UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

A PREDICTIVE MODEL FOR USER PERFORMANCE TIME WITH NATURAL USER INTERFACES BASED ON TOUCHLESS HAND GESTURES

TESIS PARA OPTAR AL GRADO DE DOCTOR EN CIENCIAS
MENCIÓN COMPUTACIÓN

ORLANDO RAMIRO ERAZO MORETA

PROFESOR GUÍA:
JOSÉ A. PINO URTUBIA

MIEMBROS DE LA COMISIÓN:
NELSON BALOIAN TATARYAN
MIGUEL CARRASCO ZAMBRANO
RAMÓN HERVÁS LUCAS

Esta tesis ha sido parcialmente financiada por SENESCYT (Ecuador, Convocatoria Abierta 2011), Universidad Técnica Estatal de Quevedo (Ecuador) y NIC Chile (DCC, Universidad de Chile)

SANTIAGO DE CHILE
2016

Resumen

Las interfaces naturales de usuario (NUIs) basadas en gestos manuales sin contacto (THG) tienen ventajas sobre las interfaces de usuario (UIs) convencionales en varios escenarios. Sin embargo, éstas aún presentan problemas desafiantes por ser investigados, tales como su diseño y evaluación con la finalidad de obtener resultados satisfactorios. El enfoque clásico de la participación de usuarios para escoger gestos o analizar diseños de interfaces necesita ser complementado con evaluaciones predictivas para los casos en que los métodos basados en usuarios son inaplicables o costosos de realizar. En consecuencia, modelos de usuario cuantitativos son necesarios para efectuar esas evaluaciones.

Dado que la evaluación basada en modelos es importante en HCI y que los modelos disponibles son insuficientes para evaluar NUIs basadas en THG, esta tesis estudia modelos para interfaces de este tipo. Dos enfoques de modelamiento son abordados, aunque el enfoque principal está puesto en modelos predictivos. La tesis primeramente presenta un estudio para entender la articulación de gestos, el cual permitió derivar un modelo descriptivo sobre la concepción y producción de gestos de usuarios, y una taxonomía de gestos. Después, la tesis se concentra en (1) el análisis de la viabilidad de utilizar modelos cuantitativos existentes para abarcar THG; (2) la formulación de un modelo para predecir el tiempo de rendimiento para realizar tareas; y (3) la validación de este nuevo modelo. En todos los casos, el rendimiento de los modelos es estudiado de acuerdo a varias métricas usadas para hacer comparaciones con rangos típicos en el área.

Así, la contribución principal de esta tesis es un modelo para estimar el tiempo que un usuario necesita para hacer una tarea con una NUI basada en THG empleando su mano. El modelo propuesto, que es llamado THGLM, está basado en el modelo clásico KLM. Prescribe que los gestos manuales sean analizados de acuerdo a su estructura temporal; es decir, utilizando "gesture units". THGLM predice el tiempo de rendimiento en una forma aceptable (error de predicción = 12 %, $R^2 > 0.9$). Los experimentos realizados también confirman la utilidad del modelo para analizar y comparar diseños de interfaces. Si bien THGLM tiene ciertas limitaciones, tiene ventajas importantes tales como su relativa facilidad de usar y extender.

Más allá de las limitaciones intrínsecas de THGLM, éste debería ayudar en el diseño y evaluación de NUIs basadas en THG. Los diseñadores de UIs pueden predecir el tiempo para completar tareas sin la participación de usuarios, y luego, usar ese valor como métrica para analizar o evaluar una UI. Esta estrategia es especialmente útil en situaciones donde es difícil llevar a cabo pruebas con usuarios o como un paso preliminar a la evaluación de una interfaz. Por lo tanto, se espera que el modelo propuesto se convierta en una herramienta útil para diseñadores de software para realizar evaluaciones de usabilidad, mejorar diseños de interfaces, y desarrollar mejores aplicaciones de software utilizando gestos.

Abstract

Natural user interfaces (NUIs) based on touchless hand gestures (THG) have advantages over conventional user interfaces (UIs) 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. In consequence, quantitative user models are needed to perform those evaluations.

Given that model-based evaluation is important in HCI and available models are insufficient to evaluate NUIs based on THG, this thesis studies models for interfaces of this type. Two approaches to modeling are addressed, though the main focus is on predictive models. The thesis firstly presents a study to understand gesture articulation, which allowed deriving a descriptive model on user conception and production of gestures, and an embodied taxonomy of gestures. Next, the thesis concentrates on (1) the analysis of the feasibility of using existing quantitative models to encompass THG; (2) the formulation of a model to predict performance time in doing tasks; and (3) the validation of this new model. In all cases, model performance is studied according to several metrics used to make comparisons with typical ranges in the area.

Thus, the main contribution of this thesis is a model for estimating the time a user needs to do a task with a NUI based on THG using his/her hand. The proposed model, that is named THGLM, is based on KLM. It prescribes that hand gestures be analyzed looking at their temporal structure (i.e., using gesture units). THGLM forecasts performance time in an acceptable way (prediction error = 12 %, $R^2 > 0.9$). The performed experiments also confirm the model utility to analyze and compare interface designs. Even though THGLM has some limitations, it has important advantages such as its relative ease to use and extend.

Beyond the intrinsic limitations of THGLM, it should help in designing and evaluating NUIs based on THG. UI designers can predict required time to complete tasks without users' participation, and next, use that value as a metric to assess a UI. This approach is especially useful in situations where it is difficult to conduct tests with users or as a preliminary step to analyze or evaluate an interface. Therefore, we expect the proposed model becomes a useful tool for software designers to carry out usability assessments, improve interface designs, and develop better software applications using gestures.

Cinco años atrás inicié este "viaje".

Alegrías y tristezas he experimentado en el trayecto…

Mi HIJO llegó para alegrar nuestras vidas, aunque no estuve ahí para recibirlo.

Mi PADRE dejó este mundo, y tampoco pude estar presente para "despedirlo".

Pero en todo lo vivido, cerca o lejos, siempre pude contar con mi ESPOSA.

A ellos están dedicados estos años de trabajo y sus resultados.

Agradecimientos

Para iniciar este "viaje doctoral" y llegar al destino anhelado he requerido de la colaboración de diversas personas e instituciones, ya sea en el ámbito académico/profesional o en el personal. Por ello, estos sinceros agradecimientos son para todos aquellos que de una u otra manera pusieron su "granito de arena" para que yo logre mi objetivo, aunque por limitaciones de espacio me veo imposibilitado de nombrarlos a todos.

En primer lugar, la realización de esta tesis fue posible gracias a la beca adjudicada por SENESCYT (Convocatoria Abierta 2011) y a la licencia concedida por la Universidad Técnica Estatal de Quevedo (Ecuador). Soporte económico importante también fue otorgado por parte del Departamento de Ciencias de la Computación (DCC) de la Universidad de Chile, INRIA Lille (por medio de Prof. Laurent Grisoni), e Instituto Francés de Chile.

También quiero expresar mi enorme agradecimiento a todos quienes hacen el DCC, en todos sus niveles. No obstante, es imprescindible enfatizar esta gratitud hacia una persona, José A. Pino, mi profesor guía. Es indudable que siempre estaré agradecido con él por haberme ayudado y guiado con mucha paciencia y amabilidad, incluso desde antes de iniciar el doctorado. Asimismo, agradezco a los miembros de mi comisión por su valioso tiempo y comentarios.

Por otro lado, no cabe duda que los amigos juegan un rol destacable. Muchos de estos amigos me sorprendieron en más de una ocasión brindándome su ayuda desinteresada cuando más lo requería. Uno de ellos es Roberto Pico, quien junto con su esposa Silvia, "se la jugaron" confiando en mí al darme su aval para que pueda realizar el doctorado. De igual manera, mis agradecimientos a Martha Intriago (e hijos), Betty Chávez, Ariosto Vicuña, Orlando Brito (y familia), entre otros, por "darme la mano" especialmente en aquellos momentos difíciles. Además, estoy agradecido con mis amigos chilenos, como son Héctor Ferrada y la familia Sotomayor Gutiérrez, por el apoyo brindado.

Finalmente, la familia siempre jugará un rol preponderante en cualquier proyecto en que uno esté inmerso. Gracias a mi esposa por ser un pilar fundamental de este proyecto, ayudándome a afrontar mis problemas y lidiando con mis estados de ánimo negativos. Gracias a mi hijo por contagiarme su alegría. Gracias a mi madre por su apoyo incondicional. Gracias a todos mis otros familiares que también me apoyaron, en particular, a la familia Cajas Pérez.

…Gracias a Dios por su bendición, por no dármelo todo, sino solo lo necesario.

Publications

Results obtained during the development of this thesis have been published in refereed journals and refereed conference proceedings.

Journal Papers

- Erazo, O. and Pino, J. A. (2014). Estimating the Difficulty of Touchless Hand Gestures. IEEE Latin America Transactions, 12(1), 17-22.
  JCR Impact factor: 0.44 (Q4, Computer Science, Information Systems)

- Erazo, O. and Pino, J. A. (2016). Predicting User Performance Time for Hand Gesture Interfaces. Under review by International Journal of Industrial Ergonomics.
  JCR Impact factor: 1.0 (Q3, Engineering, Industrial)

Conference Papers

- Erazo, O. and Pino, J. A. (2013). “Analyzing Touchless Hand Gestures Performance”. In Proceedings of the 2013 Chilean Conference on Human-Computer Interaction (ChileCHI’13), pp. 14-17. ACM Press.

- Erazo, O., Pino, J. A., Pino, R., and Fernández, C. (2014). “Magic Mirror for Neurorehabilitation of People with Upper Limb Dysfunction Using Kinect”. In Proceedings of the 47th Hawaii International Conference on System Sciences (HICSS’14), pp. 2607-2615. IEEE Press. (Core A)

- Erazo, O. and Pino, J. A. (2015). “Predicting Task Execution Time on Natural User Interfaces based on Touchless Hand Gestures”. In Proceedings of the 20th International Conference on Intelligent User Interfaces (IUI’15), pp. 97-109. ACM Press. (Core A)

- Erazo, O., Pino, J. A., and Antunes, P. (2015). “Estimating Production Time of Touchless Hand Drawing Gestures”. J. Abascal et al. (Eds.): INTERACT 2015, Part III, LNCS, vol. 9298, pp. 552–569. Springer, Heidelberg. (Core A)

- Erazo, O., Baloian, N., Pino, J. A., and Zurita, G. (2016). “Easing Students’ Participation in Class with Hand Gesture Interfaces”. Accepted for the 10th International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI’16).

Table of Contents

List of Tables

List of Figures

List of Acronyms

CLC	Curves, Line segments, and Corners
DH	Dominant hand
FPS	Frames per second
GCP	(A model on) Gestures Conception and Production
G-phrase	Gesture phrase
G-unit	Gesture unit
HCI	Human Computer Interaction
KLM	Keystroke-Level Model
LM	Leap Motion
MT	Movement time
NDH	Non-dominant hand
NUI	Natural User Interface
OP	Operator
RGB	Red, green and blue
RMSE	Root mean square error
RT	Reaction time
SD	Standard deviation
SDK	Software development kit
ST	Stroke time
TG	Touchless Gestures
THDG	Touchless Hand Drawing Gestures
THG	Touchless Hand Gestures
THGLM	Touchless Hand Gesture Level Model
UI	User Interface
WOz	Wizard of Oz

Chapter 1

Introduction

1.1 Motivation

People communicate naturally among them through voice and body language so that it would be desirable to be able to do it similarly with computers. Starting with connection/disconnection of cables, then using command lines and graphical interfaces, the communication between user and computer has experienced notable improvements over the years. These improvements have contributed to the fact users get results from computers perhaps in an easier way than they may expect or need. Much credit should be given to the “direct manipulation” paradigm in most cases, using haptic devices such as keyboards and mice. However, users must learn to use these devices or other related hardware, while “new interface technologies should be more accessible without requiring long periods of learning and adaptation” (Wachs, et al., 2011). This observation leads to explore the development of new interaction styles that people may be able to use in a way as “natural” and intuitive as possible (Wachs, et al., 2011).

A trend to employ one or more body parts to interact with software applications has been shown in recent years with the aim of achieving more natural interactions. These interaction styles are based on the recognition of user’s actions through, for example, speech, touches or movements of either hands or other body parts in the space, eliminating the need for manipulating intermediate input devices. The term used to refer to these types of interfaces is “Natural User Interfaces” (NUIs). The purpose of NUIs is users “act and feel like a natural” (Wigdor & Wixon, 2011) while interacting with the corresponding content. Although this purpose may be achieved in several ways, we focus mainly on the use of hand gestures.

Hand gestures have long been considered a promising interaction style in the search for natural and intuitive human-computer interactions. Bolt was one of the pioneers in proposing the use of gestures as an alternative input style more than thirty years ago (Bolt, 1980). The expected advantages of this interaction style were evidenced and confirmed in later works. For example, in 1993 when Baudel and Beaudouin-Lafon proposed the Charade system to assess the effectiveness of gesture input for real applications noted that natural, terse, powerful, and direct interactions are some advantages of using gestures (Baudel & Beaudouin-Lafon, 1993). Despite these and many other works have demonstrated the benefit of gestures as an interaction technique, the application of gestures in practice has been limited (Karam & Schraefel, 2005). One of the reasons may be the constraints placed by hardware devices, such as the use of data gloves which also make interactions cumbersome (Wachs, et al., 2011).

An interesting situation is that a recent sustained trend of hardware innovations has now made feasible the communication using touchless hand gestures (THG) instead of touching keys or displays. THG are performed in mid-air without holding or wearing input devices or markers for tracking. Inexpensive devices like Microsoft Kinect, Intel RealSense, or Leap Motion can capture those gestures people make with their hands without haptic contact. Furthermore, availability of large high-resolution displays and THG set the stage for a computer-human interaction in which perhaps the computer will present most of the information/options/results and the human will provide few choices/requests/specific data.

These recent advances have broadened the domain of applications based on THG. The advantages of THG have been exploited in various applications including entertainment (Guo, et al., 2011; Vatavu, 2012), health (Gallo, et al., 2011; Erazo, et al., 2014), education (Jagodziński & Wolski, 2014; Blum, et al., 2012; Erazo, et al., 2016), augmented reality (Piumsomboon, et al., 2013), and robotic (Obaid, et al., 2012). In fact, THG-based interfaces are more suitable than others in those scenarios where due to reasons such as hygiene, safety, and/or distance is not advisable, desirable or possible to touch a device or surface. Take the hospital environment as an example: the shared use of computers by practitioners, patients and visitors is not recommended due to the dangers of contagion. Actually, public tactile devices are potential sources of sickness contagion in general. Also, public spaces like airports, bus stops or shopping malls can be places where passers-by may ask directions, locations, providers, etc. with THG. More specific uses are also foreseeable, e.g., consider the case of physicians in an operating room interacting with medical image data (Gallo, et al., 2011).

The devices employed by users to manipulate the content in all these scenarios are simply their own hands. Moving hands in the air is something that most people can do with minimal or no training. Then, interactions using THG may appear natural to users who may also reuse existing skills (Blake, 2012; Webb & Ashley, 2012).

Consequently, the benefits of THG and the progress reached in this field lead to hypothesize that traditional point-and-click user interfaces may not be the only ones available in the near future for a wide range of applications. It is reasonable to expect NUIs based on THG should gain much wider dissemination than the current situation. Therefore, UI designers should have available the necessary tools to design and evaluate their applications properly according to the adoption and proliferation of this interaction style.

1.2 Justification

Despite the applications and benefits offered by NUIs based on THG, there still remain challenges (Wachs, et al., 2011; Norman, 2010) that need to be addressed before UI designers begin to develop software systems to be deployed, e.g., at the nearby shopping mall. Although there are significant advances in hardware devices, enhancements are desirable especially in order to improve aspects such as hands and fingers tracking, number of tracked users, environmental conditions, distance, accuracy, etc. Better devices and algorithms should induce larger gesture sets and better user experiences. Also, new and/or better usability evaluation methods would be appreciated by UI designers, considering there is no standard set of hand gestures and the diversity of possible applications.

Even assuming a set of gestures is available to design UIs, we still have the problem of choosing specific gestures to associate them to meanings. Several criteria may be used for this purpose. One of them may be the appropriateness of the candidate gestures to the intended meaning by considering features like user preferences (Wobbrock, et al., 2009; Vatavu, 2012; Nielsen, et al., 2004; Piumsomboon, et al., 2013; Obaid, et al., 2012), social acceptability (Rico & Brewster, 2010), or memorability (Nacenta, et al., 2013). Another selection criterion may be the difficulty of teaching gestures to the user (Kamal, et al., 2014; Anderson & Bischof, 2013; Ismair, et al., 2015) or the simplicity to perform them. Guidelines and heuristics for designers along these design criteria provide useful recommendations, but they do not allow quantitatively selecting gestures or making comparisons.

There is yet another design criterion to choose hand gestures from a set of candidate ones, and it is the subject of this dissertation. It concerns the time efficiency to perform the various gestures. This criterion might not be extremely relevant in the case of occasional users asking for something simple to a computer system in a shopping mall (though they may give up if they do not succeed immediately (Walter, et al., 2013)), but it is particularly important in the case of repeated executions of the gesture in a time sensitive scenario, e.g., the physicians browsing images forward and backwards while operating a patient.

Whereas absolute gesture execution times may be useful, UI designers may be interested in comparing two or more design options instead. A classical approach is to involve users to choose gestures or analyze interface designs. However, this approach requires dealing with the logistic difficulties of doing tests with real users, regarding planning, timing, laboratory setup, recruiting, and conducting experiments. Therefore, designers may do their selection based on predictive evaluations –if available – instead of recruiting users to collect and analyze data, especially at early design stages.

Predictive evaluation is a strategy to evaluate interfaces that provides indications on how users could execute interaction tasks. Simulation of the biomechanics of human motion (Nunes, et al., 2015) is a method that is gaining attention for obtaining descriptions of users' movements with relatively low costs (Bachynskyi, et al., 2014; Bachynskyi, et al., 2015). Another approach is the use of predictive models (MacKenzie, 2013; MacKenzie, 2003) to quantify human performance, for instance, in terms of time (the period a user takes to accomplish a set of tasks (Card, et al., 1980)). Many problems in Human-Computer Interaction (HCI) are explored using model-based evaluation especially due to its advantages such as analyzing interface designs and making changes without implementing a real system (MacKenzie, 2013; MacKenzie, 2003; Kieras, 2003). Thus, model-based evaluation is a valuable supplement to conventional usability evaluation that is especially useful in situations where it is difficult to conduct tests with human subjects or as a preliminary step to evaluate an interface (Kieras, 2003). Of course, a consequence of adopting model-based evaluation to produce predicted measurements of usability is that quantitative user models are necessary. Though predictions of some interactions like typing or pointing, clicking and selecting can be computed using models formulated previously (e.g., (Card, et al., 1980; Fitts, 1954; Accot & Zhai, 1997)), more complex interactions as in the case of gestures performed in the air require either extending/adapting the existing models with new parameters or developing new models.

Accordingly, this thesis proposes to study model-based evaluation for NUIs based on THG. The corresponding model(s) should allow forecasting required time to complete tasks without users' participation to analyze interface designs. Thus, we hope the resulting model(s) become a useful tool for software designers to carry out usability assessments, improve UI designs, and develop better software applications using NUIs and THG.

1.3 Problem Statement

Although there exist methods to evaluate NUIs based on THG, they have various limitations. On the one hand, the proposed methods to select gestures require user participation (e.g., (Wobbrock, et al., 2009; Vatavu, 2012; Nielsen, et al., 2004; Piumsomboon, et al., 2013; Obaid, et al., 2012; Barclay, et al., 2011)), and it may not be advisable to use those methods at early design stages due to the cost of collecting and analyzing data (MacKenzie, 2003; Kieras, 2003). On the other hand, designers can use predictive models to assess interfaces quantifying human performance (MacKenzie, 2013) particularly the time. Despite model-based evaluation (MacKenzie, 2013; MacKenzie, 2003; Kieras, 2003) has been widely used to analyze interaction problems in HCI (MacKenzie, 2013), existing quantitative models are insufficient to evaluate NUIs based on THG due to any of the following reasons:

- They were formulated for other interaction styles (for example, (Card, et al., 1980; Accot & Zhai, 1997; Cao & Zhai, 2007; Isokoski, 2001; Ferreira, et al., 2009)).

- Their extended versions are not applicable to THG (for instance, (Holleis, et al., 2007; Luo & John, 2005; Song, et al., 2013; Pettitt, et al., 2007)), or the feasibility of applying them has not been verified yet (as in the case of (Card, et al., 1980; Accot & Zhai, 1997; Cao & 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 & Lalanne, 2013; Pino, et al., 2013; Polacek, et al., 2012; Zeng, et al., 2012) and to compare devices such as Kinect and Wii (Sambrooks & Wilkinson, 2013; Pino, et al., 2013; Polacek, et al., 2012)).

Therefore, new models to evaluate NUIs based on THG are necessary because a comprehensive model to estimate users' performance time does not exist. This fact leads us to ask the following research questions:

Can we apply a predictive model to forecast the required time to execute tasks on NUIs based on THG?

Can we use or adapt existing models initially developed for other types of user interfaces to assess NUIs based on THG?

1.4 Objectives

The general objective of the present thesis is to develop a predictive model that allows estimating required time to execute a simple task using a NUI based on THG in an acceptable way, easing the analysis of the user interface concerning options that may help increase user performance. Computed values using this model and actual user times are used to determine whether the predictions are acceptable or not, i.e., to calculate the values of selected metrics. Therefore, error forecasts should be within usual ranges in the field, and there should be high positive correlation between estimated and observed times.

We propose four specific objectives to achieve the general objective.

O1. Analyze the possibility of extending/adapting and applying quantitative models created for other types of interaction to predict performance time of touchless hand drawing gestures (THDG).

O2. Formulate a model to predict performance time in doing tasks on NUIs based on THG from basic theory.

O3. Empirically validate the proposed model comparing predicted times with observed times.

O4. Generate a procedure to apply the developed model.

1.5 Models

We need to begin with an account of models in order to accomplish the objectives stated in the previous section. Models play an important role in many scientific and professional fields. However, there is no uniform terminology concerning the term “model” (Koperski, 2016). From a broad perspective, a model is a representation of a selected part of the world (i.e., “the target system”) that one wants to understand (Koperski, 2016; Frigg & Hartmann, 2012). Significant components of that system are represented in the model doing abstractions of it by using physical materials, equations, pictures, analogies, etc. In a few words, a model is a simplification of the real world presented with other means.

Beyond the needed simplifications, models have proven to be powerful instruments. When somebody needs to study a system, s/he might observe or experiment with the actual system, or build a model that represents the system to study it as a surrogate for the actual system (Law, et al., 1991) (Figure 1.1). This approach is especially useful when it is difficult, impractical or impossible to obtain measures of the target system directly (though model outcomes may be less reliable). Overall, models usually serve as tools for understanding, simulating, predicting, or communicating results (Mulligan & Wainwright, 2004). A significant number of disciplines have exploited the valuable characteristics of models. This fact has conducted to a proliferation of types of models (e.g., (Frigg & Hartmann, 2012; Weisberg, 2007; Morton & Suárez, 2001)), which has derived in a lack of universal categorization (Mulligan & Wainwright, 2004). Some notions to categorize models are phenomenological models, scale models, mathematical models, and computational models (Frigg & Hartmann, 2012). Although an alternative could be thinking of models as varying along a continuum (MacKenzie, 2003), two kinds of models are mainly used in our area: descriptive and predictive (MacKenzie, 2013; MacKenzie, 2003; Holleis, 2009). The first ones help in understanding users interacting with computers by visual aids or verbal descriptions, whereas predictive models use mathematical expressions to forecast users' performance. Broadly speaking, descriptive models are concerned with qualitative (sometimes graphical) rather than quantitative analysis/results.

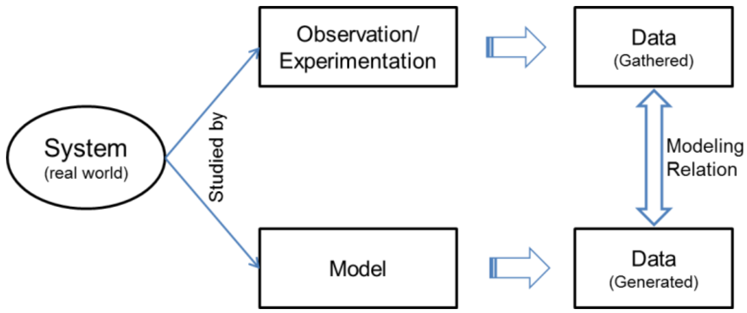


Figure 1.1. System and model (based on (Law, et al., 1991; Zeigler, et al., 2000)).

Either models of interaction or other kind of models, they must be abstracted from the real world; i.e., it is necessary to follow a process of *model building*. Several common steps are applied in most cases, though they may change depending on the used strategy, model type or discipline. The building of a model can start with the specification of the target system (e.g., an interface type in HCI) and the intended aspects to be represented by the model (Weisberg, 2007). The desirable characteristics of model building—such as precision, simplicity and generality—should also be considered (Weisberg, 2003). Likewise, a set of *assumptions* must be stated carefully because the value of the model depends on them directly (Mulligan & Wainwright, 2004). Moreover, a theoretical definition of the model may be made since the beginning; that is, we can begin by *formulating* the model, and then, going on to perform an analysis of it (Weisberg, 2007). The next step, *model calibration*, can take one of two forms: fixing parameters (e.g., apply the model to a concrete system) and refining descriptions (e.g., correct factors) (Morton & Suárez, 2001). After formulating and calibrating the model, it must be validated. *Validity* is the modeling relationship that links the target system with the model (see Figure 1.1). It concerns with how well the data generated with the model agrees with the data gathered from observing the system (Zeigler, et al., 2000). In other words, the validation of the model indicates how close the output of the model is in comparison to the output of the real system using, for example, an error tolerance (Weisberg, 2007). Later, the model could be revised, improved and/or extended.

