyzed two types of simple decision tasks, which correspond to the $\boldsymbol{M_S}$ and $\boldsymbol{M_P}$ operators, but the definition of $\boldsymbol{M_P}$ was modified slightly. The user matches the used hand to the stimulus for our physical matching, but the stimulus is presented on the left or right sides of the screen. In other words, the user employs the hand that corresponds to the same side of the screen where the stimulus appears. These two operators were considered as a starting point towards the analysis of diverse interactions requiring user attention and cognition on UIs based on THG. On the other hand, the time to perform several strokes using both hands was measured, including *pulling* ($\boldsymbol{U}$ operator). This allowed us to introduce a new S-phrase operator and determine the consistency with some previously found values. As a result, four new operators are available to be included in the model.

7.2.1. *Method*

7.2.1.1. Procedure and tasks. In general, the experiment consisted on performing several gestures using both hands in two phases. Participants had to execute twelve times a gesture using only one hand in the first phase (P1). Next, they repeated the process using the other hand. This process was repeated for all four gesture strokes (pull, tap, grip, and release), which were selected from (Erazo and Pino, 2015; Walter et al., 2014). Though we could use other gestures, we selected those ones considered representative/adequate for this experiment. For example, Hold was considered not adequate for this case due to its nature; i.e., "hold" a hand a constant time on a button to be recognized. The same gestures were performed in the second phase (P2), but participants used both hands; that is, they randomly selected the hand they had to use to execute the gesture. Each gesture was repeated twelve times using each hand in a way similar to the first phase.

Taking into account handedness plays a special role in the experiment, we first asked participants to accomplish an Edinburgh Handedness Inventory questionnaire for hand dominance assessment (Oldfield, 1971). They answered questions about their degree of preference toward a hand to do ten common tasks such as writing, drawing, etc.

The experiment started with written instructions about the tasks participants had to perform. The instructions consisted on an explanation about the way the application worked, the gestures to execute and the way to do it. When the participant was ready to start, we asked him/her to adopt the right position before starting a practice session. Four trials were performed (with each gesture) as part of the practice session in order to allow participants knowing the software and the gestures. This practice was only performed during one phase (the corresponding one according to task order) for all gestures since participants knew the protocol when it was the turn of the other phase.

The task consisted on performing a gesture to select a square button centered in the screen in phase P1. A trial began with the presence of a beige button which changed to gold when the cursor was placed over it. Next, the button turned red (*preparatory stimulus* (Jensen, 2006)) when the software detected the participant's hand was still. This change alerted the subject to the impending reaction stimulus and started the *preparatory interval* (Jensen, 2006). The preparatory interval, which is the time between the preparatory stimulus and the reaction stimulus, was in the range 1–3 s according to Jensen's (Jensen, 2006) suggestion. Participants were instructed to avoid moving the hand until the button turned green, which is the *reaction stimulus* (Jensen, 2006). The participant had to execute the gesture as quickly as possible after the reaction stimulus appeared and trying to balance speed and precision. If the application interpreted the gesture as correct, then the percentage of progress was displayed. If not, then the participant had to repeat the trial. Next, the same process was repeated employing the other hand. The task continued with next gesture after having a short rest. When the participant accomplished the four gestures using both hands, the second phase started.

Phase P2 was similar to phase P1, but two buttons, two cursors and both hands were used at the same time instead of one. Each button appeared at the center of the left and right half of the screen, separated 280 pixels horizontally, and at the same height. A button could only be selected using one cursor as each cursor was linked to each hand. Also, both buttons changed their colors as described for phase P1, but only one button turned green in each trial. In other words, if the left button color changed to green, then the participant had to use the left hand to perform the gesture, and likewise, use the right hand if the button on the right turned green.

Furthermore, we collected some data about demographics, computer use, and THG experience at the end of the experiment. The whole experiment lasted 50 min on the average.

7.2.1.2. Apparatus. The hardware setup was the same used in the experiment to validate the model with UI designers (described in section 6.1.2), but a new custom application was developed. The application interface is inspired on (MacKenzie, 2013), but making the needed adaptations as that application is intended for keystrokes whereas our application is based on THG. Thus, the application controls stimulus presentation (change of colors) and times (delays) as needed, and logged the required data.

Leap Motion was used as input device to track user hands and recognize gestures. This decision was made taking into account the high sensor accuracy (below 0.2 mm) stated by the manufacturer and confirmed in studies that evaluated the sensor (Weichert et al., 2013; Guna et al., 2014). These advantages allow the application to detect when the hand (or both hands) is not moving to enter into preparatory interval. Also, hand tremor, which was set to 0.2 mm for young and healthy people according to previous studies (Sturman et al., 2005; Weichert et al., 2013), was used to avoid detecting false movements. On the other hand, hand positions and thresholds were used to recognize gestures as follow: move the hand forward or backward 15 cm to detect tapping or pulling respectively; hand open or close at 95% for gripping and releasing (according to Leap.Hand.GrabStrength property provided with LM SDK). In addition, while users were performing a gesture, the cursors turned blue varying the color intensity according to the gesture progress (i.e., from light blue to dark blue) with the aim of providing feedback. The same feedback was provided for all gestures to prevent a possible feedback effect.