The application of these steps, conforming to each of our objectives, has allowed us to build several models for interactions based on THG as main results of this thesis (Figure 1.2). First, using related works we derive a descriptive model aimed at helping in understanding how user conceive and produce gestures in mid-air. Second, the literature provides various predictive models that we extend (i.e., calibrate) to encompass THDG. Although we provide several models, only the one suggested as the best one is used in our later studies. Afterwards, we introduce a model which is the most important model built as part of this dissertation. We refer to THGLM, which is a comprehensive model enabling designers predict the time to accomplish tasks on NUIs based on THG. All these models are relatively easy to use and have acceptable performance according to our results, but they also have some limitations due to the needed simplifications. However, if these models are utilized in line with the corresponding assumptions, they should be a useful complement for practitioners and/or researchers in the analysis or evaluation of their UI designs.

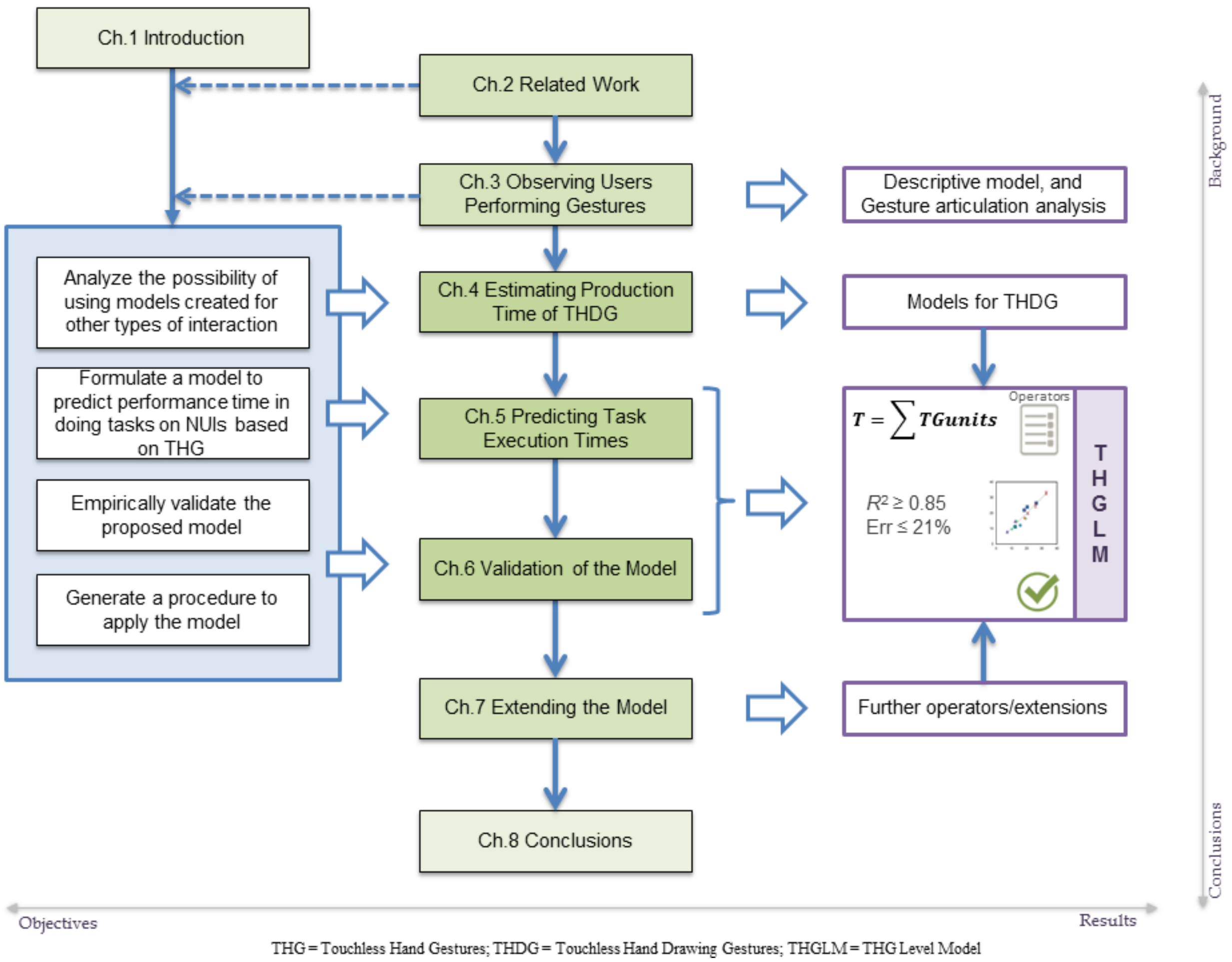


Figure 1.2. Structure of the thesis.

1.6 Methodology

The methodology consisted of various steps that were carried out to achieve the proposed objectives. We started by performing a bibliographic review to know the state of the art in NUIs, THG and models in HCI. This review provided the background to prepare an initial exploratory phase with the aim of clarifying initial ideas and "prepare the ground" for the following studies. Some trial experiments allowed making initial decisions (such as data collection and selection of participants) and verifying some technical aspects that were necessary to develop experimental software and to perform the experiments (e.g. participant's location, devices for gesture acquisition, experiment setup, etc.). Moreover, we conducted a study to better understand how users articulate gestures in the air under unconstrained conditions during this preliminary phase. We moved to gestures modeling armed with the experience and findings gained from this exploration.

In general, the methodology used to undertake the core of this research was informed by the model building method (summarized in the previous section). As stated above, the goal of this dissertation is to model interactions based on THG in a quantitative way. To achieve it, we have assumed that interactions are carried out by young healthy adult users (maybe teenagers as well) using only one hand. Interactions occur at a distance from a display. The hands hold no device. In summary, the interaction is performed without haptic contact. Moreover, only novice users took part in the experiments because there are still few expert gesture users nowadays. These assumptions were applied for the remaining of the research.

We analyzed the possibility of reusing existing models that were formulated for other styles of interaction before building the proposed model. For this goal, we performed a bibliographic review to find the candidate models. Next, the descriptions of these models were examined and refined. The corresponding parameters were also estimated because the existing ones were only applicable to the interaction styles for which those models were proposed (e.g., pen-based interactions). All model versions were evaluated before comparing the best versions of each model. Overall, these processes were carried out by asking users to perform various gestures in the air following explicit instructions.

The previous findings and related works from HCI and other fields provided the basis to develop the proposed model. We were especially inspired by Card et al.'s work (Card, et al., 1980; Card, et al., 1983) to build and evaluate our model because KLM is a well-known and validated model, and we tried to provide designers with a relatively easy-to-use model that could be applied by novice designers as well. Accordingly, we started with a theoretical definition of our model based on KLM and several works from psychology that refer to the analysis of gestures (e.g., (Kendon, 2004; McNeill, 1992)). This formulation was complemented with a systematic bibliographic review (Kitchenham, 2004; Webster & Watson, 2002; Lisboa, et al., 2010). Its goal was to identify and select the operators to be included in the model. Next, the calibration of the model was made by estimating the model parameters (i.e., conducting a study to measure/estimate the operator times). Subsequently, the model was validated empirically with the participation of users, as well as UI designers. The model precision was determined by using the corresponding metrics and considering typical ranges in the field for which we used KLM as a baseline. Finally, we performed an additional user study with the goal of extending the developed model.

Besides, we developed *experimental software* (MacKenzie, 2013) for conducting the experiments and collecting data (though we also used a couple of third party applications to validate the proposed model). The functionalities of these programs depend on each study, but some of them are: (1) guide participants providing the necessary instructions; (2) compute the interaction space (or gesture space); (3) recognize performed gestures; (4) log/record and compute the necessary data on the interaction (e.g. participants' body/hand tracking, tasks, times, etc.); (5) segment gestures in phases. Further software was developed to validate the model (e.g., solve a puzzle or view images).

Referring to the hardware setups, they consisted of a screen to allow the interaction and gesture acquisition devices. TV monitors and projected displays were used as screens (as needed). Participants stood at a distance from the screen to interact with the software for performing the tasks. Participants' gestures were collected employing devices such as MS-Kinect and Leap Motion.

1.7 Thesis Structure

The structure of this thesis is depicted in Figure 1.2 and described below.

Chapter 2 introduces some basic concepts related to model-based evaluation, NUIs and gestures in HCI.

Chapter 3 presents the results of an exploratory study for better understanding touchless gestures. The study described in this chapter focuses on the execution of gestures in general and provides a basis to analyze and understand user gestures before formulating the model.

Chapter 4 describes the assessment and adaptation of several quantitative models, which have been previously proposed to estimate the production time of mouse and pen interactions, to be applied to touchless hand drawing gestures (air figures of letters and numbers).

Chapter 5 describes a model to predict performance time to complete tasks without users' participation. The user study to compute the needed parameters and the use of the model are also described.

Chapter 6 presents the results of the validation of the model by describing the empirical studies that were performed and by discussing the model performance. This chapter also examines the validity of the model to make predictions and analyze interface designs with the participation of UI designers in the corresponding study.

Chapter 7 shows how to extend the model and it presents some further possible extensions.

Chapter 8 provides the conclusions of this dissertation and it discusses some limitations and possible future research directions.

Chapter 2

Related Work

This chapter presents some concepts and related works relevant for this dissertation. It starts by defining Natural User Interfaces (NUIs) and presenting some application scenarios for those NUIs based on touchless hand gestures (THG). Then, the chapter provides further details about hand gestures by reviewing some basic aspects to work with them. The chapter also describes the use of model-based evaluation in HCI as well as some of the most commonly used models. Since these models were developed for other interaction styles, their descriptions also include a brief discussion about the possibility of extending or adapting them to THG.

2.1 Natural User Interfaces

User Interfaces (UIs) have evolved and today there is a trend to build them as "natural" as possible. Many of the advances reached over the years have been focused on attaining this naturalness. This progress has derived in the current availability of new body-tracking devices such as Microsoft Kinect and Leap Motion, which have contributed significantly to the emergence of a new class of graphical UI labeled as Natural User Interfaces.

A NUI is a user interface designed to reuse existing skills (Blake, 2012; Webb & Ashley, 2012) to interact appropriately with content that should become invisible with successive interactions (Webb & Ashley, 2012). NUIs should allow interacting with applications directly without using intermediate input devices such as mice or keyboards. Users can control applications through body movements or gestures of everyday life that can be performed with fingers, hands or whole body. In reality, the use of gestures as a new interaction modality is perhaps the most distinctive characteristic on NUIs. However, NUIs can also include more than gestures, e.g., speech recognition, facial expressions or multitouch (all beyond the scope of this dissertation). In any case, "a NUI is not a natural user interface, but rather an interface that makes your user act and feel like a natural" (i.e., a *natural user* interface) (Wigdor & Wixon, 2011).

Based on these previous ideas, we define a NUI based on THG for this thesis as follows: a UI that may appear “natural” to users because little to no training is needed to use it. Users interact with it through physical movements (i.e., gestures) that convey meaningful information and are performed in the air with the hand and without haptic contact.

2.1.1 Examples of Application Domains

NUIs are becoming increasingly popular in different fields that go beyond only entertainment. Entertainment industry is one the most important application fields of this kind of interfaces. A primary challenge is tracking and recognizing users and ensuring smooth and natural interactions (Guo, et al., 2011). Devices like Kinect provide technologies that combined with a careful UI design may provide spectacular gaming experiences. However, entertainment is not the only area where NUIs can be advantageous. In fact, the application scenarios for this type of interfaces are still expanding.

In general, those scenarios where users interact with software applications without physical contact or at a distance from a display without touching it provide good opportunities for NUIs based on THG. Hospitals, classrooms, museums and bus stops are some examples of these scenarios.

Today there are many proposals available for classrooms or education. Among the various options, NUIs based on THG have been used to teach/learn subjects (Blum, et al., 2012), control presentations (Erazo, et al., 2016), simulate laboratories (Jagodziński & Wolski, 2014), etc. These proposals usually require users stand in front of a display for interacting, and hence, a dedicated space is needed. Beyond this constraint, the use of NUIs in classrooms may contribute to make the teaching-learning process becomes fun and to encourage students to participate in classes (Erazo, et al., 2016; Hsu, 2011).

Healthcare is other field in which NUIs can take advantages. An example is sterile environments where users need to interact with software applications and avoid contamination at the same time (e.g., exploring medical images in operating rooms (Gallo, et al., 2011)). Another example is helping people with movement limitations in their rehabilitation process, especially at home, where patients’ motivation may decrease, the control of therapies is low, and patients may also need to control the application by themselves (Figure 2.1) (Erazo, et al., 2014). These examples show that NUIs based on THG can be used either at hospitals or at homes with medical purposes.

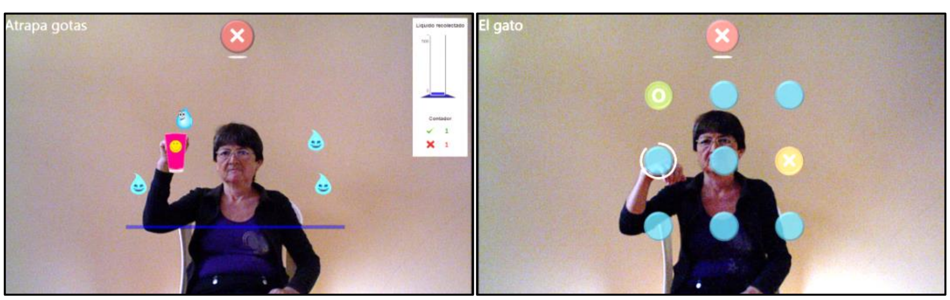


Figure 2.1. Application example of NUIs to help patients with upper limb dysfunction in neurorehabilitation process: Catch virtual water drops (left), tic-tac-toe (right) (Erazo, et al., 2014).

Another application area can be found in public places (Ren, et al., 2013), such as hospitals, bus terminals, airports, etc. Public displays enable the transition from static broadcast displays to interactive displays, and thus, allowing users to select different contents or applications, and increasing their utility (Walter, et al., 2014). NUIs play an important role in these scenarios because most users may approach this kind of applications for first time and employ little time to learn how to interact with applications before deciding whether to continue or not (Walter, et al., 2013). Additionally, both hygiene and security are reasons to prefer the use of THG in public displays.

Of course, NUIs are not only limited to the mentioned areas. Other possible scenarios/fields are kitchens (e.g., user' hands could be dirty with food substances while cooking (Panger, 2012)), stores (e.g., to try on clothes without undressing (Giovanni, et al., 2012)), interaction in 3D environments (e.g., with 3D databases (Herrera-Acuña, et al., 2015)), and geographic information systems (Boulos, et al., 2011). In general, conditions such as sterile environments, vandalism-prone environments, or shared spaces would be appropriate for interfaces that employ touchless gestures instead of interfaces requiring touching a surface or device.

2.2 Gestures

People commonly perform gestures as part of everyday activities to express some information. Of course, there are too many scenarios in which gestures are used. The possibilities range from very simple situations, for example talking with a friend to discuss ideas, to complex scenarios such as operating rooms or battlefields. Consequently, gestures are studied in different areas such as Sociology, Biology, Linguistics and Computer Science.

Although gestures generally refer to movements performed with the hand 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 was proposed by Kurtenbach and Hulteen (Kurtenbach & 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 and extending the arms horizontally, or pushing with a hand (Figure 2.2). Nevertheless this thesis mainly studies "hand gestures" without considering those ones performed with one or more fingers (e.g., fingerspelling).

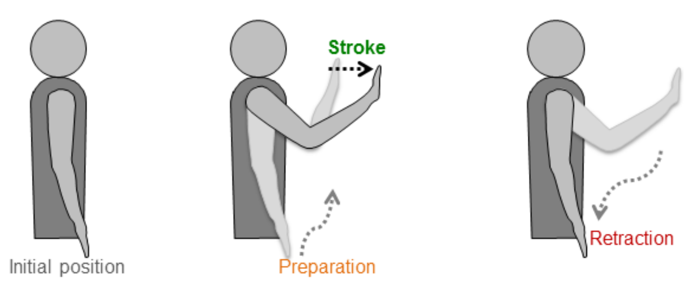


Figure 2.2. Example of gestures used in this thesis decomposed in phases.

2.2.1. Understanding Gestures

Despite multidisciplinary studies from gesturing in human-human communication have provided most of the basics behind gestures as an interaction technique, this section explains some concepts for analysis of gestures from a Computer Science perspective. The use of gestures as a mechanism of interaction with computers has been studied for more than thirty years. The gained insights from those studies are varied. Several attempts to grasp and unify findings have been made (e.g., (Karam & Schraefel, 2005; Wachs, et al., 2011; Argelaguet & Andujar, 2013)), but they may not cover all possible aspects for understanding gestures. Alternatively, we use five categories to organize the literature for understanding gestures: structure, taxonomies, analysis, formalisms, and tools. Although we are particularly interested in the structure of gestures, all categories are briefly reviewed to give a global view.

Gesture Structure: Gesture Units

A good way to analyze hand gestures is by looking at their temporal structure as they are a dynamic process. Gestures are defined in terms of *gesture units* (*g-units*), *gesture phrases* (*g-phrases*) and *phases* according to this approach (Figure 2.3; see also Figure 2.2) (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. It may contain one or more g-phrases.

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. Stroke is the only obligatory phase in which the gesture meaning is expressed; i.e., the peak of effort and shape of the gesture are clearly expressed in this 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). The final phase, *retraction* or *recovery*, is not considered to be part of any g-phrase. It may happen when the hand relaxes and is returned to some resting position or to the original one. In addition, some gestures can have a single meaningful still phase (called *independent hold* (Kita, et al., 1998)) instead of a stroke for static gestures (or *hold gestures* (Neff, et al., 2008)). As a result, we distinguish two types of phrases *hold-phrase* (*H-phrase*) and *stroke-phrase* (*S-phrase*) (Erazo & Pino, 2015; Neff, et al., 2008).

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

Figure 2.3. G-units, g-phrases and phases (based on (McNeill, 1992; Kendon, 2004; Kita, et al., 1998)).

The main application of g-units is the analysis of gestures performed as part of humans' conversations or speeches, but they have also been used in our area. Some examples of the use of g-units are the production of gestures of animated characters (Neff, et al., 2008), develop languages or notations for gestures (Giuliani & Knoll, 2008), annotation and analysis of gestures using video annotation tools (Neff, et al., 2008), and temporal segmentation of gestures from other unintentional movements (Pavlovic, et al., 1997). These works demonstrate that the advantage of using g-units is that they allow analyzing continuous production of gestures (Kita, et al., 1998). Consequently, we consider g-units are a good starting point towards the analysis of gestures and the formulation of our model.

Taxonomies

A significant number of taxonomies can be found in the literature to help understanding gestures in the context of HCI. In a broad perspective, gestures are predominantly clustered in two types (Quek, et al., 2002): semaphores (symbols used to convey specific meaning to machines) and manipulatives (used to control objects by a tight relationship between user's and object movements). However, there exist other specific ways of labeling gestures, such as semantic (Nielsen, et al., 2008), form (Wobbrock, et al., 2009; Obaid, et al., 2012; Piumsomboon, et al., 2013), functional (Nielsen, et al., 2008), etc. These labels have also been used in those taxonomies extended or adapted from previous ones. For instance, Aigner et al.'s taxonomy (Aigner, et al., 2012) was formulated based on Karam and Schraefel's taxonomy (Karam & Schraefel, 2005), which in turn is based on Quek et al.'s taxonomy (Quek, et al., 2002). Moreover, several taxonomies have been proposed for specific scenarios as in the case of controlling humanoid robots (Obaid, et al., 2012), augmented reality (Piumsomboon, et al., 2013), and storytelling (Kistler & André, 2013). (In fact, these three taxonomies were adapted from one proposed for surface gestures (Wobbrock, et al., 2009).) Taxonomies allow classifying and organizing user gestures, but one should note that "the best one for use in a given application depends on the concrete context" (Nielsen, et al., 2008)(p. 85). Actually, many systems do not focus on a single type, and hence, multiple gesture types are used instead (Karam & Schraefel, 2005).

Gesture taxonomies are useful for describing different gestures made by users. For example, a couple of works on user-defined gestures showed, using the corresponding taxonomies, that more than half the gestures proposed by participants for the requested tasks fell in the categories dynamic and one-handed (Obaid, et al., 2012; Piumsomboon, et al., 2013). Gesture taxonomies are useful as illustrated in this example, but they are not enough for analyzing and understanding gestures.

Analysis of Gestures

Today, there is a large body of research that has contributed with insights to increase knowledge and understanding of gestures, and particularly THG. We consider two groups in an attempt to organize that research. The first group refers to those studies intended for analyzing specific characteristics or features of gestures such as usage of gesture types (Aigner, et al., 2012), social acceptability (Rico & Brewster, 2010), memorability (Nacenta, et al., 2013), teaching and learning (Kamal, et al., 2014; Anderson & Bischof, 2013; Ismair, et al., 2015), trajectory (Erazo & Pino, 2013), difficulty (Erazo & Pino, 2014). The second group clusters the works in which specific gestures, (e.g., dwell or holding, and pushing (Hespanhol, et al., 2012; Yoo, et al., 2015)), or gestures for particular tasks (e.g., pan and zoom (Nancel, et al., 2011), select items (Walter, et al., 2014), and browsing image collections (Koutsabasis & Domouzis, 2016)) have been studied. In general, works of both groups have contributed with guidelines for designing and/or evaluating UI based on gestures. Some examples of these guidelines/recommendations are the following ones:

- Users would prefer employing both different types of gestures and number of hands depending on the meaning of the gesture (Aigner, et al., 2012).

- Both location and audience would play a significant role in the acceptability of gestures (Rico & Brewster, 2010). For instance, users would prefer interacting with gestures at home with family and/or alone (Rico & Brewster, 2010).

- Users would change preferences and acceptance even after a first positive experience (Rico & Brewster, 2010).

- User-defined gestures would be easier to remember and take less time to learn than pre-designed gestures (Nacenta, et al., 2013).

- A combination of texts with videos or images should be preferred for instructing people to collect gestures in order to create data sets (Fothergill, et al., 2012).

- Dwell (or holding, or feedback time) time should be in the range of 350-600 ms (Müller-Tomfelde, 2007).

- Dwell is more accurate and easier than other gestures to make selections (Hespanhol, et al., 2012; Walter, et al., 2014), but some people may prefer pushing (Yoo, et al., 2015).

Predominantly users involved in gesture studies are requested to perform gestures, interact with applications, and/or provide opinions. However, there are some studies in which participants also play a different and/or complementary role. For instance, participants can evaluate and classify gestures by watching videos (Rico & Brewster, 2010), or a participant can interpret gestures executed by another one instead of using an automatic recognition (Aigner, et al., 2012).

In addition, there are some situations in which the use of an apparent intelligent system is preferred to the one that recognizes gestures automatically. This approach is named as "Wizard of Oz" (WOz) (Dahlbäck, et al., 1993). The main idea of WOz studies is participants believe they are interacting "normally" with a system that provides the results/information but responses are in fact provided by a human operator (who is referred to as the "wizard"). WOz methodology is especially useful mainly when researchers need participants perform gestures or interact with the apparatus in the way they desire rather than being limited to the capabilities of the recognition system (as in the case of (Henschke, et al., 2015; Höysniemi, et al., 2004; Nielsen, et al., 2004)). The wizard is responsible for indicating successful recognition (e.g., by using the keyboard or the mouse (Höysniemi, et al., 2004)), which give a better control of the experiment.

Formalisms

Formal specifications or notations are useful when working with gestures to express them in a way that other designers or application can understand. Gesture notations are often used to animate virtual characters that move and gesticulate. The characters produce the gestures that are given using the corresponding notation (Neff, et al., 2008). Other formalisms allow annotating gestures that can be readable by both humans and machines, for example, in conversations (Gibbon, et al., 2003). A final use mentioned here is the definition of gestures that should be recognized by the intended application (Spano, et al., 2012). Approaches of this type should enable including gestures easily in user interfaces. On the other hand, the proposals for specifying gestures formally employ various data formats including XML (Giuliani & Knoll, 2008), Petri Nets (Spano, et al., 2012), context-free grammar in Backus–Naur Form notation (Gibbon, et al., 2003), and specific codes or schemas (Choi, et al., 2014; Lausberg & Sloetjes, 2009). Moreover, some of these works also provide the corresponding tools to annotate or analyze gestures. Summing up, formalisms have been proposed to help in various fields where gestures are analyzed, but their usefulness for evaluating gestures quantitatively may be limited because the current trend is describing gestures in a relative or qualitative form and not using absolute values (e.g., distance and time).

Tools

The advent of new tools for gesture research has contributed to gestures study in HCI. The classical approach is video recording given that gesture analysis often relies on human observations. The executions of gestures or the interactions with applications using gestures are video recorded and subsequently are analyzed for a certain purpose, such as to identify and segment gestures in discrete units and phases (e.g., g-units, g-phrases and phases) and/or to examine time lapses. This kind of tasks can be carried out using programs for video viewing and editing (e.g., VirtualDub[1]), but nowadays there are custom-designed video coding tools that also allow doing gesture transcriptions. A couple of examples of these custom applications are ANVIL (Kipp, 2001) and ELAN (Lausberg & Sloetjes, 2009), which allow doing annotations of multimodal audiovisual material and exporting data for further (statistical) analysis. Though software of this type helps significantly in the analysis of gestures, researchers still have to do annotations manually.

2.2.2. Enabling Technologies for Acquisition

Several technologies allow enabling gesture acquisition to be interpreted by software. The type of device to be selected is influenced by the type of needed gestures. Input devices can be broken down into active and passive depending on whether physical contact is required or not (LaViola, 2013; Karam & Schraefel, 2005). An active device involves holding or wearing the device in some way to transmit information about movements, i.e., there is some physical contact. Devices of this type usually employ accelerometers, gyroscopes, electromyography (EMG), etc. Data gloves, Nintendo Wii Remote (Wiimote) and Myo[2] armband are examples of active input devices.