As mentioned above, the graphical interface consisted of one or two buttons and cursors (depending on the phase). Button sizes were set to 120pixels per side (consistent with (Jude et al., 2014)) and have neither labels nor images. The cursors were white circles, with black border, 50pixels diameter, and controlled with hand movements. The cursors were only shown inside a white rectangle of 800 × 600 pixels. This rectangle was mapped to the interaction space in which subjects move their hands. The background of the remaining area was set to black as suggested by participants in pilot trials.

7.2.1.3. Participants. The participants in the experiment were University students (20 in total, 19 right-handed, 10 female, 13 undergraduate students, aged between 18 and 37 years) invited by email and social networks. They self-declared to use computers at least 10 h per week, and thirteen had some basic experience on touchless interaction, such as using Wii remote or Microsoft Kinect for playing games. The other seven participants had no prior touchless interaction experience. The participants were not paid for their participation and signed a written informed consent before starting the study.

7.2.2.4. Design. A within-subjects design was used where each participant performed 96 gesture-trials per phase in total (4 gesture strokes × 2 hands × 12 trials). The initial hand and phase were counterbalanced across participants; Latin squares were used to determine gestures order; and the preparatory interval was randomized to prevent participants from anticipating the onset stimulus (MacKenzie, 2013) (p. 57). Moreover, we followed Kosinski and Cummings' suggestion regarding the minimum reaction times per person and per treatment to be collected (Kosinski and Cummings, 1999).

Given this scenario, we gathered data of reaction time (RT) and stroke time (ST). RT is the delay between a fixed (or reaction) stimulus and the initiation of a response (e.g., a detectable movement) (Jensen, 2006; MacKenzie, 2013). In some cases, it is also named response time, but the latter should be preferred in experiments in which speed is neither emphasized nor mentioned in instructions (Jensen, 2006). In this study, RT is the elapsed time since the button turns green and a participant starts moving the hand to perform a gesture. On the other hand, ST is the interval between participants start performing the gesture until the gesture is recognized.

7.2.2. Results

The collected data was used to estimate the values of the four selected operators: RT data was utilized to estimate the values of the $\boldsymbol{M_S}$ and $\boldsymbol{M_P}$ operators; ST and RT data were used to analyze the $\boldsymbol{Hp}$ operator value; and the $\boldsymbol{U}$ operator value was obtained using the times of pulling gestures. Table 2 summarizes the obtained times.

The first operator in Table 2, *Pulling*, is an S-phrase operator that belongs to the group of expressive operators. Its value was computed as the period of time since a participant started to move the hand towards the back until the hand was moved 15 cm (i.e., using ST). Values of trials performed with the DH in both phases were used to compute the stroke time because the difference between phases was not statistically significant ($F_{1,19} = 0.709$, ns).

$\boldsymbol{M_S}$ and $\boldsymbol{M_P}$ operators were estimated using the data from both phases, P1 and P2, respectively. Their values correspond to the participants' reaction times; that is, the period of time between the response stimulus appeared until a participant started to move the hand (i.e., execute the gesture). In general, the analysis was performed following the general recommendations for analyzing RT data described in (Jensen, 2006; Whelan, 2008), such as cutoff values (e.g., exclude values greater than three standard deviations above the mean), use arithmetic mean, etc. The analysis of variances using these values revealed the main effect of gesture strokes on RT was statistically significant in both phases ($F_{3,57} = 3.755$, $p < 0.05$ in P1; $F_{3,57} = 5.406$, $p < 0.05$ in P2), but differences between gestures were small (less than 5% on the average for both phases). Moreover, the main effect of used hand was not statistically significant in both phases ($F_{1,19} = 1.291$, $p > 0.05$ in P1; $F_{1,19} = 0.387$, ns in P2), and no significant gesture × hand interaction effects were found ($F_{3,57} = 0.701$, ns in P1; $F_{3,57} = 1.013$, $p > 0.05$ in P2). Consequently, we decided to keep only one value per operator in order to not increase the model complexity.