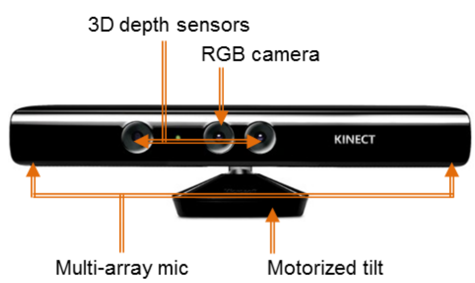
Passive devices, in contrast, provide an unobtrusive tracking of body parts, without requiring any physical contact with intermediate devices or wearing gloves or markers. They mostly use computer vision to capture movement and enable gestures, but they can also include other technologies like sound (Gupta, et al., 2012), light (Gupta, et al., 2011), or electromagnetic noise (Cohn, et al., 2012). Computer vision recognition has traditionally employed conventional cameras (e.g., webcams), though current depth cameras provide more information and better opportunities such as extraction of 3D data.

[1] VirtualDub is a free video tool for basic editing, mainly geared to AVI files (http://www.virtualdub.org/).
[2] Myo armband is a wearable device that lets you use the electrical activity in your muscles to wirelessly control your computer, phone, etc. (https://www.myo.com/).

MS-Kinect (http://www.microsoft.com/en-us/kinectforwindows/) is perhaps the most popular device easing the use of depth data. Kinect includes a RGB camera, depth sensors, a microphones array and a motorized tilt (Figure 2.4a). One of its main features is that it allows detecting and tracking specific points of the body that are obtained from depth maps and a posture detection algorithm (Shotton, et al., 2013). This characteristic allows avoiding calibration and pose detection before interacting with applications, aspect that was critical in applications based on RGB cameras. Moreover, and in addition to the great number of studies conducted using Kinect, evaluations of the performance of Kinect confirm its suitability to acquire gestural commands, especially since device range, noise, accuracy and errors are acceptable (Livingston, et al., 2012). Conversely, Kinect's performance may experience some degradation in some lighting conditions (e.g., in outdoor environments where sunlight can interfere). Furthermore, there are other characteristics that may be improved in future versions (for instance, field of view, and number of recognized and tracked people). Despite these limitations, Kinect is suitable for the type of gestures used in this dissertation. Even with its popularity and qualities, Kinect has nowadays several competitors such as Intel RealSense and Sony PlayStation Move. Like Kinect, these devices enable tracking the whole body and recognizing gestures.

Leap Motion (LM) is a device with similar advantages to the Kinect. Unlike Kinect, LM only allows tracking the hands, but it can also provide 3D data of fingertips. It consists of three infrared LEDs emitters and two infrared cameras (Figure 2.4b) to determine positions of tracked hands/objects in a limited space of less than one meter. LM works with a precision of less than 1 mm and frame rates of about 100 fps as confirmed experimentally (Guna, et al., 2014; Weichert, et al., 2013). These characteristics make LM suitable for applications based on hand gestures, and particularly, finger gestures. Moreover, and like Kinect, there are some technical aspects that should be improved in the future as in the case of field of view and tracked joints of fingers. Beyond the current limitations of Kinect and LM, some efforts have been made trying to exploit the positive characteristics of both devices together for gesture recognition purposes (Marin, et al., 2015).

a)



b)

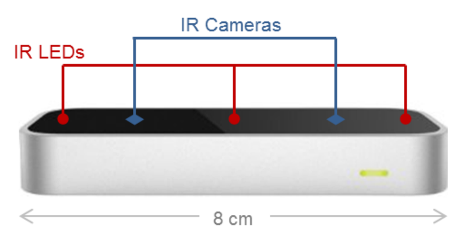


Figure 2.4. Main components of (a) Kinect and (b) Leap Motion.

2.2.3. Design Aspects

Various elements should be considered when designing NUIs based on THG. We have derived several of them from the analysis of related works. Other elements may be needed depending on the intended application, but we describe only those ones considered most relevant to this dissertation.

Gesture Space

It is the input space or input area where users will perform the gestures; it defines the scope of users' actions (Argelaguet & Andujar, 2013). Other names used to refer to gesture space are *interaction area/space* or just *input space*. The gesture space can be *in front of* or *around the user* according to (Walter, et al., 2014) (Figure 2.5). Gesture space *in front of users* has been the preferred option in previous works. In this case, the user moves the hand in front of him/her to perform the gestures. Several gesture spaces of this type have been proposed, such as the gestural input space used in conversations (McNeill, 1992), the whole space users can reach with the hands (Erazo & Pino, 2014; Erazo & Pino, 2013), the Physical Interaction Zone (PhIZ) (Microsoft Corporation, 2013), and the comfort zone (Kölsch, et al., 2003). Additionally, both types of gesture spaces have been used in commercial products as exemplified by Walter et al. (Walter, et al., 2014). A different approach is to use a personal space that is defined by the user to be mapped into the screen space (Jude, et al., 2014b), for example, by performing a gesture. In summary, *in front of* could be considered more general and preferred for our case taking into account that *around the user* may be adequate mainly for applications focused only on selections on a display.

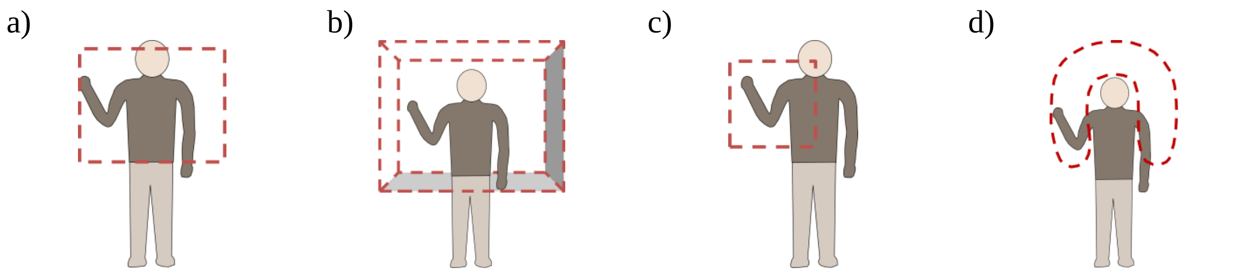


Figure 2.5. Gesture space: In front of the user: a) McNeill's (McNeill, 1992), b) whole reachable space with the hands (Erazo & Pino, 2014; Erazo & Pino, 2013), c) PhIZ (Microsoft Corporation, 2013); d) Around the user (Walter, et al., 2014).

Gesture Set

It is the set of gestures used to interact with an application; it is also called *gesture vocabulary* (Nielsen, et al., 2004). Each gesture of a set is usually associated to a meaning in the intended application. These gestures are defined by UI designers or selected by applying some procedure. Given that currently there is no standard set of THG, UI designers may use their own criteria to choose gestures or make their decisions based on gestures commonly used in touchless interfaces (e.g., (Walter, et al., 2014; Erazo & Pino, 2015)). Examples are gestures for selecting and/or browsing (as point and swipe (Walter, et al., 2014)), confirming elements (such as dwell, swipe, push, point, grip, wave (Walter, et al., 2014)), shortcuts (e.g., draw an "X" to close a window (Vatavu, 2012)), etc. However, designers can be interested in using more and/or different gestures for other actions.

UI designers can apply several methods for choosing gestures. For example, Nielsen et al. developed a procedure for finding a gesture vocabulary for an application by considering aspects like learning, ergonomics, and intuition (Nielsen, et al., 2004). Nevertheless, the most commonly used methodology nowadays may be the one proposed by Wobbrock et al. (Wobbrock, et al., 2009). They introduced a methodology for eliciting gesture-based commands from users for surface computing, and follow-up works verified the methodology for THG (Vatavu, 2012;

Piumsomboon, et al., 2013). The method is based on non-technical users' opinions about the cause and effect of gestures, and the final selection is made from the scores of each gesture for each command. Several researchers have contributed with sets of user-defined gestures following this methodology. Those sets are intended for various purposes, such as for controlling TVs (Vatavu, 2012), performing tasks in augmented reality applications (Piumsomboon, et al., 2013), navigational control of humanoid robots (Obaid, et al., 2012), and interactive storytelling (Kistler & André, 2013). In addition, Morris et al. proposed that user elicitation studies could be improved by generating various gestures, priming users, and involving partners (Morris, et al., 2014). Later, Hoff et al. experimentally tested the first and second suggestions in a follow-up work, but their results indicated that the practical effectiveness of these suggestions might be limited (Hoff, et al., 2016). Beyond, methodologies like these ones could be complemented or replaced by other possibilities. A first option is to consider other aspects not covered by these methodologies (e.g., difficulty (Erazo & Pino, 2014), or quality (Barclay, et al., 2011)). Another option is the use of predictive models to compute values for each gesture, and then, select those ones with the best scores (see Chapter 4). All in all, these proposals may help getting better gesture sets, but we consider the perfect solution does not exist yet.

User Representation

The user who is interacting with the application needs a way to know whether the application is recognizing his/her actions. This can be done in three ways (Walter, et al., 2014): *cursor*, *avatar* and *mirror image* (Figure 2.6). In the first case, a cursor (e.g., a hand cursor) represents the user's movements. The second one consists of a virtual avatar that mimics the user's movements (e.g., a skeleton drawn with lines). In the last case, the user's body form is represented in a way similar to s/he is looking him/her at a mirror (e.g., like a silhouette). An additional option is to use more than one representation (for instance, a cursor to show movements and a silhouette to identify the user). There is no best option, and hence, the selection should be done depending on the needs of each case.

a) b) c)

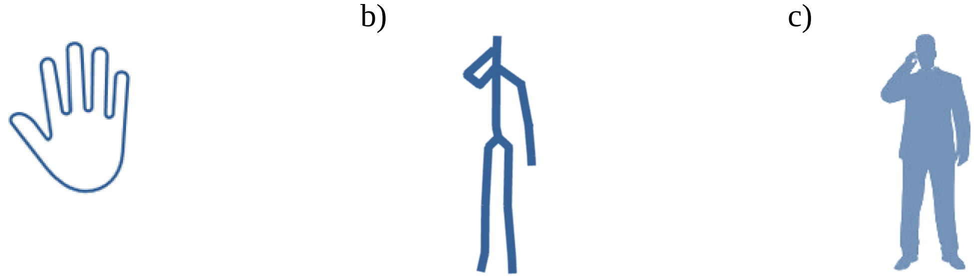


Figure 2.6. User representation: a) cursor, b) avatar, c) mirror (according to (Walter, et al., 2014)).

Interaction Style

It refers to the type of gestures to use such as those ones that can be performed using the whole hand (full-arm gestures), one or more fingers, or both. An example of the first case is Jagodziński and Wolski's virtual chemical laboratory that is based on full-arm gestures as grip and release (Jagodziński & Wolski, 2014). For the second case, the index finger can be used to move the cursor and the thumb to select (either opening or closing the thumb) as in the case of

(Schwaller & Lalanne, 2013). The selection of a style should be made depending on the target application and taking into account the input device capabilities; we use here only the first one.

2.3 Model-Based Evaluation

UI designers have available a variety of techniques to evaluate usability of UIs, each one with both advantages and limitations. Instead of analyzing all those methods, we concentrate only on models, which is the subject of this dissertation. Precisely, one way to support usability assessment is to use modeling techniques in HCI. Since user testing could be slow and expensive, model-based evaluation is a useful supplement to it (Kieras, 2003). Model-based evaluation is useful for designing, evaluating, or providing a basis to understand interfaces (MacKenzie, 2003), especially at early design stages, before starting to develop prototypes or the real UI. This approach also avoids dealing with the logistic difficulties of doing tests with real users during design stages, regarding planning, timing, laboratory setup, recruiting, conducting experiments, collecting and analyzing data. Despite the usefulness of model-based evaluation, user testing is still needed. The fact that models are simplifications of the real world derives in that there is no model that captures all possible details of the intended domain, and therefore, user testing is required to cover the remaining aspects (MacKenzie, 2003; Kieras, 2003). Then, the general goal of model-based evaluation is to get some preliminary usability results using models; i.e., "perform some of the design interactions in a lower-cost, higher-speed mode before the relatively slow and expensive user testing" (Kieras, 2003).

According to MacKenzie, models can be either descriptive or predictive depending on whether they use "verbal analogies and metaphors" to describe phenomena or mathematical equations to predict phenomena respectively (MacKenzie, 2013; MacKenzie, 2003)[3]. We discuss both kinds of models in this section but paying more attention to the predictive ones.

2.3.1 Descriptive Models

Descriptive models are tools that provide designers with frameworks or contexts to reflect on or describe a problem or situation (MacKenzie, 2003). They use depictions or verbal descriptions for studying and thinking about a UI. Though descriptive models generally do not provide quantitative measures of user performance, they are often used in HCI to approach problems. Some famous descriptive models for analyzing problems and designing interfaces are the following ones:

Key-action model (KAM)

KAM helps in the analysis of users interacting with computer keyboards to find design issues. For this goal, the keys of the keyboard are divided into three categories: *symbols keys*, *executive keys*, or *modifier keys* (MacKenzie, 2013; MacKenzie, 2003). The first category consists normally of letters, numbers, or punctuation symbols that deliver graphic symbols to applications. Executive keys carry out specific actions in the application or at the system level (e.g., ENTER or F1). Keys of the last group, instead of generating symbols or invoking actions, change the effect

[3] In fact, he suggests thinking "of models as lying in a continuum, with analogy and metaphor at one end and mathematical equations at the other" (MacKenzie, 2003).

of next keypress, which actually could type something (e.g., SHIFT or ALT). These are the ideas behind KAM that allow considering it as a simple but useful model for analyzing aspects of keyboard operations, for example, when contemplating new designs or the organization of left- and right-hand usage (MacKenzie, 2013).

Model of Bimanual Skill

Guiard's *model of bimanual control* (Guiard, 1987) focuses on human preferences in the use of the preferred and non-preferred hands in routine tasks. It is "a simple model, based on a non-quantitative physical approach, which aims at describing the logic of division of labor that appears to govern the variety of human bimanual asymmetrical actions" (Guiard, 1987). Guiard's model identifies the roles and actions of both hands. Namely, the non-preferred hand leads the preferred hand, sets the spatial frame of reference for the preferred hand, and performs coarse movements, whereas the preferred hand follows the non-preferred hand, works within established frame of reference set by the non-preferred hand, and performs fine movements (Guiard, 1987). Though the model was formulated in the field of experimental psychology, it was adapted later to analyze and inform problems about bimanual input in HCI (MacKenzie, 2013). Thus, the principles of this model have provided the basis for some approaches to create also bimanual gestural interfaces, in which, for example, the preferred hand can be assigned for pointing and panning, while the non-preferred for controlling zoom (Nancel, et al., 2011).

Three-State Model of Graphical Input

Buxton's *three state model of graphical input* (Buxton, 1990) helps elucidating properties of both pointing devices and techniques, and recognizing and exploring the relationship between them. As its name suggests, the model employs a simple state-transition diagram of three states for this goal. One state is *Out of Range* which occurs when the device moves out of the tracking range; it is used for clutching or repositioning the pointing device. Another state is *Tracking* for moving a tracking symbol around the display as in the case of a cursor. Finally, *Dragging* is the state for moving an item on the display or for grouping an amount of items together. The model allows the analysis of pointing devices in terms of these states. An example of its usefulness is that the model helped in the redesign of touchpads to be pressure sensitive (MacKenzie, 2013). Additionally, the model can be extended to capture other aspects of pointing device interaction. In particular, the Buxton's model has served as a basis to derive new models that describe interactive systems based on pointing and selecting with hand(s) gestures (Fikkert, et al., 2009; Zhou, et al., 2012).

2.3.2 Predictive Models

Unlike the models described in the previous section, we now move on to those models that concentrate on formalisms or equations to analyze user behavior instead of describing the semantics of devices or interaction techniques. They are usually labelled as predictive models because they allow getting estimated measurements of UIs usability (Kieras, 2003). These models generally use mathematical expressions to forecast user performance.

Performance can be measured using various dimensions, such as time, errors, learning, fatigue (Card, et al., 1980). However, this dissertation only addresses the efficiency aspect in terms of 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 predict users' performance time. One of the most important contributions on modelling is the Card, Moran and Newell's book (Card, et al., 1983), in which they describe several models that have been widely used to assess UIs and propose a theoretical basis to build new models afterwards. The rest of this section describes some of these and other popular models in HCI.

Model Human Processor (MHP)

The development of MHP (Card, et al., 1983) is one of the seminal works in the attempts to analyze how users interact with computers. 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 their interconnections (Card, et al., 1983). It is also assumed that these processors operate in series, and each one has a cycle time (derived from the literature) to calculate the time to perform tasks.

Despite MHP is in fact one of the most widely recognized models in our field, works related to gestures that use this model are unusual. The explanation of this fact is perhaps that MHP is not adequate to analyze gestural interactions. Although MHP provides a parameter for motor tasks, it may be insufficient for predicting performance with interfaces based on THG. While MHP oversimplifies human users' behavior, gestures may have different forms and difficulties.

Keystroke-Level Model (KLM)

KLM (Card, et al., 1980) is one of the most comprehensive models in the area (MacKenzie, 2013). Despite it was introduced in 1980, KLM is still one of the most useful models in the field. It is in fact a simplified version of another model, GOMS.

GOMS is a well-known predictive modeling technique in HCI also proposed by Card, Moran and Newell (Card, et al., 1983). Actually, GOMS models are the original approach to evaluate interfaces using modeling techniques (Kieras, 2003). GOMS allows analyzing user's behavior interacting with a system in terms of *Goals*, *Operators*, *Methods* and *Selection rules* (Card, et al., 1983). The analysis of a task begins with the goal that users want to achieve. Goals are reached using a method, which in turn contains operators. Operators are the actions that the software allows users to take. Rules are applied when there is more than one method to do the task or accomplish the goal. After writing out the method(s) to complete the task goal, the base time values assigned to the operators are added up to obtain the predicted task time. Additionally, each method can be represented at different levels of detail/difficulty depending on the selected variant since GOMS is a rather generic term used to refer to a family of models. We do not analyze all GOMS models, but we concentrate on the most practical one: KLM.

KLM (Card, et al., 1980) allows forecasting the time an expert user takes to execute[4] a task according to a certain method. It assumes the user faithfully executes the given method; i.e., performance must be error-free. This method must be specified in detail at the level of keystroke

[4] A unit task has two parts: *acquisition* and *execution* of the task [7]. Acquisition time is beyond the scope of this work.

operations using a set of operators (Table 2.1). Operators are actions—such as pressing keys, moving the mouse, etc.—that a user should follow to carry out the task. Each operator has execution times defined based on experimental data. Operators usually take constant times to be performed, but formulas can be used instead, as in the case of ***P*** and ***D*** operators. Then, the times required by all such actions are added up to estimate the total task time. The main advantage of KLM is that it allows estimating execution times in a fast and inexpensive way with little theoretical/conceptual background. Furthermore, KLM is a well validated tool as confirmed by the large body of research conducted with different types of UIs.

Table 2.1. Overview of KLM operators in relation to possible uses with THG (Card, et al., 1980). [a] Average skilled typist (more values are available for other typing skills). Applicable to THG: Y=Yes, N=Not, P=Perhaps.

Operator	Brief description	Value (s)	Applicable to THG	Comments
K	Keystroke, typing, or button press	0.20[a]	N	Gestures are different than keystrokes.
P	Pointing to a target with a mouse	1.10	Y	It is confirmed by applicability of Fitts' law (described below) with THG.
H	Homing (moving) the hand(s) between devices	0.40	N	Users do not have to operate devices.
$\boldsymbol{D(n_D, l_D)}$	Drawing with the mouse n_D straight-line segments of length l_D	$0.9n_D + 0.16l_D$	P	It needs verification, but it could be used/adapted to predict time of THDG.
M	Mentally preparing for executing physical actions	1.35	Y	It has been used in KLM extensions with the same value.
R(t)	System response time	t	Y	It depends on each system but has been used in KLM extensions.

Various KLM extensions allowing interface assessment can be found in the literature for other types of interfaces, but the use of KLM has not been verified using THG. This model has been successfully used on mobile phone interaction (Holleis, et al., 2007), handheld devices (Luo & John, 2005), touch-sensitive mobile user interfaces (Lee, et al., 2015), in-vehicle information systems (Pettitt, et al., 2007), shared workspaces (Ferreira, et al., 2009), and many more. This fact demonstrates the strength and usefulness of KLM. However, none of these works is applicable in our case because of three reasons. First, keystrokes are not gestures, which is the case of the original KLM. Second, although some KLM extensions consider the use of gestures (such as (Holleis, et al., 2007; Lee, et al., 2015)), these gestures are different from THG. Finally, most operators proposed either in the original or modified KLM versions are not (directly) applicable to NUIs based THG, and hence, their estimated values cannot be used. Beyond these restrictions, the KLM methodology could still be used.

In the case of wanting to apply the KLM methodology to THG, a method to analyze gestures to formulate and use the adapted or new model is needed. Likewise, the operators to be included as part of the model should be carefully chosen. The analysis of the original KLM operators is a good starting point (Table 2.1). ***K*** and ***H*** operators are not applicable due to the actions they represent are not present using THG. ***R*** and ***M*** can be reused, because systems may need a time to response and users require a time to mentally prepare respectively. However, ***M*** perhaps should be reviewed as suggested by MacKenzie (MacKenzie, 2013)(p. 272). The other two operators, ***P*** and ***D***, are not directly applicable because pointing and drawing tasks will be performed in the air instead of using a mouse. Actually, the utility of ***P*** is confirmed by the applicability of Fitts' law to touchless interaction (Schwaller & Lalanne, 2013; Pino, et al., 2013; Polacek, et al., 2012; Zeng, et al., 2012; Sambrooks & Wilkinson, 2013). On the other hand, ***D*** could allow modelling gestures that correspond to figures of letters, numbers, or shapes drawn in the air (i.e., THDG). These ideas are developed in detail in the rest of this thesis.

Fitts' Law

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. It is related to human movement and predicts the time to rapidly move to a target. 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 move the cursor to a target grows logarithmically with distance and decreases with target size (formula 2.1, Figure 2.7), i.e., if a target gets smaller and/or further away, it takes longer to reach that target. From a practical point of view, this model suggests making interface elements large enough because it is difficult for users select on small ones, and/or make them bigger to compensate when they are not placed close. Fairly accurate values for optimal results can be calculated by applying the formula, but it is necessary to know whether Fitts' law is applicable to the intended interaction style.

$$T = a + b * ID \quad with \quad ID = log_2\left(\frac{A}{W} + 1\right) \tag{2.1}$$

Where A is the distance to the target; W is the target width; a and b are parameters computed through empirical tests using linear regression; ID is the Index of Difficulty measured in bits.

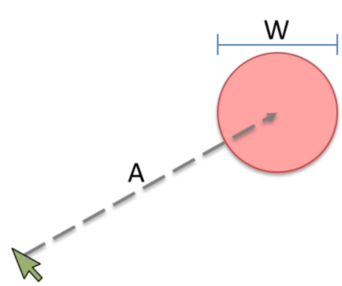


Figure 2.7. Fitts' law.

Fitts' law has been traditionally employed to analyze pointing tasks using a mouse and to compare two or more input devices, but it offers more opportunities. For instance, it has served as a basis to derive new models, such as Steering law (Accot & Zhai, 1997) (described below). Moreover, the utility of Fitts' model has been verified with new interaction styles such as those based on touchscreens (MacKenzie & Teather, 2012) and head mounted displays (Lubos, et al., 2014). In fact, Fitts' law has been successfully applied to touchless interfaces as revealed by a number of studies (Schwaller & Lalanne, 2013; Pino, et al., 2013; Polacek, et al., 2012; Zeng, et al., 2012; Sambrooks & Wilkinson, 2013; Jude, et al., 2014a).

Besides, pointing tasks in gestural interfaces are different than in other interaction styles. Users move a hand to reach a target, and then, they have to perform a gesture-stroke to select the target. Consequently, the formulas should model both the movement time and gesture-stroke time, or alternatively, the movement time and the stroke time can be computed separately.

Despite its demonstrated wide use and the possibility of being used with gestural interfaces, Fitts' law is insufficient to evaluate interfaces based on THG when they involve other gestures than just point and select. For example, a UI may include swipes or circles. Also, those gestures may depend on additional parameters, such as holding time, shape and length. In other words, Fitts' Law usefulness is constrained mainly to pointing tasks. Consequently, other models may complement it, especially in order to analyze interfaces in a comprehensive way.

Hick-Hyman Law

Hick-Hyman law (Hick, 1952; Hyman, 1953), like Fitts' law, is a well-known model in HCI, published also in the fifties. It is a model for estimating reaction time for decision making; i.e., it allows computing the time a person needs to make a choice from several simple options. Formally, the model asserts that the time to make a decision increases logarithmically with the number of possible choices (formula 2.2). From a designer's perspective, this model recommends to minimize the number of options that are present when users have to make a quick decision. Moreover, the design of menus is one of the main applications of Hick-Hyman law (MacKenzie, 2013; Cockburn, et al., 2007). Despite this model can be useful when analyzing and/or designing UIs, its use is limited if employed alone (MacKenzie, 2013)(p. 257). Thus, we consider Hick-Hyman law could be applied to UIs based on THG, but it should be complemented with other models.

$$T = a + b * log_2(n + 1) \tag{2.2}$$

Where *a* and *b* are constants determined empirically; *n* is the number of choices.

Steering Law

Based on Fitts' law, Accot and Zhai derived Steering law (Accot & Zhai, 1997) to be used in the analysis of trajectory-based tasks. It allows predicting the speed and total time to navigate through a two-dimensional "tunnel". The user task consists of traveling from one end of the path to the other as quickly as possible, without touching the boundaries of the tunnel. From a practical point of view, this model suggests designing elements as dropdown menus to be wide and short. One potential HCI application of Steering law is the comparison of performance of different input devices with the aim of selecting the best one (Accot & Zhai, 1999). It also allows modelling user's performance in hierarchical cascading menus where users navigate a pointing device through the menu bounding box (i.e., the tunnel) to select an item (Cockburn, et al., 2007). Moreover, Steering law has been used to analyze trajectory-based tasks performed with the hand in mid-air (Kim & Ren, 2014). However, it may not be adequate to analyze tasks performed in the air when/where there are no visual guides. Actually, execution of gestures should not need a tunnel generally. Then, although Steering law may be not widely applicable to UIs based on THG, it could still be useful for predicting users' performance when they have to move, for instance, a hand cursor from one side to another one passing through a tunnel.

As one could expect, there are available a large number of models that allow analyzing quantitatively different interaction styles. A few examples are the models for predicting performance of unistroke writing (Isokoski, 2001), pen stroke gestures (Cao & Zhai, 2007), and menus (Cockburn, et al., 2007). We study the possibility of applying/adapting/extending some of these models to THDG; chapter 4 provides the details.

2.3.3 Model Validity

Even though modelling is assumed not to be perfect, it is necessary to know how well a model predicts performance time. Empirical validation is usually done to confirm the model quality. One or more applications are needed to carry out this validation. Users complete a different number of tasks using those applications. Then, the required time to accomplish each task is computed and compared against the value previously calculated using the model. This process allows determining the model performance with the use of some metrics.

Several metrics can be used to confirm whether a model makes good predictions or not. The R^2 value is a common validity metric (Cao & 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 a model has good quality as long as their values are within typical ranges in the field. In other words, 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 (particularly KLM (Card, et al., 1980)).

Referring to KLM, Card et al. reported the RMSE was 21 percent of the average predicted execution time (Card, et al., 1980). Other authors have also reached lower error values in works that either extended KLM or developed new models based on it (e.g., (Holleis, et al., 2007; Luo & John, 2005; Lee, et al., 2015; Pettitt, et al., 2007)). Regarding the other selected metric, reported values are generally equal or greater than 0.90 as in the case of the original KLM and its extended versions (Card, et al., 1980; Pettitt, et al., 2007). Other models whose validity has been verified could also be taken into account (e.g., (Cao & Zhai, 2007)), but the aforementioned values are enough to establish our baseline.

Summing up, we will consider that a model predicts performance time in an acceptable way if R^2 and %RMSE are proximate to the corresponding values of the original KLM: $R^2 \geq 0.90$ and %RMSE $\leq 21\%$ (Card, et al., 1980).

Chapter 3

Observing Users Performing Gestures

An important preliminary step towards formulating a quantitative model for NUIs based on THG is the understanding of user's behavior on performing gestures. In other words, it is necessary to understand the manners in which users articulate gestures in the air (i.e., ways to produce gestures). The previous chapter described some of the efforts that have been made in this regard. Despite these efforts, the answers to the following questions are still missing: How do users perform gestures moving their hands in the air? Do they use only hands or also use other body parts? Do they use one or two hands? Do they perform gestures using one or several movements?

Accordingly, the goal of this chapter is to answer these open questions. To do it, gesture articulation is studied by analyzing the relation between gestures and the underlying symbolic patterns (Rekik, et al., 2013). The study is done from a broad perspective, not focusing just on touchless hand gestures (THG). After describing the study, the chapter presents a qualitative model and an embodied taxonomy, which are in turn used to perform a detailed qualitative and quantitative analysis of gesture articulation variability. Some implications for designing applications based on THG are also discussed.

3.1 User Study

Rekik et al. had a similar goal to ours, but they focused on the evaluation of gestures articulation for multi-touch interaction (Rekik, et al., 2013). Since we were interested in touchless interactions, we followed in general their methodology making the needed adjustments.

Our study was composed of two tasks. Like Rekik et al. (Rekik, et al., 2013), the goal of the first task was to familiarize participants with the experimental setup and to analyze their interaction styles using an uncontrolled experimental procedure. The analysis of this data served as a basis to derive both a qualitative model on user conception and production of gestures, and a taxonomy of touchless gestures (TGs). This taxonomy and this model were used to analyze the second task. The goal of the second task, like (Rekik, et al., 2013), was to perform a quantitative analysis of how users articulate symbolic gestures in mid-air by following some specific instructions and exploring several ways to do it. The remaining details of our study are described below.

3.1.1 Participants

Twenty people (mean age = 27.5 years, SD = 4.3, six female) took part in the study. They were invited by mailing lists and social networks. Eighteen participants were right-handed. Graduate students and researchers from Europe, South America, Africa and Asia agreed to volunteer for the study (UI designers were not allowed participating). All participants self-declared their experience on gestures and other demographic characteristics in a final questionnaire. Ten participants had some previous experience on touchless interaction for playing video games (e.g., using Kinect).

3.1.2 Apparatus

The hardware setup, mounted in our laboratory, consisted of a notebook, a Kinect sensor and a display. The Kinect and the display were connected to a notebook equipped with an Intel Core i7 processor, 8 GB of RAM. Participants stood about 2.5 m away from the display (with a size of 1.8×1.4 m and a resolution of 1024×768 pixels). The Kinect was placed below the display at a height of 1 m. Kinect RGB data was used to videotape the interaction and give participants some feedback while performing gestures.

The developed application interface consisted of an augmented video blending UI controls and the real environment. Augmented video was used trying to avoid participant distractions while performing the tasks. It means participants were able to see themselves on the projected display like looking at a mirror. Also, the application showed the instructions to participants (i.e., the name of each required gesture type), the progress, and whether the gesture was considered right or wrong (i.e., a green check or a red “X” respectively). Inputs were considered right if participants followed the instructions correctly. No additional visual feedback was provided to prevent some effect (Rekik, et al., 2013).

3.1.3 Procedure and tasks

Participants were given the exact procedure for each task in paper sheets to avoid instructing them in various manners. They executed the gestures guided by the software when they were ready to start. Participants were also requested and encouraged to think-aloud while performing each gesture in both tasks (i.e., perform each gesture describing it aloud) (Rekik, et al., 2013). Moreover, they had a short rest at the end of each task. Finally, we asked participants to provide some information by filling out a questionnaire. The specific instructions and differences of each task are described below.

Task 1: Open-Ended Gestures

We asked participants to produce as many gestures as possible that came to their minds such as gestures that had a meaningful sense to them or gestures that they would use to interact with applications (i.e., gestures had to be realistic for practical scenarios, easy to produce, easy to remember, and different enough from one another that they could be used for different actions). In addition, participants were asked to describe the gestures they performed using the think-aloud protocol. Participants decided when to start and stop. Thus, this task finished when participants could not produce additional gestures.

Task 2: Goal-Oriented Gestures

Participants received explicit instructions to carry out the second task. They had to create gestures for a set of twenty gesture types (Figure 3.1). The software randomly asked participants to perform a gesture by showing only its name. Paper sheets were presented on demand just for the case in which a participant was not familiar with the corresponding gesture type. Participants had to think of the gesture articulations in which they would produce the gesture type after reading its name. They were instructed to create as many different gesture articulations as possible, trying to increase variety and creativity (Morris, et al., 2014). Participants were free to select the different gesture articulations, but we gave them the requirement that executions should be realistic for practical scenarios, i.e., easy to produce and reproduce later.

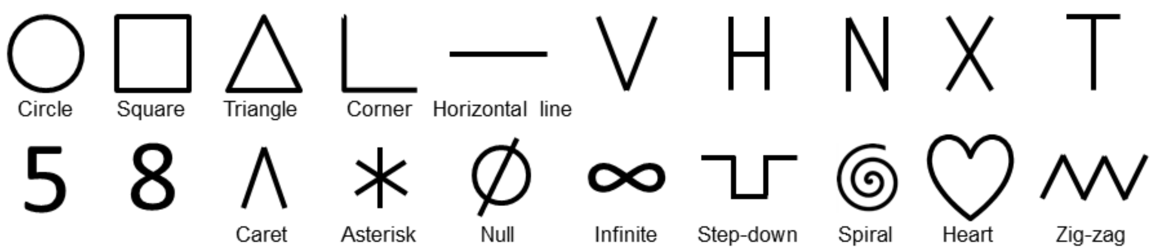


Figure 3.1. The set of 20 gesture types used in the experiment.

In addition, we decided to make use of a Wizard of Oz design (Dahlbäck, et al., 1993; Nielsen, et al., 2004; Henschke, et al., 2015) taking into account that participants were instructed to perform gestures in the articulation they desired rather than being limited to the capabilities of a recognition system. The main idea was participants believed they were interacting "normally" with the system which provided the results/information but responses were in fact given by an experimenter (the "wizard"). Hence, the experimenter pressed some keys providing the system responses according to the participant's input (i.e., start, end, and right/wrong).

3.2 Open-Ended Task Results

The first task of the study consisted in producing gestures by following an uncontrolled experimental procedure. Participants had to appeal to their imagination to carry out the task. Thus, the gathered data allowed analyzing the various gesture articulations conceived and produced by participants.

3.2.1 General Observations

Regarding the collected quantitative data, participants performed a total of 117 gestures in this task. Each participant performed from 3 to 14 gestures (mean = 6, SD = 3, median = 5, mode = 3). We obtained the following features by analyzing all these gestures:

- All participants performed drawing or writing symbolic gestures (such as shapes, letters, or numbers). Furthermore, participants performed gestures for traditional interactions actions. For instance, 12 participants produced gestures for scale, swipe, drag & drop, etc. 14 participants produced gestures for actions like typing in the air, wave, tap, clap, etc. One participant added gestures for selecting a group of objects while another participant added gestures for making a copy or paste actions.

- Participants produced both stroke and hold (keep a pose a short period of time) gestures. For instance, 17 participants performed at least one gesture stroke. Nine participants utilized poses at least once.

- Gestures were produced using one or more body parts. 15 participants executed gestures using both one and two upper limbs (i.e., arms, hands or fingers). From the remaining five, four utilized only one upper limb for all gestures, whereas one used both upper limbs (but he also utilized the whole body in a few cases). Moreover, 4 out of the 20 also employed other body parts (different than upper limbs).

- 18 participants performed gestures using both single and multiple movements. The two remaining participants executed only single movement gestures, whereas nobody used only multiple movement gestures. Multiple movements were either parallel or sequential.

- 19 participants executed most of the gestures starting from a resting position, whereas the remaining one performed 75 % of his gestures continuously, i.e., participants produced the gestures without adopting positions of resting or relaxation between strokes/movements only in a few cases. Resting positions consisted in having both hands/arms down and close to the hips or to the torso in most cases.

3.2.2 GCP: A Model on Gestures Conception and Production

We derived GCP, a qualitative model for gestures conception and production (Figure 3.2), based on both the aforementioned results (Section 3.2.1) and related work from psychology (McNeill, 1992; Kendon, 2004; Kita, et al., 1998) and neuroscience (Wong, et al., 2015). GCP aimed at helping to understand how users conceive and produce gestures. GCP models the needed processes from acquisition to execution of gestures.

According to GCP, a user initially needs to mentally prepare (think) before executing a gesture. During *mental act* phase, which was adapted from (Wong, et al., 2015), the user establishes/defines a motor goal (i.e., "what" processes), and then, s/he specifies the manner in which s/he will achieve that goal (i.e., "how" processes). In other words, mental act consists of *perception* and *gesture planning*.

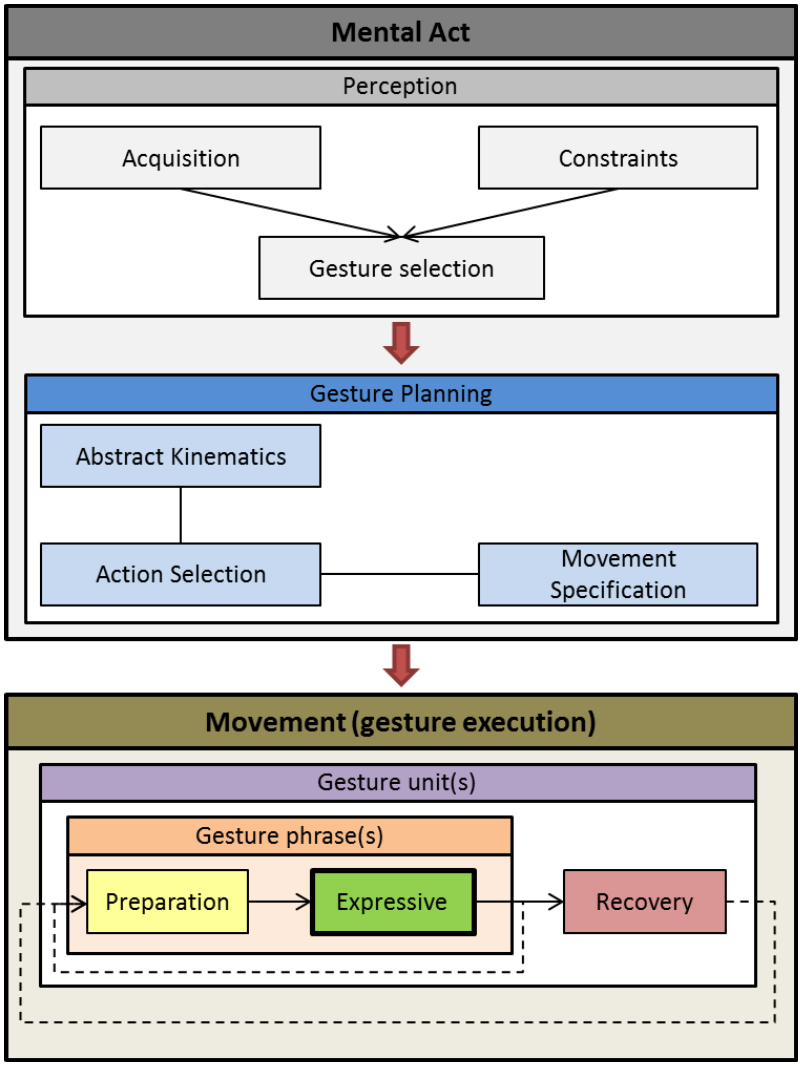


Figure 3.2. GCP, a model on user conception and production of touchless gestures (based on (Wong, et al., 2015; Kendon, 2004; McNeill, 1992; Kita, et al., 1998)).

The user selects/defines/forms motor goals during perception. Perception consists of three processes: (1) acquisition or identification of proposed symbols/referents; (2) application of rules/constraints to perform gestures (e.g., the instructions given in our study); and (3) selection of the gesture to be performed.

Gesture planning in turn refers to how the required gesture will be produced; i.e., it defines the specific movement(s) to execute the gesture. It also involves several processes that may occur in sequence or in parallel. These processes are: (1) abstract kinematics of the gesture (i.e., how the gesture will look), which is optional for single gestures such as pointing; (2) selection of body end-effector(s) action (i.e., how the effectors/body parts will achieve the goal); (3) complete specification of motor commands needed to produce the gesture. The occurrence of these processes allows translating the motor goal into the movement that will correspond to the desired gesture.

Several phases can be observed when the user executes the gesture. Actually, the results obtained in the first task of our study are consistent with the temporal nature of gestures, which is described in terms of phases, phrases, and units (Kendon, 2004; McNeill, 1992; Kita, et al., 1998) (see Section 2.1.1). Thus, the gesture execution starts with an optional physical *preparation* of the effector selected during the mental act. Next, the peak of effort and shape are clearly expressed in the *expressive* phase (*stroke* or *hold*). Finally, the effectors return to their initial or resting positions (*recovery* or *retraction*). Preparation and expression phases are encapsulated in a *gesture phrase*, while recovery together with one or more g-phrases is grouped into a *gesture unit* (see Section 2.1.1).

3.2.3 A Taxonomy of Touchless Gestures

Despite the GCP model allows capturing perception, planning, and execution of gestures proposed by our participants, it is insufficient to reach our goal. GCP only models the execution of gestures in a general way (i.e., looking at their temporal nature), and hence, it does not permit doing a fine grain analysis of gesture articulation. Therefore, we propose an embodied taxonomy of TGs. In general, the different levels of this taxonomy cannot be seen serially or as partitionable attributes because these levels represent indivisible aspects of user gestures.

Table 3.1 depicts the proposed taxonomy. Overall it captures physicality, movement composition, and structure of gestures. Physical level captures the end-effectors/body parts used to perform the gesture. Interestingly and contrarily to multi-touch input (Rekik, et al., 2013), this level does not capture only hand gestures; it considers gestures performed with the whole body as well as with upper and lower limbs with the corresponding subdivisions.

Movement level refers to the set of movements that compose a gesture. When a gesture is composed of more than one movement, these movements can be entered in parallel (i.e., multiple movements are articulated at the same time, e.g., using two hands to draw two sides of a "heart" shape at the same time) or in sequence (i.e., one movement after the other, such as in drawing the "plus" sign with one hand). This sequence may involve one of the following options:

- Single movements; i.e., one movement at the same time.
- Parallel movements. For instance, use two fingers/hands/arms at the same time to draw the two vertical lines of a square, and then, use again two fingers/hands/arms to draw the two horizontal lines.
- Both single and parallel movements. For example, use two fingers/hands/arms at the same time to draw the two diagonal symmetric lines of a triangle, and then, use one finger/hand/arm to draw the horizontal line.

The last level refers to the structure of the gesture. In particular, it captures the state of the articulated movements. Considering this, a gesture may be a combination of single (static) or a series (dynamic) of poses that follow or not a path (like in (Wobbrock, et al., 2009)). Thus, we analyze the other task of the study armed with this taxonomy.

Table 3.1. A taxonomy of touchless gestures.

Embodied Taxonomy			
Physicality	Whole Body		
	Upper limbs	Arms	One-arm
			Two-arms
		Hands	One-handed
			Two-handed
		Fingers	One-finger
			Multi-fingers
	Lower limbs	Legs	One-leg
			Two-legs
		Feet	One-foot
			Two-feet
	Other, (e.g., head)		
Movement	Single		
	Multiple	Parallel	
		Sequential	Sequence of single movement
			Sequence of parallel movement
			Sequence of single and parallel movement
Structure	Static pose		
	Dynamic pose		
	Static pose with path		
	Dynamic pose with path		

3.3 Goal-Oriented Task Results

This section presents the results obtained from the second task of the study in which participants produced various gesture articulations for specific symbolic gesture types.

3.3.1 Gesture variations

Participants were instructed to propose as many articulation variations as possible for each gesture type. We collected 1,237 total samples for our set of 20 gesture types. On the average, our participants proposed 3.1 variations per gesture type (SD = 0.4, see Figure 3.3), a result that is in agreement with the findings of (Oh & Findlater, 2013) for action gestures (mean 3.1, SD = 0.8). A Friedman test revealed a significant effect of gesture type on the number of variations (χ^2 (19) = 96.053, p < 0.001). The "*" and "step-down" gestures presented the lowest number of variations (2.2 and 2.6 variations on the average respectively). The gesture with the maximum number of variations was "X" (3.9 on the average) for which our participants managed to easily decompose it into individual strokes that were afterward combined in many ways in time and space using different gesture physicality and structure (see Figure 3.4). For example, only 3 participants produced less than 4 gesture articulations for "X". These first results suggest that the specific geometry of the gesture enables users with different affordances of how to articulate that shape. Likely, the mental representation of a gesture variation implies a particular type of articulation which is tightly related to the gesture shape.

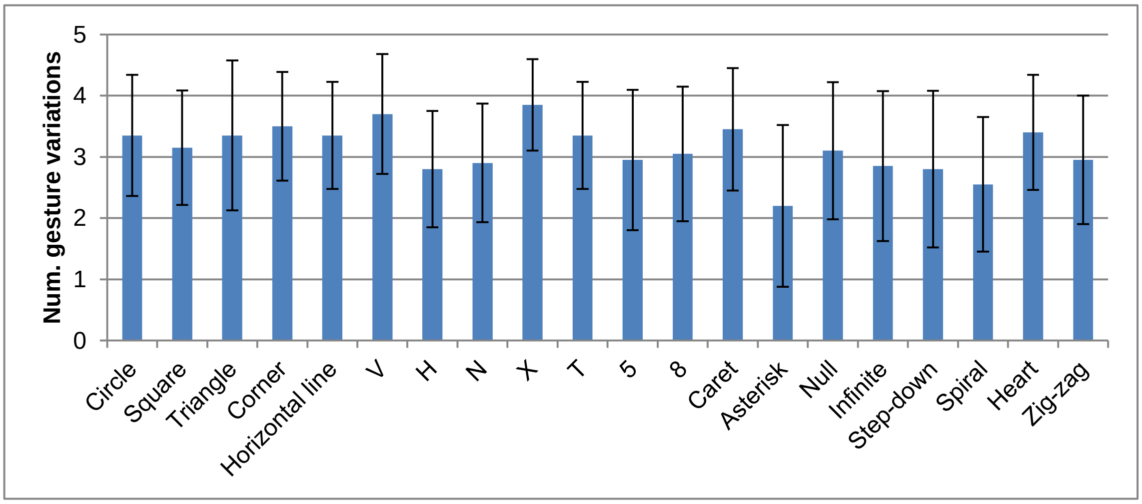


Figure 3.3. Number of variations for each gesture type. Error bars indicate 1 SD.

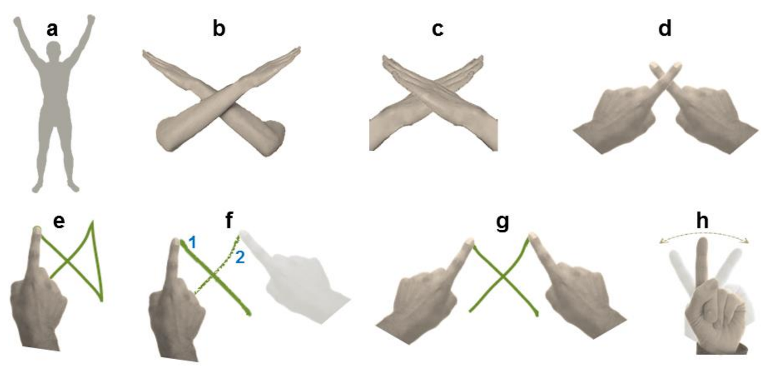


Figure 3.4. Various articulation patterns for the "X" symbol produced with several poses (a-d); number of strokes (e-h), sequential (f), and parallel movements (g, b-d), using the whole body (a), arms (b), hands (c) and fingers (d-h). Numbers on strokes indicate stroke ordering.

We can also remark that for all gesture types the maximum number of variations was 4 or 5, except "triangle" and "V" that had 6 variations. The minimum number of variations was 1, except "square", "corner", "X", and "T" with 2 variations. Meanwhile, the means (averaged over all subjects) of the percentages of the gesture types for which each participant produced at least 4, 3, and 2 gestures were 49 %, 70 %, and 91 % respectively. This result also suggests that, for some users and for some gesture types, the number of gesture articulation variations can be limited which can be explained by the previous practice but also by the geometrical shape of the gesture.

3.3.2 Physicality Breakdown

Figure 3.5 shows the ratios (averaged over all users) of gestures for each gesture type and overall. We used only single levels for arms, fingers and lower limbs to simplify the analysis. A Friedman test revealed a significant difference in the ratios (averaged over all symbols) between the six physicality types (χ^2 (5) = 17.84, p < 0.001). Post-hoc tests confirmed not significant differences only between arms and two-handed levels.

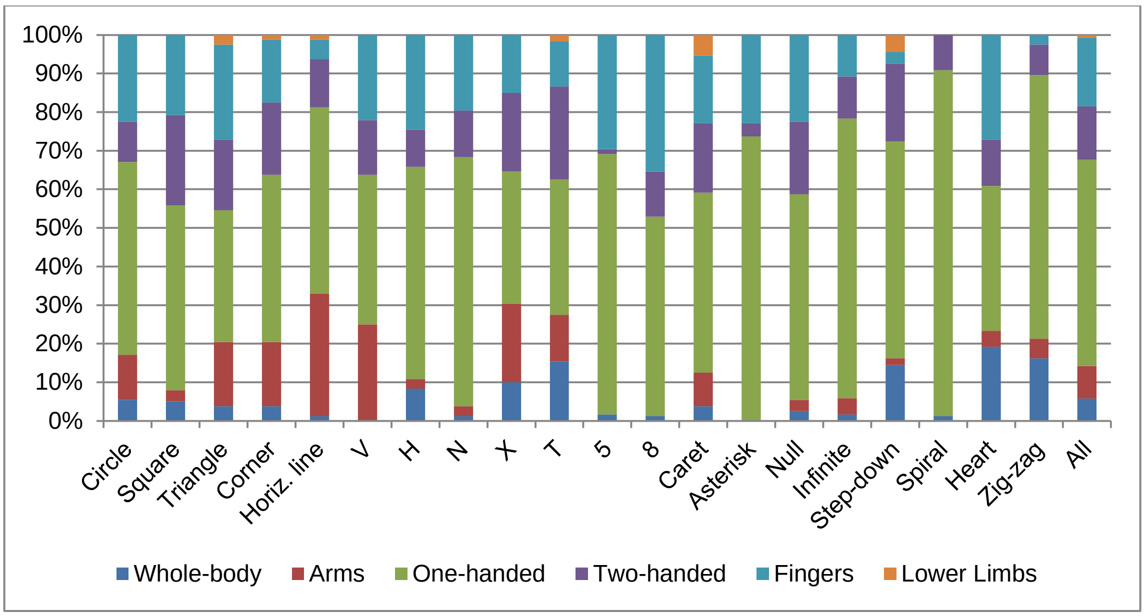


Figure 3.5. Gesture physicality ratio.

Friedman tests also revealed significant effect of gesture type on the ratio of the physicality types (Table 3.2). Referring to each physicality level, participants preferred one-handed gestures in all cases (53.4 % on the average), especially for gesture types that may be considered as more difficult or strange to articulate. This is precisely the case of "spiral" that got the highest value and differed significantly from the other gesture types (except "infinite") according to the corresponding post hoc test. Similarly, the gesture types "asterisk", "infinite", and "zig-zag", that also obtained high values, were not significantly different from one another, and showed differences between the other gesture types (except for "H", "N", and "5"). The next types are gestures made with fingers (one or multiple) and with two hands but with a short difference according to overall values (17.7 % and 13.9 % respectively). Notably, the highest values of finger type were for gesture types that represent numbers (i.e., "5" and "8", with no significant differences between them), which could be attributed to the fact that numbers can be easily represented using fingers. Actually, no significant differences were found between "5" and "8", as well as between them and other gesture types that can also be easily mapped into fingers (such as "circle", "square", "triangle", "V", "H"). Furthermore, finger gestures did not obtain high values for all gesture types (excluding one-handed type). For instance, some gestures were easier to map into two hands (e.g., “square”, “T” and “step-down”) and arms (e.g., “horizontal line”, “V” and “X”). In addition, though whole-body type is represented in a relatively negligible ratio (5.8 %), ratios between 14 % and 20 % were obtained for a few gestures (e.g., “heart” and “zig-zag”). Finally, gestures executed with lower limbs were observed only for six gestures with rates lower than 6 %. Summing up, our participants produced their gestures mainly using one hand, and then, with fingers and two hands.