Given that other authors have analyzed MT using THG and based on Fitts' Law, we concentrated just on RT and ST. Similarly to RT, the analysis of variances revealed the main effect of gesture on ST was statistically significant in both phases ($F_{3,57} = 6.160$, $p < 0.01$ in P1; $F_{3,57} = 4.761$, $p < 0.01$ in P2), whereas the main effect of hand was not significant also in both phases ($F_{1,19} = 0.209$, ns in P1; $F_{1,19} = 0.184$, ns in P2). Likewise, there were no significant gesture × hand interaction effects in both cases ($F_{3,57} = 0.344$, ns in P1; $F_{3,57} = 1.137$, $p > 0.05$ in P2). These results suggest there is no difference between hands when the analyzed strokes are produced. Thus, we infer the difference between hands is present during the movement phase to reach the target (i.e., in the pointing phase). In other words, the degradation between DH and NDH should be applied to the $\boldsymbol{P}$ operator and not to the $\boldsymbol{T}$, $\boldsymbol{G}$, $\boldsymbol{R}$, and $\boldsymbol{U}$ operators. Nevertheless, we did not analyzed MT, and hence, it is necessary to use some related work. Specifically, Jude et al (Jude et al., 2014) found the degradation in MT to be about 11%. This value could be

Table 2
Overview of the proposed times for the new operators. * Only for *P* operator (see text for details).

Operator	Group (type)	Time (in seconds if not specified)	SD (s)
U, Pulling	Expressive – S-phrase	0.941	0.121
***M*$_S$**, Simple reaction	General	0.375	0.076
***M*$_P$**, Physical matching	General	0.388	0.065
Hp, Hand preference	Movement	11%*	N/A

used for the ***Hp*** operator. Consequently, if the analysis considers the task will be performed with the NDH, then the ***P*** operator will change to "***Hp P***", that is to say, 1.11****P*** (or 1.161s using the constant value suggested in Table 1, which is lower than the average value reported in (Jude et al., 2014), but it falls into the computed intervals).

Finally, we made a comparison between the current and the previously proposed times (Table 1) for the ***T***, ***G***, and ***R*** operators taking into account that the corresponding strokes were used in the present experiment. Both set of values are very similar as shown in Fig. 11 (the mean difference between the three operators is 6%), though in fact, all the differences between each pair of values were not statistically significant. These results allow us being more confident about the values for these operators.

7.2.3. Using the new operators

After estimating the new operator times, we decided to go beyond and perform a short test with the aim of using some of these operators. One application previously used, Gester, was chosen to perform the test. Five participants that took part in the experiment interacting with Gester using the DH were still available. They then performed the same task but using the NDH. On the other hand, the same procedure was followed to produce the numeric predictions. However, the ***Hp*** operator was used on this occasion. The resulting prediction error using the observed and estimated values is similar to the one obtained for the DH (about 23%). This result allows inferring the ***Hp*** operator worked well, but the error remains higher than expected.

As suggested above, the task using Gester requires users react quickly when the next letter (target) is shown in order to "catch" it. This means that the required mental act could be a simple reaction (***M*$_S$** operator) and not a mental preparation (***M*** operator) that subsumes several cognitive processes into one. Therefore, this task should include an ***M*** for selecting the "start button" and one ***M*$_S$** for each letter that users must "catch".

In fact, the predictions doing the suggested changes are better than the previous ones. The prediction error decreased to about 11% for both the DH and NDH. This analysis and results constitute evidence that the ***M*** should be updated according to MacKenzie's suggestion (MacKenzie, 2013).

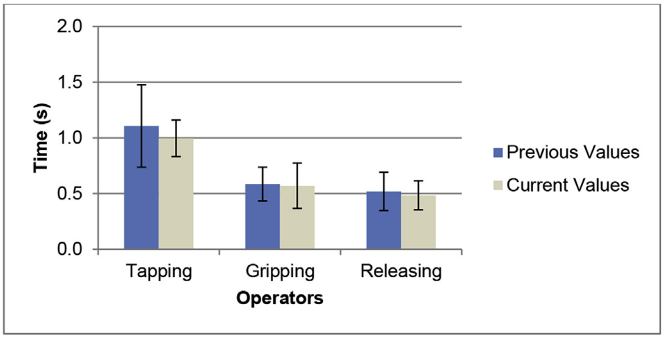


Fig. 11. Comparison of current and previous times for three operators.

7.3. Further model extensions

Four operators have been proposed with the aim of extending THGLM, but more operators can be included. However, there is a trade-off concerning the number of operators because the model may turn complex if it has many operators (Holleis, 2009)(p. 53). Conversely, having a too small number of operators will decrease the number of application functionalities that could be analyzed. Bearing in mind this suggestion, some operators that could possibly be included in the future are discussed briefly in this section. These operators come from literature in which some KLM extensions have been described. Of course, this list is not exhaustive and other operators could be taken into account, especially those concerning the group of expressive operators.

Referring to S-phrase operators, Table 1 includes an operator for swiping (as well as the corresponding ones to prepare for performing a swipe) according to Erazo and Pino's proposal (Erazo and Pino, 2015). In fact, three operators were distinguished: an average swipe (***S***), horizontal swipe (***Sh***) and vertical swipe (***Sv***). Going beyond and using the data of that study, the analysis of variances reveals the difference between horizontal swipes from right to left and vice versa is statistically significant ($F_{1,35} = 9.190$, $p < 0.01$), as well as between vertical swipes from top to bottom and vice versa ($F_{1,35} = 11.053$, $p < 0.01$). This analysis suggests two operators should be used for horizontal swipes and two for vertical swipes instead of one for each kind of swipes. This is an example of the aforementioned trade-off: a decision should be made between using a single value (for any swipe type), two values (for ***Sh*** and ***Sv***) or four values (one for each swipe direction). Given that acceptable predictions were reached when swipes were employed (Erazo and Pino, 2015), we suggest following the same idea as shown in Table 1 (i.e., using only one or two values) to keep the model simplicity.

There are several models developed for other interactions that may be adapted and included as movement operators. The first model is Fitts' Law (Fitts, 1954), whose usefulness for THG was discussed in (Erazo and Pino, 2015). Although a constant value has been used in the experiments to validate THGLM, Fitts' Law could be used to estimate the time to reach a target as demonstrated in several works (Schwaller and Lalanne, 2013; Pino et al., 2013; Polacek et al., 2012; Zeng et al., 2012; Sambrooks and Wilkinson, 2013; Jude et al., 2014). Another option could be the use of ballistic models such as (Hoffmann and Gan, 1988). Steering Law (Accot and Zhai, 1997) is another model that may be evaluated using THG and possibly be extended. It is based on Fitts' Law that allows predicting the speed and total time to navigate through a two-dimensional tunnel in trajectory-based tasks. The user's task consists of traveling from one end of the path to the other one as quickly as possible, without touching the boundaries of the tunnel. The potential HCI applications of this model are device comparison and menu design. As a consequence, if the use of this model is verified and/or adapted to be included as an operator in THGLM, it could be used to forecast performance to navigate vertical or horizontal menus, drag an element through a "tunnel", etc.

Regarding the ***M*** operator, we have introduced two additional operators that can complement or replace this single operator. We have also mentioned above other candidate operators for mental acts, but they have not been studied because it is beyond the scope of this work. However, it is worth to mention that $\boldsymbol{M_C}$ could be studied on the basis of another previous and widely used model, Hick-Hyman Law (Hick, 1952; Hyman, 1953). This model forecasts the time a user needs to make a decision when s/he has to choose the correct one from some simple options. In the context of THGLM, this model might be adapted to estimate the time a person needs to make the decision in general, or particularly, to choose the right gesture to execute a command and/or activate some option of the application.

Besides, other candidate operators that could be included as general operators have also been suggested in previous works. For example, Holleis (2009) proposes to consider age, illumination, scroll, etc. as further model extensions for advanced mobile phone interactions. (He also mentions other operators that might not be applicable to THG.) Despite Holleis includes the corresponding values obtained from related works, these values should be verified with THG. Finally, the Power Law of Practice (Card et al., 1983)(p. 27) may be used to model how practice can change the time to perform THG and complete tasks using NUIs.

8. General discussion

In this work, we have provided further details and validation of THGLM, which is a predictive model to quantitatively evaluate UIs based on THG. The results confirm the validity of the model, and thus it is important to discuss briefly some assumptions, advantages and limitations.

First, THGLM assumes additivity of time elements. It is a simplification because it does not consider possible interrelationships between gesture strokes/holds. Nevertheless, the validity of the additivity assumption has been demonstrated in various KLM extensions (e.g., (Holleis et al., 2007; Luo and John, 2005; Lee et al., 2015)). Moreover, the gestures included in THGLM are basically the same considered in Predetermined Motion Time Systems (PMTS) (Genaidy et al., 1989), used for many years by engineers for time estimation in industrial tasks with similar additivity assumption.

Like KLM, THGLM is relatively easy to use but has some limitations. In particular, the method to execute tasks must be completely specified at the level of gestures. Therefore, a set of operators to describe those tasks in terms of THG is needed. Nonetheless, although there are several gestures which are “universally” used, we have not known a standard concerning gestures to be employed in NUIs. This is why Erazo and Pino (Erazo and Pino, 2015) decided to perform a systematic bibliographic review to find the candidate gestures to be included as operators. They chose the gestures most frequently used in related works, but they did not include all options. Thus, we demonstrated here the model is extensible by adding new operators based on previous suggestions. Although all the selected operators may be used to analyze various application functionalities, more operators can be easily included by estimating the corresponding time, or finding the equations to compute it instead (Erazo and Pino, 2015).

Unlike the original KLM, which assumes tasks are performed by expert users, THGLM has been developed for novice users. The main motivation is that today there are too few expert users on UIs based on THG. Actually, many users approach this type of interfaces for the first time (Walter et al., 2014). Despite we have provided some suggestions to use the model to make predictions for expert users, the operator values could be updated in the future when more expert users become available.