Table 3.2. Friedman tests for gesture type on level types.

Level	Type	$\chi 2$ (19)	*p*
Physicality	Whole-body	91.40	< 0.001
	Arms	125.84	< 0.001
	One-handed	122.12	< 0.001
	Two-handed	46.92	< 0.001
	Fingers	85.70	< 0.001
	Lower Limbs	39.01	< 0.005
Movement	Single	128.02	< 0.001
	Parallel	144.58	
	Sequential	67.82	
Structure	Static Body pose	75.78	< 0.001
	Static Arm pose	138.53	
	Static Hand pose	91.21	
	Static hand pose and path	102.08	
	One Finger pose	121.77	
	Other	62.92	

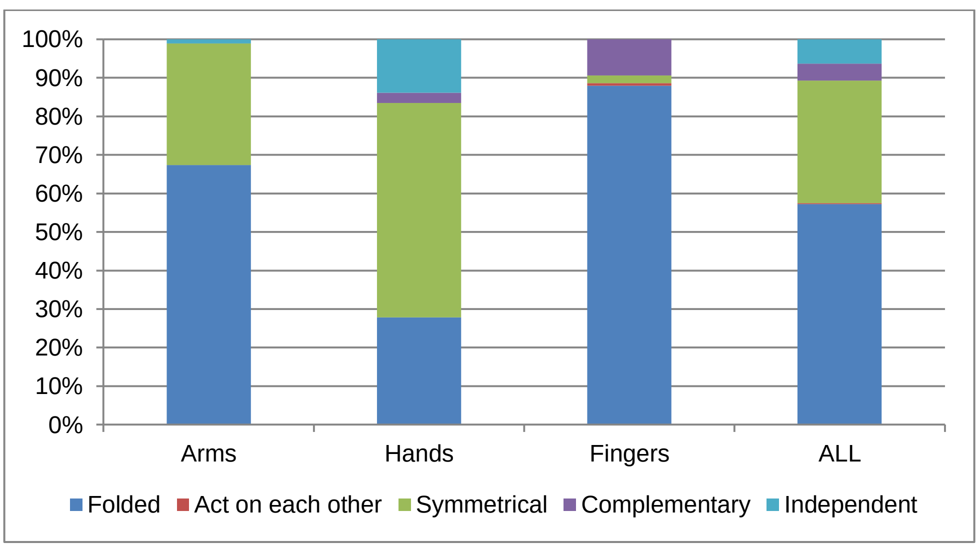


Figure 3.6. Upper limb ratios according to spatial relation.

Although types of gestures produced with more than one upper limb did not get the best ratios, we performed an additional analysis of them given that about one third of gestures fell in these physicality types. 428 gestures performed with two-arms, two-hands, and fingers of both hands were analyzed according to the spatial relation of them. This relation can be as follows: folded (limbs act as a unit), act on each other (limbs act upon each other in a dynamic contact), symmetrical, complementary (e.g., one hand acts as a reference while the other one is moved), and independent. Effectors are in touch in the first two cases, whereas they are separated in the other cases. Figure 3.6 shows the global ratios of the five levels for each kind of effectors and overall. A Chi-square test revealed that the percentage of the spatial relation significantly differed by used effectors (χ^2 (8) = 175.43, p < 0.001). Actually, Figure 3.6 shows participants utilized most often two arms (67.4 %) or several fingers (87.9 %) as a unit, whereas they preferred employing both hands symmetrically (55.6 %). The other levels are negligible (less than 14 % in all cases).

3.3.3 Movement Breakdown

Figure 3.7 shows gesture ratios for each gesture type according to movement synchronization type. The three types of sequential gestures were subsumed under a general sequential type due to the small number of occurrences especially in the case of sequence of parallel movements and sequence of parallel and single movements. A Friedman test revealed significant difference in the ratios (averaged over all gesture types) between the three movement synchronization types (χ^2 (2) = 30.70, p < 0.001). Post-hoc tests confirmed significant differences between all the movement synchronization types.

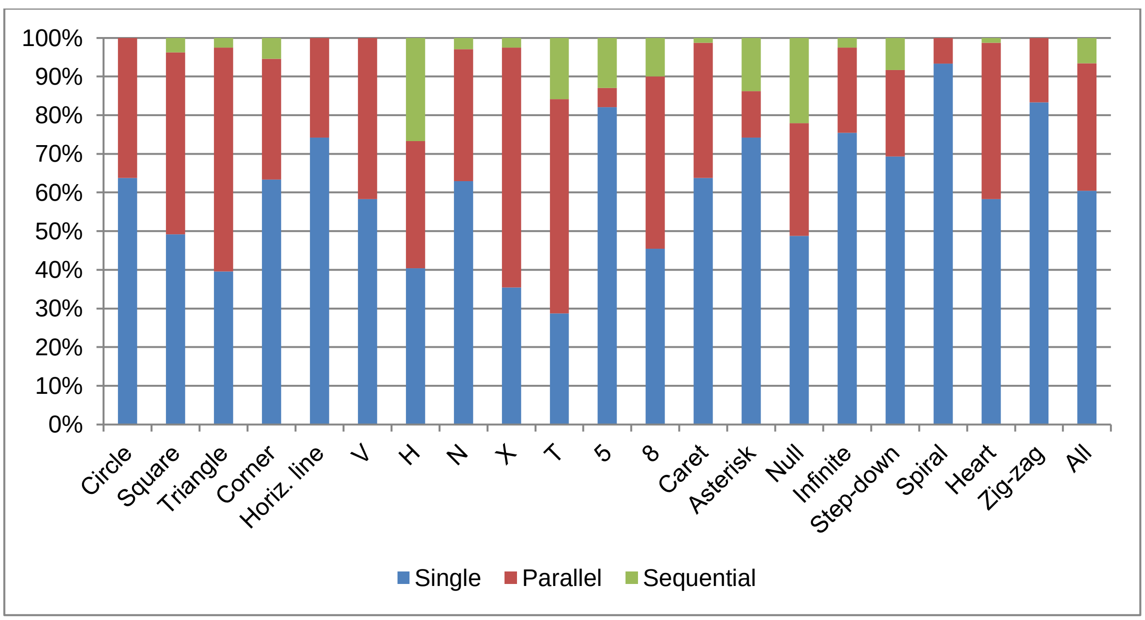


Figure 3.7. Gesture movement composition ratio.

Similarly, gesture types showed a significant effect on the ratio of the three synchronization types as reported on Table 3.2. Overall participants performed more often single gestures; its average (60.5 %) almost doubles the one of parallel gestures. However, parallel gestures were preferred to produce gesture types that expose a symmetry axis (e.g., “X”, “triangle”, and “T”, etc.), a finding in agreement with previous work (Rekik, et al., 2013; Rekik, et al., 2014b), only symmetric gestures can be conveniently parallelized during articulation. Post-hoc tests showed significant differences between these three gesture types ("X", "triangle", and "T") and the remaining ones, except for the following pairs: ("triangle", "rectangle"), ("T", "rectangle"), ("T", "V"), and ("T", "8"). All these gesture types can be produced using two end-effectors with ease, e.g., crossing the arms/hands/fingers to form an "X" (Figure 3.4 b-d). Other gesture types, such as "8" and "square", got in parallel type ratios lower than in single type, but they demonstrated a behavior similar to the previously described for "X", "triangle", and "T". Concerning sequential type, its overall ratio was very small (6.6 %), but non-negligible ratios were observed for three gesture types: "H", "null", and "T". These gesture types showed no differences between them, as well as significant differences between them and the other gesture types (except between "*" and "null", and "*" and "T"). This finding suggests that participants may have produced gestures composed of a sequence of movements only when it was worth doing it; they still preferred the other two types.

3.3.4 Structure Breakdown

This section provides details of gesture articulations based on both the dynamics (i.e., poses and paths) of gestures and the used end-effectors. The five structure types with highest scores were selected, while the remaining levels were grouped into another one labeled as "other". Figure 3.8 shows the corresponding ratios for each gesture type. Significant difference in the ratios between these six structure types was found (χ^2 (5) = 74.11, p < 0.001). Post-hoc tests confirmed not significant differences between static body pose and the levels one finger pose and "other".

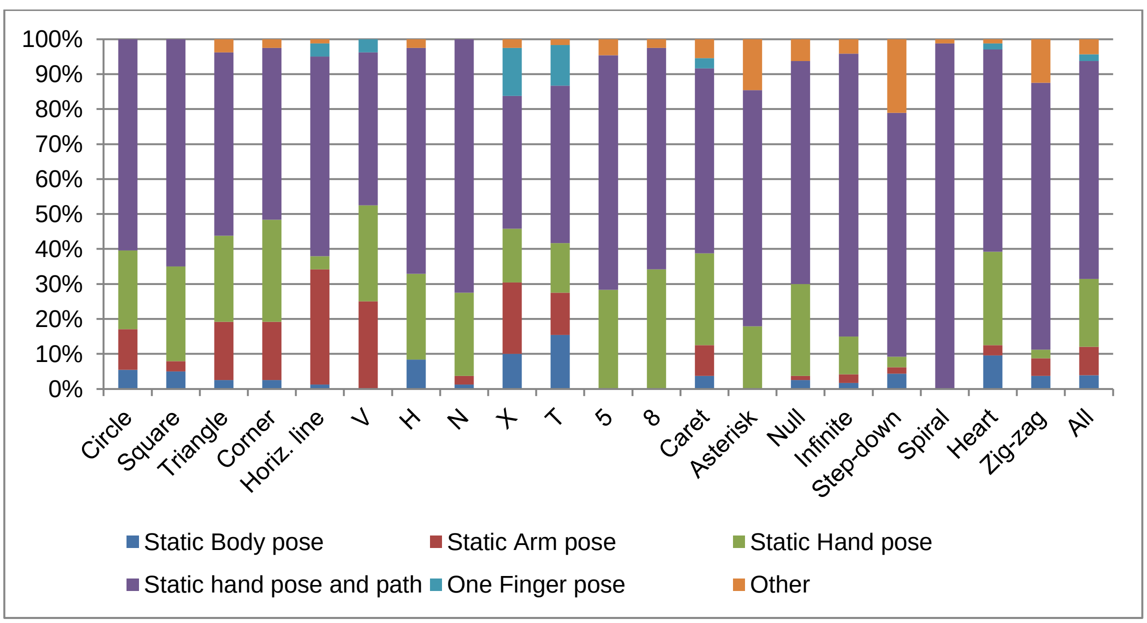


Figure 3.8. Gesture structure ratio.

Likewise, Friedman tests revealed a significant effect of gesture type on the ratio of structure levels (see Table 3.2). For instance, participants preferred gestures made with one hand pose plus path (62.3 %) to produce all gesture types, particularly for those ones that may be considered more difficult or strange (e.g., "spiral", "infinite" and "zig-zag", which showed no difference from one another). Although static hand pose had a relatively non-negligible average ratio of 19.5 %, it got the second place in most cases. It was outperformed by static arm pose to produce two gesture types, "horizontal line" and "X" (there was no difference between them), that could be mapped better into arms as explained above. Moreover, static poses held with the whole body or with one (index) finger of both hands were rarely used. Participants employed them especially for articulating the gesture types "X" and "T" (with significant differences between them and the other gesture types, but no difference within both), which were in fact the gesture types more fairly distributed among the five structure types (excluding "other"). "Other" level was observable basically for "step-down" and "asterisk".

From another point of view, Figure 3.9 shows global ratios for each effector type according to structure levels. It reveals that participants performed few dynamic pose gestures (about 2 %), and they mostly produced gestures that followed a path (63 %) in comparison to only hold poses (37 %). In other words, the participants most often executed their gestures by holding a single (hand) pose while drawing the corresponding gesture type.

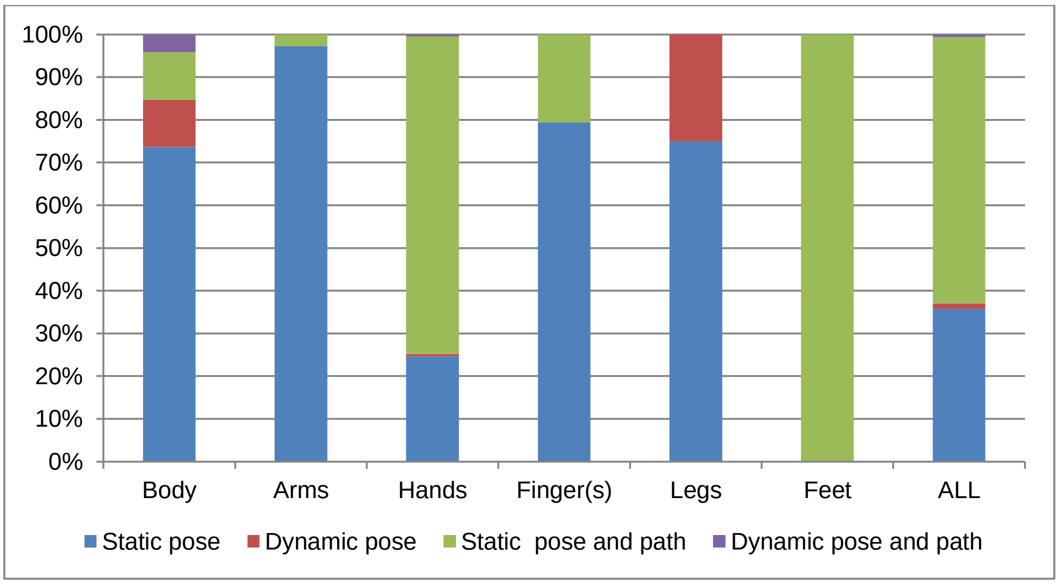


Figure 3.9. Gesture structure ratio by used effectors.

3.3.5 Mental Model Observations

At a high level, GCP facilitates the analysis and understanding of the second task. A user doing the task identifies the current symbol and defines a gesture for it (see Figure 3.2). Next, s/he executes that gesture departing from a resting position, and returning to it or going to another one. Overall we observed that participants executed their gestures in this manner, both gestures composed of single movements as well as those ones composed of multiple movements. Actually, only one subject tried to perform the gestures consecutively in most cases, i.e., without having a retraction.

Contrary to a general/common participants' behavior, there are several particular observations that are worth mentioning:

- **Gesture shape complexity influences touchless input.** Overall participants were able to produce various articulations for the predefined gesture set, but they felt less creative for gesture types with complex/strange geometry (e.g., “spiral”). Two participants proposed exclusively drawing gestures in all cases. Nonetheless, they still used both hands and/or sequential movements to “draw” their symbolic gestures.

- **Preference for vertical plane.** While we did not constrain participants on the direction of the plane when articulating a gesture, all participants performed their gestures in the vertical plane. As an exception, one participant executed some gestures in the horizontal plane.

- **Gesture position, size and direction can be a source of variation.** Several participants in some cases changed the used hand, starting point, size and/or direction of paths to produce various articulations.

- **Mentally embodied gestures.** Though we instructed participants to articulate a gesture for each gesture type, some of them gestured in ways current technologies could not detect. One participant counted the number of fingers to define a number. Another participant touched his heart to define the “heart”. One user walked by making three steps to define the “zig-zag”. Furthermore, a participant proposed a few gestures by drawing a part of the shape, and next, performing another movement to deform it and get the desired figure. For example, he drew a horizontal line with a hand, and then, he put his hands apart on the line and moved them down to form the step-down symbol.

3.4 Discussion and Design Implications

The results described above provide evidence on how users articulate gestures. As a first point, several previously proposed taxonomies may be used to make comparisons with our results. These taxonomies were proposed as part of studies on user-defined gestures for specific scenarios such as virtual reality (Piumsomboon, et al., 2013), humanoid robot (Obaid, et al., 2012), and storytelling (Kistler & André, 2013). All these three works reported that participants proposed more dynamic gestures (or strokes) than static gestures (or holds), which is consistent with our results. Equally, (one) hand was the most used end-effector in both our study and the other two studies that considered various body parts (i.e., (Obaid, et al., 2012; Kistler & André, 2013)). Lastly, we found a high preference towards static hand pose with path gestures similar to (Piumsomboon, et al., 2013).

Although our study had other focus, we might also do a comparison with gesture elicitation studies, namely referring to production of gestures to enhance this type of studies. Morris et al. (Morris, et al., 2014) suggested using five gestures, whereas Hoff et al. (Hoff, et al., 2016) advised that requesting participants to produce more than three gestures would impact on practical utility of gesture elicitation. Though our participants were shown symbol names instead of desired effects of actions, they proposed at least three “natural” gestures for 70 % (SD = 30 %) of the utilized symbols on the average. Our study also included a first task in which the participants performed several gestures in a free manner, which could be comparable to priming. However, our participants were not able to reach the threshold of three gestures in all cases despite of this “priming”. In conclusion, this finding suggests that proposing more than three gesture articulations can be not natural, which is coherent with (Hoff, et al., 2016) but different than (Morris, et al., 2014).

The reported findings also have implications for designing applications based on touchless gestures, particularly with the aim of enabling several gestures to one command. Interfaces based on gestures often use an association between one gesture and one meaning. However, UI designers may alternatively use a many to one mapping that better reflect the high variability of user gestures due to users may perform a specific gesture in different manners expecting it has equal meaning. Therefore, informed by our findings, we are able to outline a set of guidelines for designing TGs interfaces that address TG ergonomics, design and recognizers with the aim of enabling several gestures to one command.

3.4.1 Touchless Gesture Ergonomics

Our findings indicate that strokes are preferred to poses to articulate gestures. These strokes were especially expressed by following a path, mostly with hands. Our findings also demonstrate that producing paths with hands matters to users more than the posture maintained while they do it. The participants generally kept the same pose while executing a gesture; i.e., they rarely used more than one posture between different strokes. Specific hand postures would be needed if users should discriminate between drawing paths and displacing the hand to the point where the path (or a part of it) starts (i.e., the system is not capable of doing this distinction automatically), for example, to perform multi-stroke gestures. Likewise, and contrary to the findings for multi-touch gestures (Rekik, et al., 2014b), paths performed with a variable number of fingers should not be a problem because users would adopt a single pose (e.g., putting together middle and index fingers, or touching the tips of thumb and index fingers).

Despite the high preference for drawing gestures reported here, we do not advocate that gestures based on poses should not be employed. They have proven to be useful in various scenarios (e.g., fingerspelling (Sridhar, et al., 2015)), but an additional issue is that postures should be learned to interact with applications (Ismair, et al., 2015). Beyond this possible limitation, our results show that static poses maintained not only with hands/fingers may be suitable just for gesture types that could be easily produced through this structure (e.g., letter “X” or “T” and numbers). This finding indicates that the use of postures would depend on the facility to map gestures into the corresponding end-effectors. On the other hand, our results suggest that users would prefer static poses instead of dynamic poses, given that the second ones were very scarce in the second task.

Gestures expressed using either single strokes or single holds should be preferred according to the results of the second task. Although parallel movements may be used as a complement, sequential movements may not result “natural” to users. Unexpectedly, when participants produced the candidate multi-stroke gestures (i.e., gestures for symbols “H”, “X”, “T”, “asterisk”, and “null”) using static hand poses with paths, they did it frequently using single strokes. Furthermore, participants did not worry or notice a need to discriminate between drawing paths and just moving hands.

Concerning used end-effectors, participants clearly preferred employing hands and fingers to produce symbolic gestures. Despite the high tendency to use one hand, the presence of two-handed, two-arms, and multi-finger gestures was also noticeable. Our findings suggest that users would use both hands symmetrically (principally to produce static hand pose with path gestures) more than folded hands. Conversely, two arms and several fingers would be mainly used in touch acting as a unit. Moreover, unlike previous work (Aigner, et al., 2012; Rekik, et al., 2013), we observed no two-handed gestures in which a hand was used as a reference and the other one was used to express the gesture. Additionally, overall the use of arms or whole-body to execute gestures would be preferred depending on the ease to map gestures into them as mentioned above. Finally, more gestures executed with feet may have been expected, but they were hardly ever used by participants. This may have happened due to any of these causes: the instructions were insufficient, the participants did not imagine these gestures, or simply, foot gestures were not good enough or "natural" to participants.

3.4.2 Touchless Gesture Design

Our findings show that inferring flexible input when articulating TGs would be more suitable when users are provided with little to no instructions or when symbols are unfamiliar or difficult to them. Otherwise, UI designers and researchers should observe how users articulate a gesture set before designing it. Familiar shapes should also be preferred to unfamiliar ones, and gesture articulations should be connected to users' previous gesture practice whenever possible. Additionally, gesture shapes with complex geometries should be used with care, and learning and memorization should be integrated into the design of such gesture shapes. Furthermore, the available methods to analyze the difficulty of symbolic gestures (e.g., (Rekik, et al., 2014b; Vatavu, et al., 2011; Erazo & Pino, 2014)) should be taken into account during the design.

3.4.3 Touchless Gesture Recognizers

Many of the gestures we witnessed had strong implications for TG recognition technology. Our results demonstrate that UI designers and researchers should design flexible recognizers that are invariant to users' preferred articulation patterns. TG recognizers should be trained with different articulation patterns in terms of physicality, synchronization, and structure. For example, for the same gesture type, our participants articulated it using different number of strokes that are combined sequentially or in parallel using arms/hands/fingers etc. and mixing path and pose structures, such as (Kratz & Rohs, 2011; Anthony & Wobbrock, 2010; Vatavu, et al., 2013; Rekik, et al., 2014a).

3.5 Conclusions

This chapter has analyzed the manners in which users articulate touchless gestures. The conducted study addressed gestures performed with various body parts instead of focusing only on hand gestures. Its two tasks allowed providing new findings and confirming previous ones. First task results were especially used to derive a qualitative model and an embodied taxonomy of gestures. The model was used to analyze the gestures performed by participants in the second task, but it should be also useful to analyze not only symbolic gestures. Though the taxonomy may be generalized (e.g., by including a "nature" category (Wobbrock, et al., 2009)), it was good enough to analyze participants' articulation variability.

Our findings indicate that, additionally to hands, users would use other body parts to articulate gestures if the proposed gesture type can be mapped better into other body parts to hold postures. Similar to multi-touch input [27], gestures in mid-air could be articulated with single as well as multiple movements entered in sequence or in parallel. Our findings also suggest that users would prefer producing gestures in mid-air mainly using one hand to iconically describe single motion paths. This preference does not mean that users would not produce gestures in other articulations.

These findings are important in the context of proposing new interaction techniques that make use of the variability of user gestures, and hence, this study is a first step towards enabling designers to use more than one gesture for a same command. These many-to-one mapping should lead to better user interaction experiences by giving more flexibility and avoiding penalizations when gestures are executed in various manners. However, we will move on to predictive modeling of gesture performance instead of studying in further detail the association between gestures and commands.

Chapter 4

Estimating Production Time of Touchless Hand Drawing Gestures

Static hand pose(s) with path gestures have proven to be useful as reported in the previous chapter. We refer to them as touchless hand drawing gestures (THDG) because they usually consist of drawing figures of shapes, letters, numbers, etc. in the air. Some uses of THDG are "writing" in the air or sketching, and commands or shortcuts to execute actions (Ismair, et al., 2015; Vatavu, 2012). Possible scenarios of application are museums, education and interaction with TVs (Mehler, et al., 2014; Erazo, et al., 2016; Vatavu, 2012). For instance, TV users could draw a letter "M" to show or hide the menu, or a question mark ("?") to invoke system help (Vatavu, 2012). The use of these gestures is an approach different than and/or complementary to pointing tasks. Therefore, modelling this type of gestures requires either extending existing models with new parameters or developing whole new user models.

Considering this landscape, this chapter[5] studies existing models with the aim of encompassing THDG (according to objective O1). We refer to those models that have been used to estimate production time of drawing gestures on other interaction styles, such as mouse and pen interactions. Then, this chapter tries to answer the following questions: Which candidate models can be used? Are the metric values calculated for each model similar to those obtained in the corresponding original versions? (i.e., are the results acceptable?) Which is the best candidate model to make predictions?

The analysis of selected models is performed using each model as it was originally proposed and making modifications to adapt it. Accordingly, the chapter starts describing the candidate models found in the research literature. Next, the corresponding formulas are defined and the required values/constants are computed from parameter estimation phases. Thus, all models are evaluated and compared.

[5] Most part of this chapter was published as a conference contribution (Erazo, et al., 2015).

4.1 Candidate Models

Some interactions like pointing and clicking can be done both with a mouse and in the air using hand gestures, and therefore can be modeled using for example Fitts' Law (see Chapter 2). However, more complex interactions such as those based on THDG require the use of other models. For that reason, we performed a bibliographic review with the aim of finding models that may be extended or adapted to this kind of interactions. Thus, we found three candidate models for analyzing or evaluating drawing gestures in other types of interfaces: Isokoski's, CLC, and ***D*** operator of KLM.

Isokoski (Isokoski, 2001) introduced a conceptually simple model that predicts production time for unistrokes in pen-based interactions done by expert users. According to this model, a gesture is first decomposed into a number of "needed straight-line segments" which are then counted to estimate the overall time-complexity of the gesture. The number of considered segments is the minimum necessary to make the gesture recognizable. Additionally, it is assumed that drawing a straight-line segment takes a constant time.

By comparing the model with real-user interactions, Isokoski (Isokoski, 2001) measured the strength of the relationship between estimated and observed times (R^2) to be <0.85. The percentage root mean square error (%RMSE) of these measures was 30 % (Isokoski, 2001).

Although the definition of "needed straight-line segments" is ambiguous (Cao & Zhai, 2007; Vatavu, et al., 2011) (a procedure describing the reduction of gestures with curves into straight lines is missing), Isokoski's model seems conceptually easy to extend to hand gestures because it requires estimating the constant time to produce a straight line gesture segment from experimental data.

Cao and Zhai (Cao & Zhai, 2007) suggested a model to estimate the production time of single pen-stroke gestures. The model considers three features found in pen-stroke gestures: Curves, Line segments, and Corners (the reason why the model is referred to CLC).