Another limitation lies in being limited to error-free execution. Participants made a number of errors when performing the tasks, mainly due to the learning process. Thus, we proceeded in a way similar to Card et al. (Card et al., 1980), i.e. “ignoring the tasks containing errors and only predicting the error-free tasks”. The model performance would decrease if tasks with significant errors were used.

The model performance may also decrease depending on users' characteristics especially because the operator values were estimated by involving healthy young adults. For instance, prediction error could worsen if applications are used by children, elderly people, impaired people, etc. Therefore, the model may be generalized by analyzing differences and/or estimating operator values with the participation of other types of users as future work (e.g., the aforementioned operator for age).

Other aspect that Erazo and Pino (Erazo and Pino, 2015) discussed in the initial model formulation refers to the variability of the operator times. They reported a relatively high coefficient of variation (CV) of gesture production times (about 30%). Though we reached smaller CVs for the new operators, they remained near 20%. However, these values are comparable to previously reported ones (e.g., 31% in (Card et al., 1980)).

Although current devices allow detecting and tracking human body and hand fingers, this interaction type has not been considered. Actually, the model only allows forecasting times of tasks where gestures are performed with one hand. Nowadays, the model includes an operator for hand preference to allow analyzing tasks assuming users will use either the left or the right hand. However, this operator is not enough to model two-handed interactions. (In fact, the use of this operator should be verified for other operators such as *drawing*). Despite the model may be extended, there are several aspects that should be studied in order to extend it to support bimanual interactions. For example, one hand could be used to perform gestures while the other hand is used as reference, or both hands could be used to perform gestures symmetrically or asymmetrically. A good starting point towards this goal may be the analysis of previous models for bimanual tasks such as (Guiard, 1987; Ruiz et al., 2008).

There are several areas to which the model might be applied in the future. A first one is related to the rehabilitation of people with upper limb dysfunctions. In this scenario, THGLM may be used not only to forecast performance time after computing the proper values. The model might also be adapted to estimate time that patients require in executing a routine and defining new operators, according to their limitations. Subsequently, these values may be used to encourage patients to continue training (e.g. achieve the value established as a goal) (Erazo et al., 2014). Other possible application or extension of the model refers to virtual/augmented/mixed reality based on THG. For instance, applications in which users utilize a head-mounted display and THG for interacting (e.g., (Kohli, 2013)). The model might be applied to analyze UI designs for this kind of applications after performing the corresponding studies.

Going beyond the advantages and limitations of THGLM, it is important to notice that the model only addresses a single aspect of performance: time. Although performance time is commonly used to evaluate interfaces (MacKenzie, 2013), and particularly using model-based evaluation, there are other dimensions of performance. Fatigue is one of these dimensions that should be considered to design UIs based on THG. Though some methods have been proposed to quantify fatigue (Hincapié-Ramos et al., 2014), they require user participation. Other dimensions that could be considered are errors, learning, etc. The analysis of these and other aspects can be very useful to get a good product, but the main utility of THGLM is at early design stages before implementing a prototype and collecting data from users to apply other metrics.

9. Conclusions

This work describes, verifies and extends THGLM which is considered the first comprehensive model to quantitatively evaluate UIs based on THG (Erazo and Pino, 2015). The model predicts the time to do a task given the method in an acceptable way according to the initial validation (Erazo and Pino, 2015). Since the initial version of the model had some limitations, we started completing that initial formulation by describing the needed rules and suggestions to produce quantitative predictions. Namely, we updated the heuristics rules and recommendations for including mental operators, and adapted a procedure to use THGLM. Then, we used these enhancements to study the model performance in further detail.

The empirical validation confirms the quality of THGLM to forecast performance time. The model reached a prediction error (RMSE) of 12%, while the error obtained by Card et al. (Card et al., 1980) for the original KLM is 21%. The model performance is also confirmed by the high relationship between estimated and observed times ($R^2 > 0.9$). These results confirms hypothesis *H1*.

Likewise, the results from the study to analyze UI designs validate hypothesis *H2*. The comparison between predicted and observed values for the three design options for the proposed application continued acceptable; i.e., the percentage of error was lower than 21% in all cases. More important, the comparison performed to select the best design option gave the same result using the model and observing users. Thus, we conclude THGLM can be employed as a design tool based on both these results and the ones obtained in a later study.

As the model should become useful for UI designers, its validity as a design tool was confirmed by conducting a study with the participation of independent designers. Notably, designers' predictions stayed stable in comparison to the researcher's ones, confirming hypothesis *H3*. The designers were able to produce numerical predictions for all required tasks with no problems and in short periods despite having no previous experience on model-based evaluation. Hence, THGLM can be used to analyze UI designs in a relatively easy way.

In addition, we have demonstrated the model is extensible by introducing and using several new operators. This means that other operators can be included in the future to expand the possibilities offered by THGLM.

Regarding the general hypothesis, we observe it is validated for the three specific hypotheses. Therefore, UI designers and/or researchers have available a model that could be used without undertaking time-consuming and resource-intensive ad-hoc experiments. Thus, the model should be useful for designers to be able to develop good software products using THG.

Finally, in spite of the good results and advantages of THGLM, it has several constraints that must be taken into account when it is used. These restrictions mainly refer to users' characteristics and the fact that the model only allows analyzing one-handed interactions. Therefore, the model could be improved by adding new operators to cover a wide range of users and conditions, as well as by extending it to two bimanual interactions.

Acknowledgments

This work was supported by “Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación” (Convocatoria Abierta 2011), Ecuador; and NIC Chile (Dept. of Computer Science, Universidad de Chile).

References

Accot, J., Zhai, S., 1997. Beyond Fitts' law: models for trajectory-based HCI tasks. In: Proceedings of CHI 1997, pp. 295–302.

Bachynskyi, M., Oulasvirta, A., Palmas, G., Weinkauf, T., 2014. Is motion-capture-based biomechanical simulation valid for HCI Studies? Study and implications. In: Proceedings of CHI 2014, pp. 3215–3224.

Bachynskyi, M., Palmas, G., Oulasvirta, A., Weinkauf, T., 2015. Informing the design of novel input methods with muscle coactivation clustering. ACM Trans. Computer-Human Interact. (TOCHI) 21 (6).

Barclay, K., Wei, D., Lutteroth, C., Sheehan, R., 2011. A quantitative quality model for gesture based user interfaces. In: Proceedings of OzCHI 2011. ACM, pp. 31–39.

Cao, X., Zhai, S., 2007. Modeling human performance of pen stroke gestures. In: Proceedings of CHI 2007, pp. 1495–1504.

Card, S., Moran, T., Newell, A., 1980, July. The keystroke-level model for user performance time with interactive systems. Commun. ACM 23 (7), 396–410.

Card, S., Moran, T., Newell, A., 1983. The Psychology of Human-computer Interaction. L. Erlbaum Associates.

de la Barré, R., Chojecki, P., Leiner, U., Mühlbach, L., Ruschin, D., 2009. Touchless interaction-novel chances and challenges. In: Jacko, J.A. (Ed.), Human-computer Interaction, Part II, HCII 2009, LNCS 5611. Springer, pp. 161–169.

Erazo, O., Pino, J.A., 2015. Predicting task execution time on natural user interfaces based on touchless hand gestures. In: Proceedings of IUI 2015. ACM, pp. 97–109.

Erazo, O., Pino, J.A., Antunes, P., 2015. Estimating production time of touchless hand drawing gestures. In: Abascal, J., et al. (Eds.), INTERACT 2015, Part III, LNCS 9298, pp. 552–569.

Erazo, O., Pino, J.A., Pino, R., Fernández, C., 2014. Magic mirror for neurorehabilitation of people with upper limb dysfunction using Kinect. In: IEEE (Ed.), Proceedings of HICSS 2014, pp. 2607–2615.

Fitts, P.M., 1954, June. The information capacity of the human motor system in controlling the amplitude of movement. J. Exp. Psychol. 47 (6), 381–391.

Gallo, L., Placitelli, A.P., Ciampi, M., 2011. Controller-free exploration of medical image data: experiencing the Kinect. In: 24th International Symposium on Computer-based Medical Systems, pp. 1–6.

Genaidy, A.M., Mital, A., Obeidat, M., 1989. The validity of predetermined motion time systems in setting production standards for industrial tasks. Int. J. Industrial Ergonomics 3 (3), 249–263.

Guiard, Y., 1987. Asymmetric division of labor in human skilled bimanual action: the kinematic chain as a model. J. Mot. Behav. 19 (4), 486–517.

Guna, J., Jakus, G., Pogačnik, M., Tomažič, S., Sodnik, J., 2014. An analysis of the precision and reliability of the Leap motion sensor and its suitability for static and dynamic tracking. Sensors 14 (2), 3702–3720.

Hick, W.E., 1952. On the rate of gain of information. Q. J. Exp. Psychol. 4 (1), 11–26.

Hincapié-Ramos, J.D., Guo, X., Moghadasian, P., Irani, P., 2014. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In: Proceedings of CHI 2014. ACM, pp. 1063–1072.

Hinrichs, U., Carpendale, S., Valkanova, N., Kuikkaniemi, K., Jacucci, G., Vande Moere, A., 2013. Interactive public displays. Comput. Graph. Appl. 33 (2), 25–27.