For any gesture, the production time is calculated by summing up the estimated time durations of all gesture segments (see formula 4.1 below). The estimated production times of Curve and Line are defined in formulas 4.2 and 4.3. (Formula 4.4 can be used instead of formula 4.3.) The Corner, which is an abrupt change in stroke direction, was discarded by Cao and Zhai after empirical studies showed its insignificant impact on production times (Cao & Zhai, 2007).

$$T = \sum T(line) + \sum T(corner) + \sum T(curve) \tag{4.1}$$

$$T(curve) = \frac{\alpha}{K} r^{1-\beta} \tag{4.2}$$

$$T(line) = mL^n \tag{4.3}$$

$$T(line) = aL + b \tag{4.4}$$

Where: α is the sweep angle; r is the radius of the arc; β and K are empirical constants. L is the length of the line; a, b, m and n are empirical constants.

The CLC model reveals a strong relationship between estimated and observed times ($R^2 > 0.90$). Even though the model has been used in several research studies (e.g., (Vatavu, et al., 2011; Tu, et al., 2012)), we do not have evidence that it has been used beyond pen-stroke gestures.

A third model to be analyzed is the Keystroke Level Model (KLM) (Card, et al., 1980). As explained in Chapter 2, it is a well-known user model that defines a set of primitive operations. The $\boldsymbol{D}$ primitive is the only one relevant to this chapter, in spite of having several constraints worth noticing: drawing is done with the mouse, it only concerns straight-line segments, and it is assumed to be done on a square grid with 0.56 cm. According to KLM, the production time of a drawing interaction is defined as a linear function of the number of segments (n_D) and the total length (l_D) of all segments (see formula 4.5) (Card, et al., 1980).

$$D(n_D, l_D) = an_D + bl_D \tag{4.5}$$

Where: a and b are constants (a = 0.9 and b = 0.16 in the original KLM version (Card, et al., 1980)).

All in all, the Isokoski's, CLC and KLM models advocate: (1) a gesture is not an atomic entity, i.e., it can be decomposed in components, (2) decompose a gesture either into a series of straight-line segments (Isokoski's and KLM) or into a series of curved and straight-line segments (CLC); (3) use a set of formulas and parameters to calculate the production time of each segment, adding them to obtain the overall production time of a gesture; and (4) derive the parameters from empirical studies with various users, so that the formulas adjust to reality. Nevertheless specific parameters for touchless hand gestures do not currently exist and will have to be further researched. We detail that process in the following section.

4.2 Hypotheses and Research Design

As mentioned above, there is no evidence that the three described models may be used for touchless hand gestures. Therefore, the following hypothesis should be tested:

H1: CLC, Isokoski's and KLM models can be adapted to predict the production time of hand gestures.

Table 4.1. Research design: adaptations of models (A) and experiments (E).

Steps	Models		
	CLC	Isokoski's	KLM
1. Definition of formulas	A1	A2	A3
2. Estimation of parameters	E1	E1	E2
3. Evaluation of models	E2	E2	E3

The starting point to test this hypothesis is the definition of formulas. Either the original formulas can be applied to hand gestures or they have to be extended to encompass the new conditions imposed by hand gestures. Hence, we introduce a first step in our study where the formulas are evaluated and adapted if necessary. This step is shown in Table 4.1 (step 1) and denoted as "A*n*" (A - Adaptation).

After adapting the formulas, it is necessary to define new parameters for hand gestures. The second step requires carrying out several experiments with real users (E*n* in Table 4.1, step 2; E - Experiment) and then tuning the parameters so the models may reflect the users' performance.

At this point, the models should be ready to use. However, we still have to consider the quality of the estimations. Thus, we have to evaluate the models, i.e., to verify the second hypothesis:

> *H2: The adapted models can predict the production time of hand gestures with acceptable quality.*

We will test this hypothesis using the two metrics also adopted by Cao and Zhai (Cao & Zhai, 2007), considering both the strength of the relationship between estimated and observed times (R^2), and percentage root mean square error (%RMSE). Furthermore, we will consider that a model has acceptable quality if R^2 and %RMSE are near the values obtained by Cao and Zhai (Cao & Zhai, 2007): $R^2 > 0.90$ and %RMSE < 30%.

H2 must be tested with a set of experiments with real users (E*n* in Table 4.1, line 3). With a careful experimental setup, some of the experiments required to test *H1* can also be used to test *H2*. That is the reason why we see in Table 4.1 that E1 and E2 are shared by steps 2 and 3. This is possible because: (1) the parameters required by CLC and Isokoski's models can be tuned using the same experiments; and (2) we use two different sets of users in E1 and E2, so that the users being used to tune a model can be reused to validate another model, but one set of users is not simultaneously used to tune and validate the same model.

In the remainder of this section we give more details about the experimental setup. The validation of *H1* and *H2* is discussed in detail in the following sections using the stepwise structure described in Table 4.1.

4.2.1 Apparatus and Method

The hardware setup for the experiments consisted of notebook, Kinect sensor and TV screen mounted in a controlled laboratory setting. The notebook was equipped with an i7 processor and 8GB of RAM. The Kinect sensor was used with a refresh rate of 30 fps, connected to track users' hand position and recognizing gestures, placed at a height of 0.9 m and below the TV screen. The TV screen had 42in, 1360×768px resolution. The participants stood in an uncluttered space, 2.5m away from the Kinect sensor.

A custom software tool (Figure 4.1) was developed for precisely controlling the experiments. The tool was developed using MS Visual C# and Kinect for Windows SDK V1.8 on Windows 7. The tool logged time marks and hand coordinates while a participant performed a gesture. The Dynamic Time Warping algorithm (Sakoe & Chiba, 1978) was used for gesture recognition. The tool interface consisted of an augmented video blending user interface controls and the real environment. Augmented video was adopted in order to avoid participants' distractions while performing the tasks. For instance, a person may judge his/her movements based on a hand cursor and try to make adjustments (Sutter, et al., 2008), especially because of the sensor noise (Livingston, et al., 2012; Sambrooks & Wilkinson, 2013; Zeng, et al., 2012) which should be avoided. We fine-tuned these experimental conditions through a set of trial experiments.

The tool had two additional software modules focused on gesture analysis. The first module allowed recording gestures (Figure 4.1a), while the second one was able to reproduce every user-generated gesture using segmentation and logged hand coordinates (Figure 4.1b).

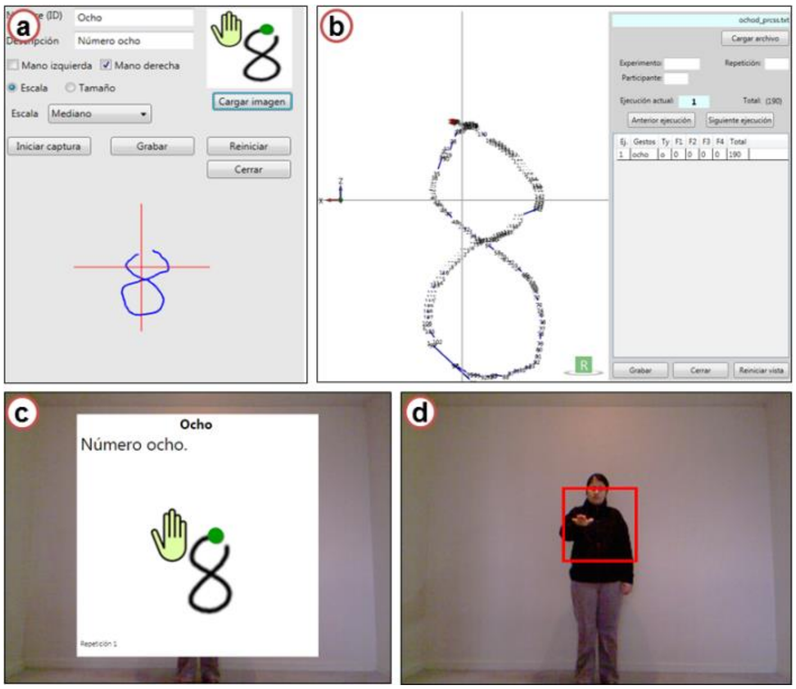


Figure 4.1. Interface of the experimental software.

Each gesture instance was segmented in the phases proposed by (McNeill, 1992; Kendon, 2004) to measure the production times of gestures. More specifically, the measured stroke-phase time was defined as corresponding to production time, which does not account for the time spent by the participants in other gesture phases.

The participants in the experiments were University students (33 in total, aged between 17 and 28) invited by email, social networks, etc. The participants were not paid for their participation. Written informed consents were obtained before starting each experiment. A student was allowed to participate in a single experiment.

Before the experiment, each participant received written instructions and an explanation about the research goals. Then, the participant performed some training gestures guided by the software. When the participant indicated s/he was ready, more specific instructions appeared on the TV screen using a PowerPoint slideshow. Enough time was allowed to read the instructions. The instructions required that every gesture should be done inside a red box (gesture input area or gesture space, see Figure 4.1d), having approximately the same size, and balancing speed and accuracy. (The tool adjusted the size of the input area according to the size of the required gesture.) The instructions also noted the participants should use the dominant hand, and should start (preparation phase (McNeill, 1992; Kendon, 2004)) and finish (retraction or recovery phase (McNeill, 1992; Kendon, 2004)) a gesture with both hands in a relaxed position below hips.

The tool immediately started the data acquisition phase after displaying the instructions. The tool was programmed to randomly pick a gesture within a gesture set and display it for two seconds (Figure 4.1c). The gesture image was displayed along with a name and a very short description. After the two seconds period, the description disappeared, the red box was displayed, and the participant's gesture was collected (Figure 4.1d). When the gesture was correct, the tool displayed a green check and moved on to the next gesture. When a gesture was wrong, a red "X" mark was displayed, the input was discarded, and the participant had to re-enter the gesture.

Besides a practice session, every experiment included three blocks with gestures to be performed by the participants (e.g., a block only including straight lines). The specific characteristics of these blocks are defined in Sections 4.4 and 4.5. The tool included a resting period between blocks of gestures.

4.3 Definition of Formulas

This section discusses the definition and adaptation of each model to hand gestures. The tuning and validation of formulas are explained later.

4.3.1 CLC Model

Although Cao and Zhai adopted several formulas (4.1, 4.2, 4.3 and 4.4) for the CLC model (Cao & Zhai, 2007), there are other options that might be used to improve predictions. Based on regression analysis (discussed later), we suggest that formula 4.6 can be used instead of formula 4.2 to estimate the production time of curves. Another simpler formula that may be applied to curves, which Cao and Zhai did not test, is a linear function of the curve's radius and angle (formula 4.7). These two formulas may contribute to reduce the %RMSE of CLC, but they should be tested against the original one.

$$T(curve) = \frac{\alpha^{a}}{K} r^{1-\beta} \tag{4.6}$$

$$T(curve) = m\,r + n\,\alpha \tag{4.7}$$

Where: α is the sweep angle; r is the radius of the curve; a, β, K, m and n are empirical constants.

4.3.2 Isokoski's Model

Given the conceptual simplicity of Isokoski's model (Isokoski, 2001), we consider it may be straightforwardly adapted to hand gestures. Formula 4.8 may be applied bearing in mind that hand gestures will be reduced into a sequence of straight lines and an empirical constant is necessary to estimate the time taken to generate every straight line. This reasoning has two implications. First, Isokoski did not provide a constant time for performing a straight-line segment, so we have to estimate that constant. Second, Isokoski does not provide a clear procedure to reduce curves into straight lines, which may range between 1 (too much error) and an arbitrary large number (less error but more difficult to calculate). We adopt the procedure suggested by Vatavu et al.: "if the angle α inscribed by an arc was greater than 270°, then use 3 segments; if α < 120°, then use 1 segment; otherwise use 2 segments" (Vatavu, et al., 2011) (p. 97).

$$T = \#\text{segments} * \text{constant\_time} \tag{4.8}$$

4.3.3 KLM

As already mentioned, we are only using the ***D*** operator of KLM, and hence using formula 4.5 to estimate production time of hand gestures. Analyzing the original definition in detail (Card, et al., 1980), we note that formula 4.5 may not be applicable to gestures with curves and corners. We suggest that curves be approximated to straight-line segments by applying the procedure adopted by Vatavu et al. (Vatavu, et al., 2011) (p. 97). Regarding corners, we suggest counting the number of corners (n_C) multiplied by an empirical constant, as shown in formula 4.9.

$$D_c(n_D, l_D, n_C) = a\, n_D + b\, l_D + c\, n_C \tag{4.9}$$

Where a, b and c are empirical constants.

4.4 Estimation of Parameters

In this section we give further insights about formulas 4.6 to 4.9 delineated above. The discussion is organized in two steps. In the first step, we describe the experiments (E1 and E2, see Table 4.1) conducted to obtain empirical data about user's hand gestures. In the second step we present the final formulas with the estimated empirical constants.

4.4.1 CLC Model

Experiment

We repeated most of the experimental process described by Cao and Zhai (Cao & Zhai, 2007) using hand gestures to obtain empirical constants for formulas 4.2, 4.3, 4.4, 4.6 and 4.7. Experiment E1 involved gathering data for three gesture components: straight lines, curves and corners. This experiment then implied configuring the tool discussed in Section 4.2 to request participants to produce variations of these individual gesture components (see below). Each participant would produce the same gesture three times in order to increase precision. Trying to avoid learning and/or sequence effects, the order of components was counterbalanced.

Various lengths (L = {0.4, 0.6, 0.8} meters in motor space) and orientations (0, 45, 90 and 135 degrees counterclockwise) were tested when producing straight lines. For curves, various radiuses (r = {0.2, 0.3, 0.4} meters in motor space) and sweep angles (α = {90, 180, 360} degrees) were tested. Start angle (90 degrees) and direction (clockwise) were treated as control variables for curve gestures. Various corner angles (θ = {45, 90, 135} degrees) and directions (CW and CCW) were tested to produce corners. Length was kept constant (0.6 cm in motor space). Twelve persons participated in E1 (mean age 21y, $\sigma = 2$).

Results

In general, the results obtained from E1 for the CLC model have similar significance to those obtained by Cao and Zhai (Cao & Zhai, 2007), which provides a first indication that CLC can be used for estimating production time of touchless hand gestures. In detail:

Straight Lines. We observed statistically significant differences when measuring production time and varying length ($F_{2,22} = 10.47$, $p < 0.05$) and orientation ($F_{3,33} = 2.92$, $p < 0.05$). No significant length × orientation interaction effects were found ($F_{6,66} = 0.32$, ns). Figure 4.2 shows the relation between length and production time for each orientation. We note that Cao and Zhai (Cao & Zhai, 2007) did not take into account orientation in their estimations because its effect was considered smaller than length. We made the same decision due to the similarity of our results, but we also computed the correlation coefficients to confirm it. We found no correlation between orientation and time production ($r = -0.022$). Finally, performing regression analysis of our experimental data, we obtained the empirical constants shown in formulas 4.10 and 4.11.

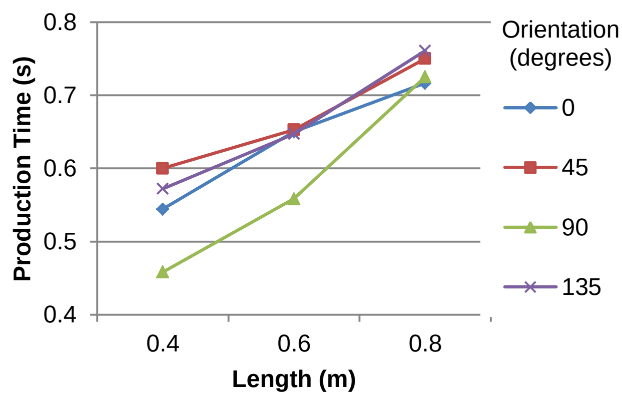


Figure 4.2. Straight line production time.

$$T(line) = 0.486\,L + 0.345 \qquad (R^2 = 0.796) \tag{4.10}$$

$$T(line) = 0.803\,L^{0.442} \qquad (R^2 = 0.746) \tag{4.11}$$

Where L and T are given in meters and seconds respectively.

Curves. As we expected, measured differences in production time were statistically significant for both radius ($F_{2,22} = 12.33$, $p < 0.05$) and angle ($F_{2,22} = 110$, $p < 0.05$). We found significant radius × angle interaction effects ($F_{4,44} = 3.72$, $p < 0.05$). Figure 4.3 shows the relation between sweep angles and production time for each orientation radius. After performing regression analysis with our experimental data, we obtained formulas 4.12, 4.13 and 4.14 below.

$$T(curve) = \frac{\alpha}{1.939} r^{1-0.711} \tag{4.12}$$

$$T(curve) = \frac{\alpha^{0.615}}{1.249} r^{1-0.711} \qquad (R^2 = 0.919) \tag{4.13}$$

$$T(curve) = 1.338\, r + 0.236\, \alpha \quad (R^2 = 0.942) \tag{4.14}$$

Where α, r and T are given in radians, meters and seconds respectively.

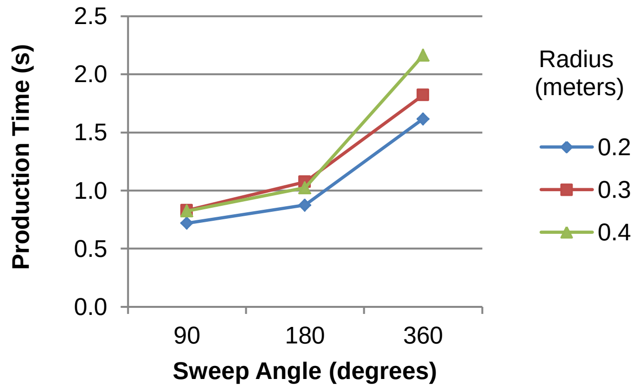


Figure 4.3. Curve production time.

Corners. Following Cao and Zhai's method, we computed the "net contribution time" of corners (Cao & Zhai, 2007). Thus, for our experimental data and using the average time to perform a line 0.6 m long (see Table 4.2):

T(corner) = sample production time – 2 * 0.627 (seconds).

The measured differences in production time were statistically significant for corner angle ($F_{2,22}$ = 6.49, p < 0.05), but not for direction ($F_{1,11}$ = 2.05, p > 0.05). We found no significant angle × direction interaction effects ($F_{2,22}$ = 0.24, ns). Taking into account that the average *T(corner)* seems to fluctuate around zero (Figure 4.4), we made a deliberate simplification (formula 4.15): to omit corners in the model (Cao and Zhai made the same decision (Cao & Zhai, 2007)). Although these results confirm previous preliminary findings, which postulate that corners have influence on production time of hand gestures (Erazo & Pino, 2014; Erazo & Pino, 2013), we think further research is necessary to adequately model the impact of corners in hand gestures.

$$T = \sum T(line) + \sum T(curve) \tag{4.15}$$

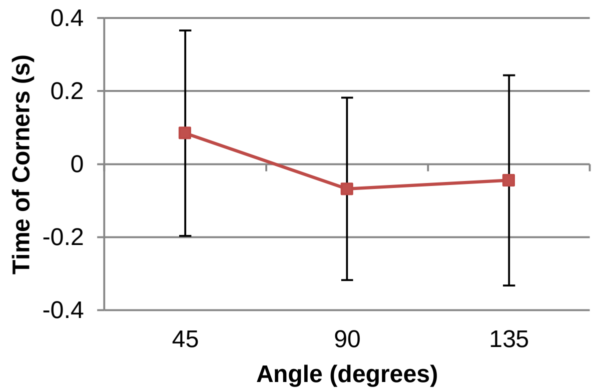


Figure 4.4. Net corner time contribution. Error bars indicate 1 SD.

4.4.2 Isokoski's Model

Experiment

As mentioned above, data obtained from experiment E1 was reused to build an estimation model based on Isokoski's proposal (Isokoski, 2001). Although Cao and Zhai state that a constant time model should be invalidated (Cao & Zhai, 2007), we nevertheless decided to build this model because of its conceptual simplicity. An average time was calculated for each straight line produced by the participants in the experiment (Table 4.2). Moreover, we estimated a fourth value to evaluate the model with a smaller straight-line segment (0.2 m). These times must be verified for selecting the best one by using different gestures (see next section).

Results

Table 4.2. Constant times for Isokoski's model.

Straight line lengths, *L*(m)	Observed time, t(s)	SD (s)
0.2	0.442	NA
0.4	0.544	0.111
0.6	0.627	0.180
0.8	0.738	0.219

4.4.3 KLM

Experiment

Taking into account that experiment E1 is focused on curves, straight lines and corners, we had to perform another experiment (E2) to estimate the empirical constant for the ***D*** operator of KLM, given that it is based on the number of segments and the total length of all segments. The experiment consisted of drawing 14 gestures (Figure 4.5) in random order. Gestures were performed inside the gesture space, which was a 0.6 m square. Twelve persons took part in E2 (mean age 23y, $\sigma = 2$).

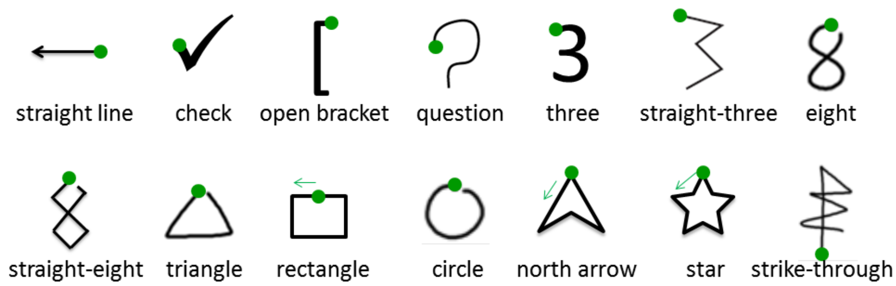


Figure 4.5. Gestures used in experiment E2. The green point indicates the start of the gesture. Strike-through was selected from (Vatavu, et al., 2011).

Results

The procedure adopted by Card et al. (Card, et al., 1980) for drawing straight-line segments (formula 4.5) was tested with hand gestures. This means that gestures with curves ("question", "three", "eight" and "circle", in Figure 4.5) were not used to build the model. The number of segments (n_D) of each gesture produced by participants in the experiment was counted and the total length (l_D) of each gesture was computed (geometrically). Formula 4.16 was obtained by performing regression analysis. The resulting R^2 value was high (0.988), but we thought the model could still be improved. We obtained a higher R^2 value (0.99, see formula 4.17) considering the number of corners and using formula 4.9. (Corners were counted depending on the gesture start point.)

$$D(n_D, l_D) = 0.386\, n_D + 0.349\, l_D \qquad (R^2 = 0.988) \tag{4.16}$$

$$D_c(n_D, l_D, n_C) = 0.223 n_D + 0.297 l_D + 0.173 n_C \quad (R^2 = 0.99) \tag{4.17}$$

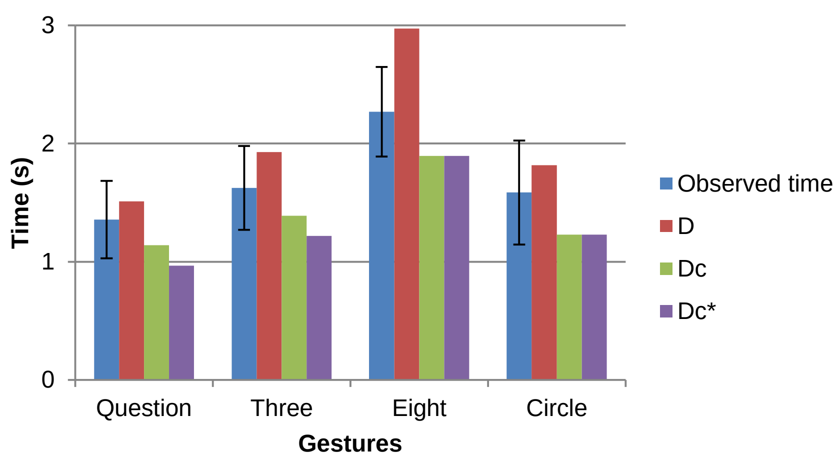


Figure 4.6. Comparison of observed and predicted times of 4 gestures with curves using the D, D_c and D_c* conditions. Error bars indicate 1 SD.

Formula 4.16 was then tested to estimate the production times of the four gestures with curves shown in Figure 4.5, which had not been previously used to estimate the empirical constants (condition D in Figure 4.6). Additionally, formula 4.17 was tested with and without corners (conditions D_c and D_c*). The obtained results for these three conditions are shown in Figure 4.6. These results indicate that these models can also be applied to gestures with curves.

4.5 Evaluation of Models

The production times of real hand gestures must be compared against predicted values in order to evaluate the adapted models. We tried to reduce the number of experiments to a minimum and thus, we decided to reuse data from experiment E2 to evaluate the adapted CLC and Isokoski's models, whose parameters were developed from E1 using a different cohort and different gestures. Regarding the evaluation of the adapted KLM model, a new experiment had to be setup (E3), since E2 was used to estimate the parameters for this model.

4.5.1 CLC Model

Formula 4.15 was suggested to estimate production time using the CLC model, with the provision that formulas (4.10, 4.11) and (4.12, 4.13, 4.14) can be considered options for measuring straight-lines and curves, respectively. Six possibilities can be analyzed to identify the best estimation approach by combining these formulas.

The results obtained from E2 are shown in Table 4.3 for the six formula combinations. We note that some R^2 values are lower than the baseline (Cao and Zhai's results (Cao & Zhai, 2007)), but the obtained %RMSE values are better. Furthermore, the differences between estimates are relatively small. The best results are obtained using the linear model for straight lines (formula 4.10) and the modified model for curves (formula 4.13). Figure 4.7 displays the predicted versus observed data using this formula combination.

Table 4.3. Comparison of CLC model predictions.