Hoffmann, E.R., 1997. Movement time of right-and left-handers using their preferred and non-preferred hands. Int. J. Industrial Ergonomics 19 (1), 49–57.

Hoffmann, E., Gan, K., 1988. Directional ballistic movement with transported mass. Ergonomics 31 (5), 841–856.

Holleis, P., 2009. Integrating Usability Models into Pervasive Application Development. PhD Thesis. Ludwig - Maximilians-Universität München, Germany.

Holleis, P., Otto, F., Hussmann, H., Schmidt, A., 2007. Keystroke-level model for advanced mobile phone interaction. In: Proceedings of CHI 2007, pp. 1505–1514.

Hyman, R., 1953. Stimulus information as a determinant of reaction time. J. Exp. Psychol. 45 (3), 188–196.

Isokoski, P., 2001. Model for unistroke writing time. In: Proceedings of CHI 2001. ACM, pp. 357–364.

Jagodziński, P., Wolski, R., 2015. Assessment of application technology of natural user interfaces in the creation of a virtual chemical laboratory. J. Sci. Educ. Technol. 24 (1), 16–28.

Jensen, A.R., 2006. Clocking the Mind: Mental Chronometry and Individual Differences. Elsevier.

John, B.E., Prevas, K., Salvucci, D.D., Koedinger, K., 2004. Predictive human performance modeling made easy. In: Proceedings of CHI 2004, pp. 455–462.

Jude, A., Poor, G.M., Guinness, D., 2014. An evaluation of touchless hand gestural interaction for pointing tasks with preferred and non-preferred hands. In: Proceedings of NordiCHI 2014, pp. 668–676.

Kabbash, P., MacKenzie, I.S., Buxton, W., 1993. Human performance using computer input devices in the preferred and non-preferred hands. In: Proceedings of INTERACT 1993 and CHI 1993, pp. 474–481.

Kendon, A., 2004. Gesture units, gesture phrases and speech. In: Gesture: Visible Action as Utterance. Cambridge University Press, pp. 108–126.

Kieras, D., 2003. Model-based evaluation. In: The Human-computer Interaction Handbook. Lawrence Erlbaum Associates, New Jersey, pp. 1191–1208.

Kieras, D., 2001. Using the Keystroke-level Model to Estimate Execution Times. University of Michigan, vol. 2001. Retrieved from. http://www-personal.umich.edu/~itm/688/KierasKLMTutorial2001.pdf. http://www-personal.umich.edu/~itm/688/KierasKLMTutorial2001.pdf.

Kita, S., Van Gijn, I., Van der Hulst, H., 1998. Movement phases in signs and Co-

Speech gestures, and their transcription by human coders. In: Gesture and Sign Language in Human-computer Interaction. Springer Berlin Heidelberg, pp. 23–35.

Kohli, L., 2013. Warping virtual space for low-cost haptic feedback. In: Proceedings of SIGGRAPH Symposium on I3D 2013, p. 195.

Kosinski, B., Cummings, J., 1999. The scientific method: an introduction using reaction time. In: Tested Studies for Laboratory Teaching (ABLE Proceedings) 25.

Kurtenbach, G., Hulteen, E.A., 1990. Gestures in human-computer communication. In: The Art of Human-computer Interface Design. Addison-Wesley, pp. 309–317.

Lee, A., Song, K., Ryu, H.B., Kim, J., Kwon, G., 2015. Fingerstroke time estimates for touchscreen-based mobile gaming interaction. Hum. Mov. Sci. 44, 211–224.

Lubos, P., Bruder, G., Steinicke, F., 2014. Analysis of direct selection in head-mounted display environments. IEEE Symposium 3D User Interfaces 11–18.

Luo, L., John, B.E., 2005. Predicting task execution time on handheld devices using the keystroke-level model. Ext. Abstr. CHI 2005 1605–1608.

MacKenzie, I.S., 2003. Motor behavior models for human-computer interaction. In: HCI Models, Theories, and Frameworks: toward a Multidisciplinary Science. Morgan Kaufmann, San Francisco, pp. 27–54.

MacKenzie, I.S., 2013. Human-Computer Interaction: an Empirical Research Perspective. Morgan Kaufmann.

MacKenzie, I.S., Teather, R.J., 2012. FittsTilt: the application of Fitts' law to tilt-based interaction. In: Proceedings of NordiCHI 2012, pp. 568–577.

McNeill, D., 1992. Guide to gesture classification, transcription, and distribution. In: Hand and Mind: what Gestures Reveal about Thought. The University of Chicago Press, pp. 75–104.

Müller-Tomfelde, C., 2007. Dwell-based pointing in applications of human computer interaction. In: Baranauskas, C., et al. (Eds.), Human-computer Interaction - INTERACT 2007, LNCS 4662, Part I. Springer Berlin Heidelberg, pp. 560–573.

Neff, M., Kipp, M., Albrecht, I., Seidel, H.P., 2008. Gesture modeling and animation based on a probabilistic recreation of speaker style. ACM Trans. Graph. (TOG) 27 (1).

Nielsen, M., Störring, M., Moeslund, T.B., Granum, E., 2004. A procedure for developing intuitive and ergonomic gesture interfaces for HCI. In: Camurri, A., Volpe, G. (Eds.), GW 2003, LNAI 2915. 2915. Springer Berlin Heidelberg, pp. 409–420.

Norman, D., 2010, May. Natural user interfaces are not natural. Interactions 17 (3), 6–10.

Nunes, J.F., Moreira, P.M., Tavares, J.M., 2015. Human motion analysis and simulation tools: a survey. In: Handbook of Research on Computational Simulation and Modeling in Engineering.

Oldfield, R.C., 1971. The assessment and analysis of handedness: the Edinburgh inventory. Neuropsychologia 9 (1), 97–113.

Peters, M., Ivanoff, J., 1999. Performance asymmetries in computer mouse control of right-handers, and left-handers with left-and right-handed mouse experience. J. Mot. Behav. 31 (1), 86–94.

Pino, A., Tzemis, E., Ioannou, N., Kouroupetroglou, G., 2013. Using Kinect for 2D and 3D pointing tasks: performance evaluation. In: Kurosu, M. (Ed.), Human-computer Interaction, Part IV, HCII 2013, LNCS 8007. Springer Berlin Heidelberg, pp. 358–367.

Polacek, O., Klíma, M., Sporka, A.J., Zak, P., Hradis, M., Zemcik, P., Procházka, V., 2012. A comparative study on distant free-hand pointing. In: Proceedings of EuroiTV 2012, pp. 139–142.

Rosenbaum, D.A., 2009. Human Motor Control. Academic press/Elsevier, San Diego.

Ruiz, J., Bunt, A., Lank, E., 2008. A model of non-preferred hand mode switching. In: Proceedings of Graphics Interface 2008, pp. 49–56.

Sambrooks, L., Wilkinson, B., 2013. Comparison of gestural, touch, and mouse interaction with Fitts' law. In: Proceedings of OzCHI 2013, pp. 119–122.

Schwaller, M., Lalanne, D., 2013. Pointing in the air: measuring the effect of hand selection strategies on performance and effort. In: Holzinger, A., et al. (Eds.), SouthCHI 2013, LNCS 7946, pp. 732–747.

Sridhar, S., Feit, A.M., Theobalt, C., Oulasvirta, A., 2015. Investigating the dexterity of multi-finger input for mid-air text entry. In: Proceedings of CHI 2015, pp. 3643–3652.

Stanton, N.A., Young, M.S., 2003. Giving ergonomics away? The application of ergonomics methods by novices. Appl. Ergon. 34 (5), 479–490.

Sturman, M.M., Vaillancourt, D.E., Corcos, D.M., 2005. Effects of aging on the regularity of physiological tremor. J. Neurophysiology 93 (6), 3064–3074.

Vatavu, R.D., 2012. User-defined gestures for free-hand TV control. In: Proceedings of EuroiTV 2012, pp. 45–48.

Walter, R., Bailly, G., Müller, J., 2013. StrikeAPose: revealing mid-air gestures on public displays. In: Proceedings of CHI 2013, pp. 841–850.

Walter, R., Bailly, G., Valkanova, N., Müller, J., 2014. Cuenesics: using mid-air gestures to select items on interactive public displays. In: Proceedings of MobileHCI 2014. ACM, pp. 299–308.

Webb, J., Ashley, J., 2012. NUI. In: Beginning Kinect Programming with the Microsoft Kinect SDK. Apress, pp. 170–172.

Weichert, F., Bachmann, D., Rudak, B., Fisseler, D., 2013. Analysis of the accuracy and robustness of the Leap motion controller. Sensors 13 (5), 6380–6393.

Whelan, R., 2008. Effective analysis of reaction time data. Psychol. Rec. 58 (3), 475–482.

Wigdor, D., Wixon, D., 2011. The natural user interface. In: Brave NUI World: Designing Natural User Interfaces for Touch and Gesture. Morgan Kaufmann, Boston, pp. 9–14.

Wobbrock, J.O., Wilson, A.D., Li, Y., 2007. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In: Proceedings of UIST 2007, pp. 159–168.

Wobbrock, J., Morris, M., Wilson, A., 2009. User-defined gestures for surface computing. In: Proceedings of CHI 2009. ACM, pp. 1083–1092.

Zeng, X., Hedge, A., Guimbretiere, F., 2012. Fitts' law in 3D space with coordinated hand movements. In: Proceedings of Human Factors and Ergonomics Society Annual Meeting. 56. SAGE Publications, pp. 990–994.