Formulas	R^2	**%RMSE**
(4.15), (4.10) and (4.12)	0.834	18.8
(4.15), (4.11) and (4.12)	0.779	18.8
(4.15), (4.10) and (4.13)	**0.859**	**15.7**
(4.15), (4.11) and (4.13)	0.810	15.7
(4.15), (4.10) and (4.14)	0.849	16.9
(4.15), (4.11) and (4.14)	0.798	16.9

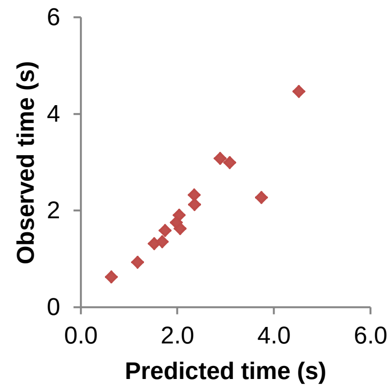


Figure 4.7. CLC model prediction.

4.5.2 Isokoski's Model

E2 also allowed validating the Isokoski's model expressed in formula 4.8 with the empirical constants defined in Section 4.4.2. The obtained results, shown in Figure 4.8, suggest that selecting a constant straight-line length of 0.4 m gives the least estimation error. Figure 4.9 shows the relationship between predicted and measured production times for the suggested straight-line length (R^2 = 0.935, L = 0.4 m, t = 0.544 s).

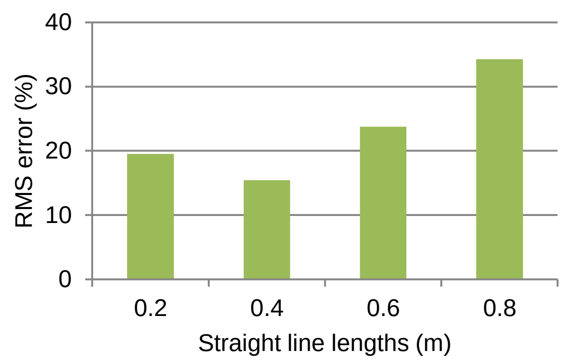


Figure 4.8. Comparison of Isokoski's model prediction errors.

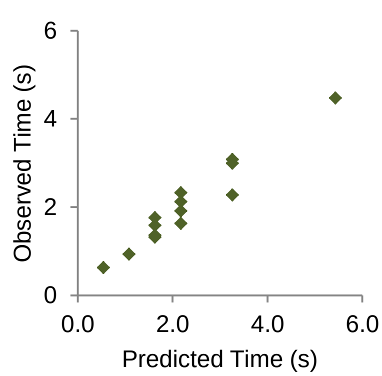


Figure 4.9. Isokoski's model prediction.

Finally, we compared the measured production time with the best results estimated by the CLC and Isokoski's models (Figure 4.10). Isokoski's model is slightly better than CLC, but the difference is quite small to choose the best one. Also, we note the worst predictions were made for gestures "three" and "eight", which are outside ±1 SD.

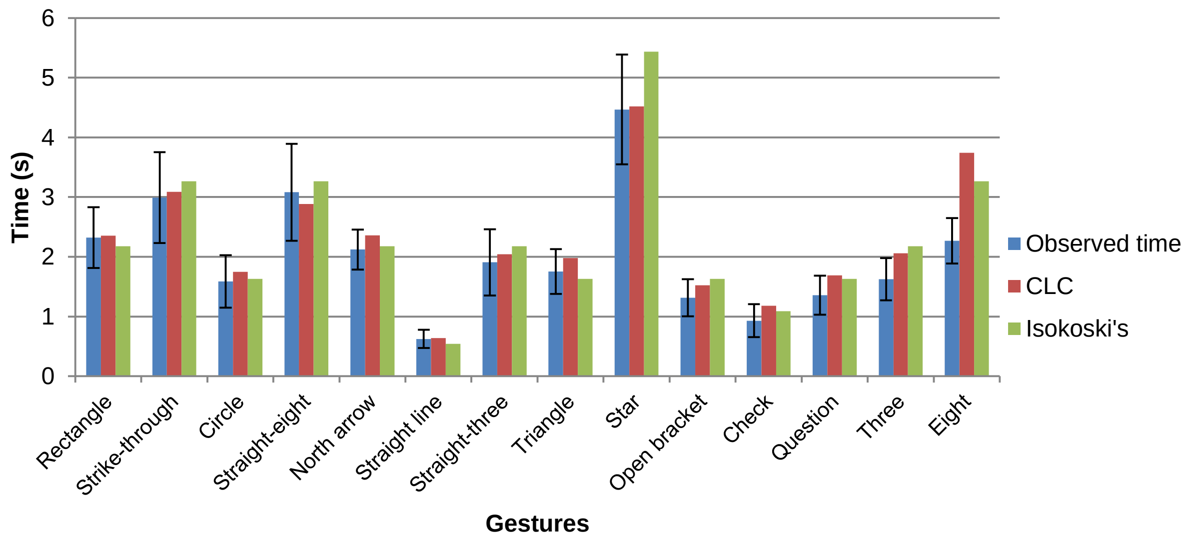


Figure 4.10. Comparison of observed and predicted times using both CLC and Isokoski's models. Error bars indicate 1 SD.

4.5.3 KLM

Experiment E3 was set up in a similar way to E2. Nine participants (mean age 21y, σ = 3), performed the 6 gestures shown in Figure 4.11.

Figure 4.11. Gestures used in E3. The green point indicates the start of the gesture. Steep-hill was selected from (Vatavu, et al., 2011).

Stroke times were compared against predicted values using formulas 4.16 and 4.17. Before applying the formulas, gestures with curves ("5", "E" and "steep-hill") were reduced into straight lines. Additionally, formula 4.17 was calculated with and without corners (D_c and D_c* conditions). For instance, gesture "E" was evaluated using 1 and 0 corners. The obtained results are shown in Figure 4.12.

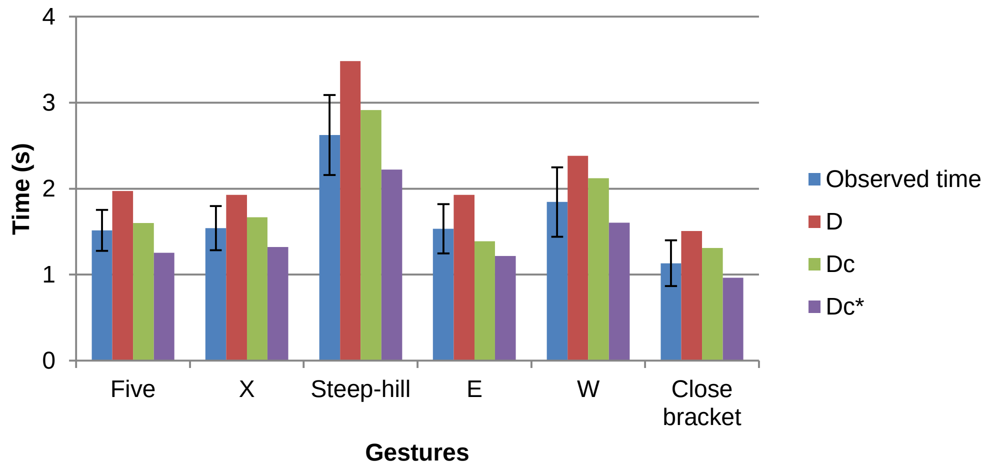


Figure 4.12. Comparison of observed and predicted times for experiment E3 using D, D_c and D_c* conditions. Error bars indicate 1 SD.

The highest R^2 (0.995) value was observed for the D condition, while the lowest (0.947) was observed for D_c, even though they were quite approximate. Since the %RMSE was favorable to the D_c condition (10.4), we suggest that D_c could be considered the best one overall.

4.6 General Comparison

In this section we finally compare the three models, using again the data collected in experiment E3 and focused on the formulas and empirical constants that produced the best estimates.

Table 4.4 shows the selected formulas and quality of estimates using the two quality criteria adopted by this study. Figure 4.13 provides a more detailed comparison using the observed and predicted production times for the six gestures used to evaluate the estimation models. The highest R^2 value was obtained for the CLC model, but the differences to KLM (D_c condition) are quite small. On the other hand, the lowest %RMSE was obtained with KLM (D_c). Consequently, we suggest using KLM (D_c) to predict the production time of hand gestures.

Table 4.4. Comparison of the three models.

Model name	Formulas	R^2	%RMSE
CLC	(4.15), (4.10) and (4.13)	0.996	25.7
Isokoski's	#segments * 0.544	0.881	25.7
KLM (D_c)	(4.17)	0.947	10.4

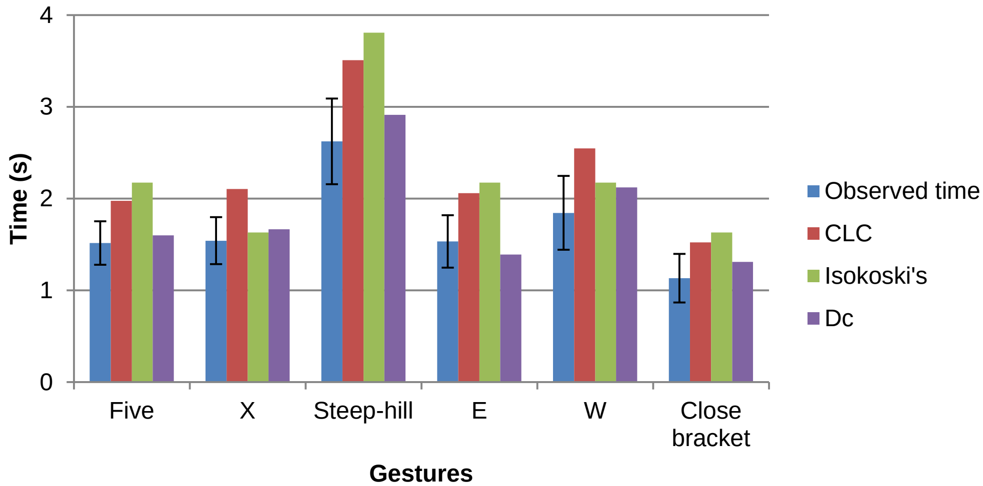


Figure 4.13. General comparison of observed and predicted times using CLC, Isokoski's and KLM models. Error bars indicate 1 SD.

4.7 Discussion

In this chapter we analyze three estimation models for predicting the production time of users' interactions with other types of user interfaces. We extend these models to hand gestures. Empirical experiments were accomplished to tune and validate the models. The quality of the estimates was evaluated using two criteria: strength of the relationship between estimated and observed times, and percentage root mean square error. In a broad perspective, we can conclude that the three models can be used with touchless hand drawing gestures, which confirms hypothesis *H1*. Furthermore, we provide new or updated formulas and empirical constants required to use the models with THDG.

The constant-time estimation model, which was proposed by Isokoski for unistroke writing (Isokoski, 2001), is the simplest model. This model is very easy to use because it reduces gestures to straight-line segments, counts them, and uses a constant multiplier that reflects the average time necessary to produce a straight-line segment. The constant multiplier depends on the constant length assigned to a segment.

According to our results, if gestures are drawn inside a square gesture space with 0.6 m sides, acceptable results can be achieved with a segment that is 0.4 m long. Conversely, the simplifications required by this model lead to erroneous estimations when using variable gesture spaces (i.e., making gestures of different sizes).

An alternative approach, which we also analyzed, consists in using the CLC model, which breaks down gestures into curves, lines and corners (Cao & Zhai, 2007). This model avoids reducing curves and corners to straight-line segments.

We provide new or updated formulas and empirical constants required to use the CLC model with hand gestures. Moreover, our experiments indicate that corners influence production time and therefore should not be neglected. Additionally, slightly different formulas were evaluated, leading us to suggest a new formula for estimating the production time of hand gestures using CLC.

The KLM model also reveals easy to adapt to hand gestures, because it is only based on the number of straight-line segments and the total length of a gesture. Conversely, this strategy also reveals a limitation because the KLM ***D*** operator does not take into account other components like corners and curves. Trying to overcome these limitations, we included corners as a third parameter in KLM estimation formula. The experimental results indicate these modifications provide good results.

Gestures with curves were analyzed as if they were straight lines and with the options of counting or not the number of corners. The obtained results show that counting corners improves the quality of the estimation. Consequently, the adapted KLM formula we suggest counts the number of segments, the total length and the number of corners of a gesture.

Regarding the experiments, we should note the following. First, we could observe a relatively high variation of gesture production times among participants. Although we did not compute a global or final value, the coefficient of variation is, on the average, about 30%. Second, the models were adapted and evaluated only using the gestures' stroke phase, even though they could also be analyzed taking into account a more comprehensive view (e.g., (McNeill, 1992; Kendon, 2004)). Third, the gestures used in our experiments were performed using only the dominant hand, although users may perform gestures with the other hand (Annett & Bischof, 2013). This constraint may have an effect on the estimates (e.g., (Zeng, et al., 2012)).

The model we suggest as the best to estimate production time of hand gestures obtained $R^2 \geq 0.947$ and %RMSE = 10.4, which are better than the ones obtained by Cao and Zhai (Cao & Zhai, 2007) for single pen-stroke gestures. Regarding hypothesis *H2*, we observe it is validated for the CLC and KLM (D_c condition).

4.8 Conclusions

This chapter has studied the possibility of reusing models formulated for other interactions styles to compute production time of THDG. The results confirm that the three candidate models allow predicting the stroke time of THDG in an acceptable way ($R^2 \geq 0.78$, %RMSE ≤ 26). In addition, several modifications of each model were made. These derived models could be also used according to the computed values of the two metrics. Then, the best version of each model was employed to compare the three models, as summarized in Table 4.5. As a result, a variant of the ***D*** operator of KLM would seem to be the best option. However, the analyzed models only allow forecasting the required time to execute the stroke of THDG but not the time to do a whole task on NUIs. It implies these models cannot be used with gestures like hover, tap or swipe. Consequently, the model that gave the best results in this study will be included as part of a general model introduced in next chapter.

Table 4.5. Best version of each analyzed model ordered from best to worst.

Model name	**Formulas**	**Constants**
KLM (D_c)	$D_c(n_D, l_D, n_C) = a\, n_D + b\, l_D + c\, n_C$	a = 0.223 b = 0.297 c = 0.173
CLC	$T = \sum T(line) + \sum T(curve)$ $T(line) = aL + b$ $T(curve) = \frac{\alpha^a}{K} r^{1-\beta}$	For *line*: a = 0.486 b = 0.345 For *curve*: a = 0.615 K = 1.249 β = 0.711
Isokoski's	$T = \#\text{segments} * \text{constant\_time}$	constant_time = 0.544

Chapter 5

Predicting Task Execution Times

Previous chapters have provided useful insights toward developing a model to estimate performance time on interfaces based on hand gestures. Chapter 2 discussed the possibility of using existing quantitative models whose validity has been previously verified, particularly KLM (Card, et al., 1980) and Fitts' Law (Fitts, 1954). Chapter 3 showed that gesture units can be used to analyze THG. Chapter 4 extended several models to encompass THDG. However, the available models at this point cannot be used to forecast the total time to do a task employing THG. Though Fitts' law has been successfully applied to touchless hand interaction evaluation, it is insufficient to evaluate gestural interfaces when they involve other gestures than just "pointing and selecting". Likewise, the models studied in the previous chapter enable the evaluation of interactions using drawing gestures, but they only allow predicting the time to produce strokes. Furthermore, all these models do not cover other aspects such as the mental preparation that users could need before executing a gesture and the structure of gestures (see GCP model in Chapter 3). Therefore, it is desirable to explore the possibility of developing a new model managing these limitations.

A simple hypothetical example may help clarify the need of this new model. We may assume that a UI designer (or a researcher) needs to develop an interface that would allow navigating through various options (e.g., pictures or slides) to find the desired one, and then, choosing it. The designer may use two buttons to enable the navigation (i.e., to go forward and backward), another one to confirm the chosen option, and hold (or hover) gestures to select the buttons. S/he may analyze the design considering, e.g., location and size of the buttons by using Fitts' law as in a classical graphical user interface based on mouse. Then, the corresponding formula would subsume the needed sub-actions, pointing and performing the stroke, to compute the time to select a button. This formula would also include the mental operations needed to make the selections, but some initial preparations and/or movements to resting positions would not be taken into account in that formula. Furthermore, while the clicking action is always the same, the utilized gesture-stroke can change. Actually, several strokes can be used to confirm selections (Walter, et al., 2014). Hence, what would the designer do if s/he wanted to use a different "holding time" (the time the cursor is over the button) or another stroke? S/he may still use Fitts' Law, but s/he might need a different formula for each case. This scenario could be better handled provided the designer had a model covering the various possibilities enabling comprehensive analyses.

Going beyond this "point and select" analysis, the designer may be also interested in incorporating other functionalities to the application. First, s/he may want to allow users to confirm the selection using another gesture and/or replace the confirmation button. A drawing gesture could be employed for this goal, e.g., tracing a check. A model from Chapter 4 can be utilized in the corresponding analysis. Moreover, this model could be complemented with Fitts' Law because using only one of these models is insufficient to perform the analysis of the whole task. However, and as mentioned above, this combination together with the extra mental and physical operations should be addressed in a comprehensive way. What is more, the designer may need to use other gestures, such as swipes, for navigating through the options (Koutsabasis & Domouzis, 2016). Unfortunately, these gestures cannot be modeled with the aforementioned models.

This example describes a single scenario, but many designers might be experiencing similar problems that could be managed with the help of a new predictive a model. Consequently, the purpose of this chapter is to formulate a model to predict performance time in doing tasks on NUIs based on THG (in agreement with objective O2). For this goal, we decided to analyze gestures considering their temporal characteristics and to follow the KLM assumptions in accordance with the previous findings. Also, as a first step, it is assumed that gestures are 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). Thus, we hypothesize that the final model is a practical tool to help designers in the analysis and design of NUIs based on THG. Although the verification of this general hypothesis relies on the validity of three specific ones, this chapter only focuses on building the model while the next chapter describes the validation of those specific hypotheses.

This chapter[6] starts with the formulation of the model, which includes the description of the corresponding formulas and model parameters, or operators. The required bibliographic review to find a set of these operators is also described. Next, the chapter presents the user study conducted to estimate the values for the defined operators. Moreover, further details on the use of the model are provided after building it.

5.1 Touchless Hand Gesture Level Model (THGLM)

THGLM allows estimating the required time to perform a task using a NUI based on THG in a relatively easy way. The model is inspired on KLM (Card, et al., 1980) and based on g-units (Kendon, 2004; McNeill, 1992; Kita, et al., 1998) (see Chapter 2). Like KLM, the method for performing a task must be known and executed without errors. Also, various operators are defined and most of them are assumed to take constant time. However, gestures are more complex than keystrokes, and it is necessary to analyze them in a different way. That is why we decided to use gesture units.

[6] Part of this chapter was used in the writing of publications (Erazo & Pino, 2015) and (Erazo & Pino, 2016).

5.1.1 Model Description

The model asserts that the execution part of a task can be described by means of g-units (Kendon, 2004; McNeill, 1992; Kita, et al., 1998). A task may have one or more g-units depending on its complexity and the system design. A g-unit must be counted every time the user's hand begins to depart from a position of relaxation until the moment when it finally returns to one. For instance, the period of time when a user raises the hand toward a region of space, performs some swipes, and returns to a relaxing position, is equivalent to a gesture unit. Each g-unit time is added up for computing the execution time of a task (formula 5.1).

$$T_{\mathrm{execute}} = \sum_{i=1}^{m} T_{\mathrm{Gunit}_i} \tag{5.1}$$

Likewise, g-units can have different lengths (i.e., one or more g-phrases) depending on tasks and system design. For example, a drag and drop task could consist of one g-unit with three g-phrases: select, move and release an object. Besides, a g-unit may have a retraction phase (optional). Thus, a g-unit time is equal to the sum of all g-phrase times plus an optional retraction time (formula 5.2).

$$T_{\mathrm{Gunit}} = \left(\sum_{j=1}^{n} T_{\mathrm{Gphrase}_j} \right) + [T_r] \tag{5.2}$$

Although we have distinguished two types of g-phrases (see Figure 2.3 in Chapter 2), the method for computing the time for each one is similar (formula 5.3), i.e., to add an optional preparation time with hold time or stroke time as appropriate. The method to calculate hold time is explained later. For stroke phase, the time may be calculated in various ways. However, we propose to compute stroke time in a way similar to KLM (Card, et al., 1980) (formula 5.4). The used operators are defined in the following section. Additionally, we have not taken into account pre-stroke hold and post-stroke hold phases in this model version to avoid increasing the model complexity.

$$T_{\mathrm{Gphrase}} = [T_p] + \{T_{stroke} \mid T_{hold}\} \tag{5.3}$$

$$T_{\mathrm{stroke}} = \sum_{op \in OP} n * op \tag{5.4}$$

Where *OP* is the set of operators that must be defined (see below), and *n* is the number of occurrences of each operator.

5.1.2 Definition of Model Operators

Given the description made in the previous section, we distinguish three groups of operators: general operators, movement operators, and expressive operators. The operators corresponding to each group are described below. Keep in mind that preparation, retraction and hold phases have been included as operators.

General Operators

*(1) Mentally Prepare (**M**)*: Users spend some time mentally preparing to execute subsequent physical operations according to Card et al. (Card, et al., 1980) (and complying with Section 3.2.2). They estimated these mental preparations take 1.35 seconds on the average, but they noted that "the use of a single mental operator is a deliberate simplification" (Card, et al., 1980). Moreover, the ***M*** operator perhaps needs revisiting (MacKenzie, 2013). Despite these cautions, the operator has been successfully used in other works using such value (Holleis, et al., 2007; Luo & John, 2005; Lee, et al., 2015). Moreover, the use of this operator requires following some heuristics provided with KLM (Card, et al., 1980). Including mental operators may be tricky and it takes a lot of judgment; Kieras (Kieras, 2001) provides some suggestions for how to make these decisions. Thus, we decided to use the existing usage guidelines from (Card, et al., 1980) and (Kieras, 2001) making the necessary interpretations (explained below), but we did not adopt the original value for the ***M*** operator.

*(2) Response Time (**R(t)**)*: The second operator represents the time the system needs to respond to user input. This operator is used only when the user must wait before proceeding (Card, et al., 1980), but s/he can overlap other activities while the system is working (Kieras, 2001). ***R(t)*** is highly variable and application dependent. Therefore, the time is provided as a parameter, *t*, in seconds (Card, et al., 1980). Furthermore, we have no evidence for changing the original definition of this operator.

Movement Operators

*(1) Preparation (**Pr**)*: ***Pr*** is the average time a user needs to move his/her hand from a position of rest or relaxation to the position where a stroke begins. Although we propose to use a constant time, preparation time could be estimated in another way (e.g., use a different time depending on the initial location of the hand, with higher values at the top and lower values at the bottom of interaction space.) It is necessary to remember this physical preparation phase is optional and different than mental preparation. Therefore, the ***Pr*** operator should be used each time the user must physically prepare his/her hand to perform a gesture.

*(2) Retraction (**Re**)*: The ***Re*** operator is used each time a g-unit finishes. The value of ***Re*** is the average time to move the hand from the position where a stroke or hold finishes to a resting position. Similar to ***Pr***, ***Re*** may be omitted and estimated in alternative ways.

*(3) Pointing (**P**)*: The ***P*** operator may be defined in a way similar to KLM (Card, et al., 1980) and/or according to Fitts' Law (Fitts, 1954; MacKenzie, 2013). However, pointing tasks in a NUI are different than in other interface types. For instance, designers may use different methods for reaching a target (e.g., hand (Schwaller & Lalanne, 2013; Sambrooks & Wilkinson, 2013) or cross-shaped (Pino, et al., 2013) cursors) and diverse strokes for selecting it (Schwaller & Lalanne, 2013; Walter, et al., 2014). Therefore, it is necessary to consider other aspects (e.g., (Murata & Iwase, 2001; Pino, et al., 2013; Zeng, et al., 2012; Schwaller & Lalanne, 2013)) generating even modified versions of Fitts' Law. In addition, it is necessary to consider that some R^2 values reported are lower than usual values for Fitts' tasks. Beyond this, the ***P*** operator should be used just to estimate the time to point a target. This operator does not include the stroke that follows to select such target (Card, et al., 1980), and another operator is used for this sub-action (i.e., a proper S-phrase operator). Finally, we suggest care when placing the ***P*** and ***Pr*** operators together to avoid redundancy.

Expressive Operators

1) H-phrase Operators: Remembering that an H-phrase has an optional preparation and a hold phases (see Figure 2.3), we defined the ***H*** operator (*Holding*) for the second one (***Pr*** was defined above). The ***H*** operator is used when a user must perform static gestures or hold a hand on a target, position, or pose, a pre-set time. This operator has two parts. The first one is the time users have to hold a hand for considering the action valid. This time is established by designers and called *feedback time* (or *feedback delay time* according to (Müller-Tomfelde, 2007)). The second one is the time a user's hand remains in the same position or pose after feedback time is completed and the hand is moved away. This usually happens when the user receives a confirmation about the recognition of the stroke. It is called *exit time* (Müller-Tomfelde, 2007). Consequently, the total time for this operator is the sum of feedback time and exit time (Müller-Tomfelde, 2007) (formula 5.5).

$$H = feedback\_time + exit\_time \tag{5.5}$$

(2) S-phrase Operators: There are many operators that we could use as part of S-phrase since interface designers may want to use different gestures. Although there are several gestures which are "universally" used, we do not know a standard concerning gestures to be used in NUIs. Therefore, it is necessary to choose gestures that would be considered as operators for the stroke phase somehow (except ***Pr*** which was defined before).

5.2 Finding Expressive Operators

5.2.1 Method

A systematic literature review was performed (Kitchenham, 2004) with the aim of defining the expressive operators, particularly the S-phrase operators. We followed a protocol trying to answer the question:

> *Which are the most used (and/or suitable to use) gestures in NUIs based on THG or touchless interaction?*

The search was carried out in four bibliographic databases: ACM, IEEE, Springer and ScienceDirect, which are some of the most relevant sources in Computer Science (Lisboa, et al., 2010). There were nine search terms used (Table 5.1). Afterwards, a search filter was applied: the publication year of the papers (since 2011). As expected, the database search returned many results; we had to discard many of them because they were not applicable (e.g., use the whole body, multi-touch interaction, etc.)

Three inclusion and exclusion criteria for the papers were used:

1. The study participants are young adults in normal health conditions, with experience using computers, and basic or no experience in THG.

2. The interaction is based on uni-manual gestures. Moreover, if a study also comprises gestures performed with two hands and/or including fingers for creating different hand shapes (except for closing and opening the hand), then it was included but these gestures were not counted.

3. The used gestures are explicitly mentioned[7].

Initially, these criteria were applied by reading the title, keywords and abstract of each search result. This allowed eliminating repetitions and papers that clearly were irrelevant. After that, every pre-selected paper was further analyzed looking for sections in which the authors give details on the aforementioned criteria.

Table 5.1. Search terms. (*) Including repeated papers.

Search terms	No. papers returned	No. papers selected*
'"natural user interfaces" AND NOT (multi-touch)'	753	21
touchless+hand+gestures	222	13
"touchless interaction"	77	6
"freehand gestures"	71	6
"mid-air gestures"	66	3
"mid-air interaction"	56	3
"freehand interaction"	52	3
"In-Air Gestures"	43	4
"Touchless User Interfaces"	13	1

The database search returned a total of 1353 results, but applying the selection criteria, we selected 38 studies[8]. The results were divided in two groups. The first one ("Group 1" in Figure 5.1) contains 22 studies, which describe software based on THG for specific applications. The second group comprises papers (the remaining ones) in which a specific gesture or a set of gestures, for general or specific purposes, are studied.

In addition, backward and forward search (Webster & Watson, 2002) was carried out for complementing the database search. For each paper, the bibliographic section was analyzed for backward search, and the "Cited by" link (provided by Google Scholar) was used during forward search. Once again, the aforementioned inclusion and exclusion criteria, and the search filter, were applied. When no new papers were identified, the search stopped. Five papers were added to Group 1 and nine to Group 2 after performing this search. Following this procedure, some papers were repeatedly found, and recent publications were included. This contributes to increase the confidence on the review.

[7] Hand positions, such as a hand above the elbow or the head, were not considered as gestures.
[8] The selected papers are included in section Additional Bibliography.

5.2.2 Results

As a result, the final list consists of 52 papers (see section Additional Bibliography). Figure 5.1 shows the gestures used in each study as well as the number of papers mentioning them. Although *Pointing* is not included in the figure, it is the most studied operation on touchless interaction. It is mainly based on Fitts' Law (Fitts, 1954). Moreover, some authors used diverse shapes for drawing gestures, but we did not count each different shape. Besides, we consider unnecessary to mention other gestures with just a few occurrences. Finally, the reported gestures may be considered culturally independent given the diversity of participants' nationalities.

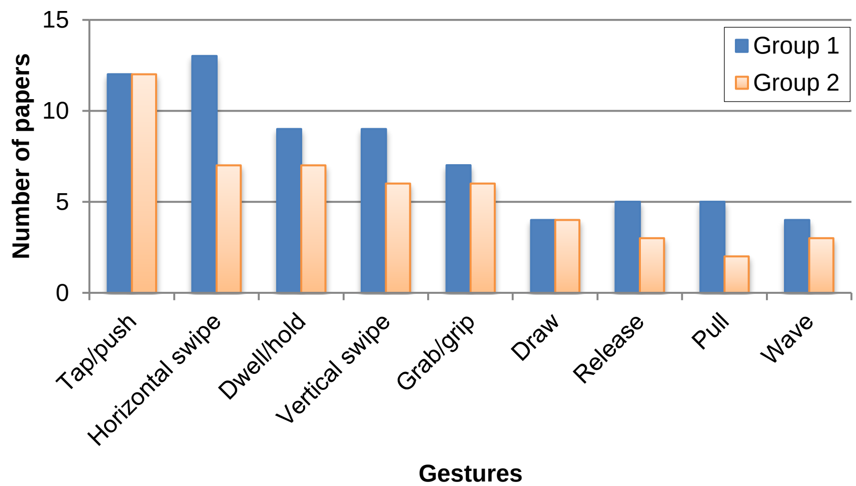


Figure 5.1. Literature review results. Gestures are ordered by the total number of occurrences in both groups. See the text for explanation about Group 1 and 2.

Using these results, the selected operators for the S-phrase are shown in Table 5.2. Although we have not defined operators for all aforementioned gestures, new operators could be added (see Chapter 7).

Table 5.2. Stroke operators.

Operators		**Description**
T	*Tapping*	Push or move the hand toward the front (i.e., make movements in Z axe).
S	*Swiping*	Move 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.
G	*Gripping*	Grasp, grab, grip or close the hand.
R	*Releasing*	Release or open the hand.
D	*Drawing*	Draw shapes, letters, etc. in the air.

5.3 User Study for Time Measurements

The time of each operator must be known in order to be able to use the model and predict required times for tasks. A user study was carried out for estimating these times. For this purpose, the study was divided in six parts with one additional implicit part; the parts are explained in the following subsections.

5.3.1 Method

Participants

36 healthy undergraduate students, 27 male and 9 female (mean age 19 years, $\sigma = 2$) took part in the study; 32 participants were right-handed; 29 of them had some basic experience on gesture interaction, such as using a Wii remote or Microsoft Kinect for playing games. The other seven participants had no prior gesture interaction experience. All participants self-declared their experience on THG and other demographic characteristics in a final questionnaire. Additionally, written informed consent was obtained from all participants.

Apparatus

The hardware setup consisted of a notebook, a Microsoft Kinect sensor and a projected display, mounted in our laboratory. The Microsoft Kinect sensor was connected to a notebook equipped with an Intel Core i7 processor, 8 GB of RAM, to allow tracking users' hand position and recognizing gestures. The Kinect sensor was used with a refresh rate of 30 fps. It was placed at a height of 0.72 cm and below the screen. Participants stood 2.5 meters away from the Kinect while performing the tasks. The notebook was connected to the projected display (with a size of 0.86 × 0.65 meters and a resolution of 1024 × 768 pixels).

A custom software application was developed for directing the experiment. It was developed using Microsoft Visual C# and Kinect for Windows SDK V1.8 on Windows 7. The application logged body tracking data from participants performing gestures and other data (time, users' answers, etc. according to the task performed). Hand movements, joint positions and Dynamic Time Warping algorithm (Sakoe & Chiba, 1978) were used for stroke recognition. Also, the application computed the gesture space for each participant based on McNeill's proposal (McNeill, 1992) and taking into account the postural comfort zone (Kölsch, et al., 2003).

The application interface consisted of an augmented video blending user interface controls and the real environment (Figure 5.2). Augmented video was used trying to avoid participants' distractions while performing the tasks. For example, a person may judge his/her movements based on a hand cursor and trying to adjust (Sutter, et al., 2008), especially because of sensor noise (Livingston, et al., 2012; Zeng, et al., 2012; Sambrooks & Wilkinson, 2013). We verified these assumptions through a set of trial experiments. Additionally, visual and auditory feedback was given taking into account the guidelines from (Microsoft Corporation, 2013; Argelaguet & Andujar, 2013).

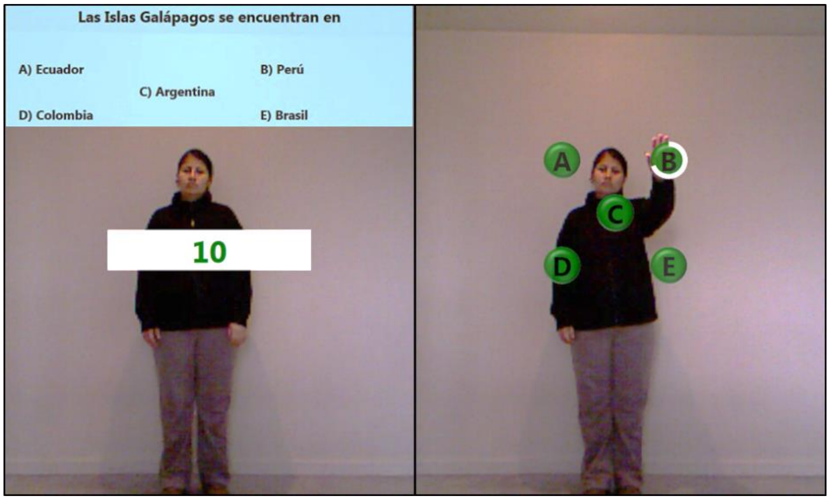


Figure 5.2. Interface of the experimental software.

General Procedure and Design

In general, the participants' task was to answer questions about a subject matter they were studying in the current semester (at the moment the study was carried out). First, they received written instructions on paper. The instructions said they had to answer the questions choosing one out of five possible answers, using their dominant hand, and starting (preparation phase) and finishing (retraction phase) with both hands relaxed and below hips. Questions and other experimental conditions were shown in random order. Each question had less than twenty words (13 to 86 characters without spaces). The time for reading the questions was set to 15 seconds taking into account the average speed for reading on computer displays (Ball & Hourcade, 2011) and a pilot study. After that, the question disappeared, and the answers were shown. The answers were represented with green buttons or images labeled "A", "B", "C", "D" and "E". Participants had to perform different hand gestures to choose an answer, according to each part of the study (one gesture per part). If a gesture was wrong, the answer was discarded and the participant had to re-enter it. Also, participants were instructed to perform gestures balancing speed and accuracy. In addition, participants had to answer four practice questions at the beginning of each part of the study. This was enough for learning the gesture. Each participant answered a total of 73 questions plus 24 practice questions.

A within-subjects design was used where the participants performed six tasks. The task order was counterbalanced using 6 × 6 balanced Latin squares. The whole experiment lasted 1.3 hours per participant on the average.

In addition, observations about interaction mistakes were made. An execution was considered wrong if the participant did not follow the instructions. This allowed discarding wrong executions later (5.6% in the worst case).

5.3.2 Holding (*H*)

This task allowed estimating the time for the ***H*** operator. For this goal, participants had to hold the hand over an answer button (hovering), and a little timer ring appeared around it indicating the remaining time allowed to select that button. Three button sizes (3.5, 5 and 7 cm. in screen size) and five possible answers were used in this task; it gave a total of 15 questions per participant. The feedback time was set to 500 ms. according to Müller-Tomfelde's findings (Müller-Tomfelde, 2007). If the hand was moved out of the button before completing the feedback time, the subtask was canceled and the participant had to repeat it. Additionally, the button appearance and the color changed when the allowed time was completed. The time for this operator was estimated subtracting the feedback time from the period of time since a participant reached the button until his/her hand moved away from it.

5.3.3 Tapping (*T*)

For estimating this operator time, participants had to put the hand over a button and push it. When the participant reached the button, the color and appearance of the button changed, and changed again when the gesture finished. Although participants were instructed to move the hand toward the front about 15 cm, somebody performed a very long movement (this was annotated as a mistake). Again, three button sizes (3.5, 5 and 7 cm. in screen size) and five possible answers were used in this task (15 questions per participant). The stroke time was computed as the period of time since a participant reached the button until s/he stopped pushing it. This value is approximated to those reported in (Polacek, et al., 2012).

5.3.4 Swipping (*S*)

In this section, two parts, horizontal swiping and vertical swiping, are described together because they are similar, although they were performed separately. Participants had to browse the answers moving the hand from right to left or vice versa for horizontal swiping—and up to down or vice versa for vertical swiping— according to an arrow that indicated the direction. They could only browse in one direction at the same time, but they could do it in a cyclical way. For choosing an answer, they had to move the hand down without performing any additional gesture. Ten questions (2 directions × 5 possible answers) were answered in each part. The average time for performing a swipe was computed for every question using only the strokes. In addition, users had to prepare the hand for next swipe, which can be represented using ***Pr*** operators. Alternatively, we decided to include another operator, ***Sp*** (*Swiping preparation*), with other value because the observed time for ***Sp*** was different than the time for ***Pr***. (The ***Sp*** and ***Pr*** operators must not be used together in the same swipe because they are redundant.) As a result, two operators may be used for swipe operations, ***S*** and ***Sp*** (Table 5.3), although designers can use four operators to distinguish between horizontal and vertical swiping (see Chapter 7 for further discussion).

5.3.5 Gripping (*G*) and Releasing (*R*)

This task was similar to the tapping task, but participants had to execute other gestures. They had to perform a grip gesture (***G*** operator) followed by a release gesture (***R*** operator). Unlike the other tasks, participants answered 8 questions: 2 button sizes (3.5 and 5 cm. in screen size) × 4 possible answers. The time for the ***G*** operator was estimated computing the time since the participant reached the target until the grip was recognized. Likewise, the time for the ***R*** operator was computed since the grip was detected until the release was recognized.

5.3.6 Drawing (*D*)

In addition, participants performed a "drawing" in the air task. They answered 15 questions by drawing a letter inside a red box of three sizes (30, 45 and 60 cm. in motor space) using Graffiti gestures (only for letters A, B, C, D and E as shown in Figure 5.3). The observed values were used to test the best evaluated model in Chapter 4 to be included in THGLM as drawing operator, which is a variant of the ***D*** operator of KLM (formula 5.6). The obtained values of the two used metrics confirm this selection given that they remain below the baseline. Therefore, we use formula 5.6 for drawing gestures.

Figure 5.3. Gesture set used for drawing task. The green point indicates the start of the gesture.

$$D_c(n_D, l_D, n_C) = 0.223\, n_D + 0.297\, l_D + 0.173\, n_C \qquad (5.6)$$

Where n_D = number of segments, l_D = total length of all segments (in meters), n_C = number of corners.

5.3.7 Preparation (*Pr*) and Retraction (*Re*)

In each part of the study, participants performed an implicit task for estimating the time of the ***Pr***, ***Re*** and ***M*** (explained below) operators. Two circles were shown inside the gesture space after answering each question. Participants were instructed to "touch" in the air the green circle and then the red one for moving to the next question. The green circle was placed at the center of the gesture space, but the location of the red one was determined using 8 angles (0, 45, 90, 135, 180, 225, 270, 315 degrees). If a participant made a mistake, s/he had to repeat the task. On the one hand, the value of ***Pr*** was estimated as the elapsed time since the participant started to raise the hand until s/he reached the first circle. On the other hand, the ***Re*** time was computed as the elapsed time since the participant moved his/her hand away from the red circle until the return to a rest position below the hip (the hip on the same side of the dominant hand). Although we could get acceptable results, the use of single operators for both preparation and retraction is another deliberate simplification.

5.3.8 Pointing (*P*)

As mentioned above, several research efforts have been performed for analyzing pointing tasks using Fitts' Law and THG (Pino, et al., 2013; Polacek, et al., 2012; Sambrooks & Wilkinson, 2013; Schwaller & Lalanne, 2013; Zeng, et al., 2012; Jude, et al., 2014a). Although we have found no reasons to estimate the time for this operator in a different way that the one in KLM, a value or formula is necessary for using ***P***. However, we have not performed experiments for determining that value or formula. Consequently, we decided to use a value from related works for validating the proposed model. We used the average time observed by Zeng et al. for pointing with the dominant hand and pushing with the other hand (Zeng, et al., 2012).

Furthermore, we subtracted our estimated pushing time from their pointing time. Thus, the used value was 1.046 seconds, which is also consistent with the values reported in (Polacek, et al., 2012).

5.3.9 Mentally Prepare (*M*)

The mental operator time was estimated using the same task used for the ***Pr*** and ***Re*** operators. It was computed as the time the participant needed to start the task (select the green circle), i.e., the period of time since the answers were shown until the preparation started. Although the estimated value is lower than the proposed by Card et al. (Card, et al., 1980), it is in the range of 0.6 – 1.35 seconds suggested by Kieras (Kieras, 2001).

5.4 Overview of THGLM Operators

Table 5.3 summarizes the results of the study showing the THGLM operators with their estimated times.

Table 5.3. Overview of the proposed operators with the corresponding values. [a] This value corresponds to the total time of holding (i.e., 1 second). [b] According to Chapter 4 and as discussed above. [c] The time was computed based on (Zeng, et al., 2012).

<table>
<tr><th colspan="4">Operators</th><th>Time (s)</th><th>SD (s)</th></tr>
<tr><td rowspan="8">***Expressive***</td><td>***H-phrase***</td><td colspan="2">***H***, Holding</td><td>0.500 + feedback_time</td><td>0.103[a]</td></tr>
<tr><td rowspan="7">***S-phrase***</td><td colspan="2">***T***, Tapping</td><td>1.108</td><td>0.370</td></tr>
<tr><td rowspan="3">***S***, Swiping</td><td>Mean</td><td>0.553</td><td>0.211</td></tr>
<tr><td>Horizontal (***Sh***)</td><td>0.613</td><td>0.208</td></tr>
<tr><td>Vertical (***Sv***)</td><td>0.493</td><td>0.198</td></tr>
<tr><td colspan="2">***G***, Gripping</td><td>0.586</td><td>0.152</td></tr>
<tr><td colspan="2">***R***, Releasing</td><td>0.520</td><td>0.172</td></tr>
<tr><td colspan="2">***D***, Drawing, $D_c(n_D, l_D, n_C)$ [b]</td><td>$a\,n_D + b\,l_D + c\,n_C$</td><td>N/A</td></tr>
<tr><td colspan="2" rowspan="6">***Movement***</td><td colspan="2">***Pr***, Preparation</td><td>0.452</td><td>0.103</td></tr>
<tr><td colspan="2">***Re***, Retraction</td><td>0.746</td><td>0.106</td></tr>
<tr><td rowspan="3">***Sp***, Swipe preparation</td><td>Mean</td><td>0.624</td><td>0.325</td></tr>
<tr><td>Horizontal</td><td>0.562</td><td>0.361</td></tr>
<tr><td>Vertical</td><td>0.685</td><td>0.274</td></tr>
<tr><td colspan="2">***P***, Pointing</td><td>1.046[c]</td><td>N/A</td></tr>
<tr><td colspan="2" rowspan="2">***General***</td><td colspan="2">***M***, Mentally prepare</td><td>0.927</td><td>0.116</td></tr>
<tr><td colspan="2">***SR(t)***, Response Time</td><td>*t*</td><td>N/A</td></tr>
</table>

5.5 Using THGLM

5.5.1 Including Mental Operators

Mentally Prepare is an operator that needs special attention as suggested above. Including mental operators is not an easy process as noted by Kieras (Kieras, 2001): 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.

Heuristic Rules for Placing *M* Operators

Figure 5.4 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.

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.

Figure 5.4. Set of updated heuristics for placing M operators (based on (Card, et al., 1980; Kieras, 2001)).

Other Recommendations for Placing *M* Operators

As we mentioned above, Kieras provided some recommendations to use the ***M*** operator (Kieras, 2001). 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 & 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).

5.5.2 A Procedure to apply THGLM

In addition to the model formulation, a procedure to apply it would be useful to achieve good predictions. Figure 5.5 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).

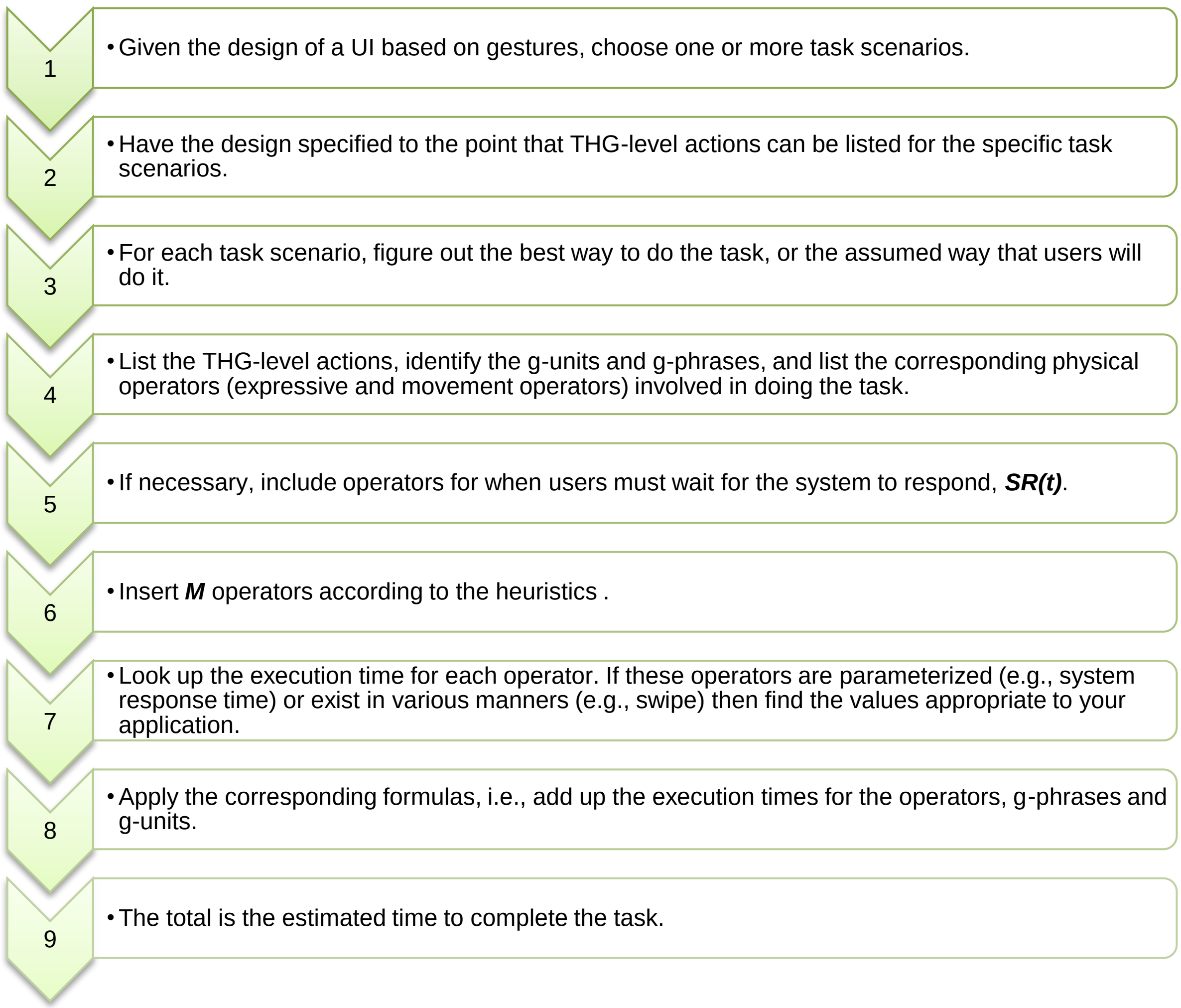
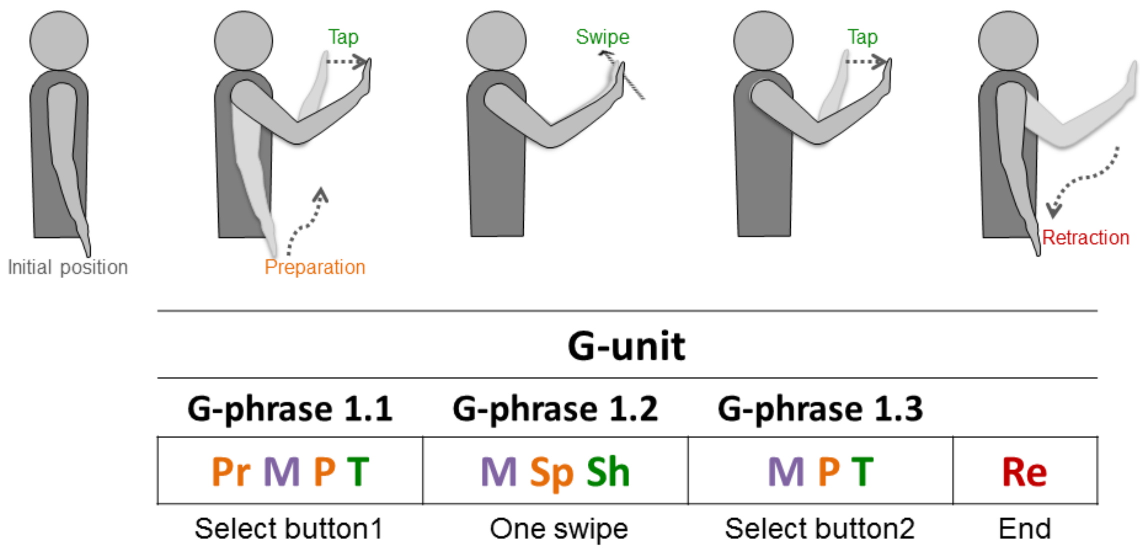


Figure 5.5. Procedure to apply THGLM (based on (Kieras, 2001; Holleis, 2009)).

5.5.3 Example

We are now able to analyze the execution of a task and compare design options using THGLM, the heuristics and the procedure. Let us consider an example to illustrate the use of THGLM (Figure 5.6), particularly according to Figure 5.5. The hypothetical task consists of making a selection (step 1). The user taps a button, performs one swipe, and taps another button (step 2). It is also assumed that the hands are in a resting position.

One gesture unit with three gestures phrases can describe the task (steps 3 and 4). The user will move the hand to the gesture space or interaction area in the first phrase, and then he/she will point or reach the target to perform the tap gesture. Next, the user will perform a horizontal swipe in the second gesture phrase, that is, prepare the hand to perform the swipe and execute it. The last gesture phrase is similar to the first one, but the physical preparation is not needed because the user's hand will be within the gesture space. The user will move his/her hand to the initial or resting position at the end. Figure 5.6 shows these physical operations required to do the task; it also contains the ***M*** operators resulting from applying the heuristics rules shown in Figure 5.4 (step 6). Finally, and after looking up the values for each operator (step 7), the corresponding formulas are applied to compute the execution time of the task (step 8).



$T = (0.452 + 0.927 + 1.046 + 1.108) + (0.927 + 0.624 + 0.613) + (0.927 + 1.046 + 1.108) + 0.746$

$T = 3.533 + 2.164 + 3.081 + 0.746 = 9.524$ seconds

Figure 5.6. Example of using THGLM: Select a button, perform one swipe and select another button.

5.6 Discussion

Similar to KLM, THGLM is relatively easy to use despite the fact some assumptions were needed. The first one is about the S-phrase operators used. Although current devices allow detecting and tracking human body and hand fingers, this interaction type has not been considered in this thesis. More importantly, the following question has not been fully answered: Which S-phrase operators should be considered as part of the model? Keeping in mind that the gestures should help the interface appear "natural" to users (Wigdor & Wixon, 2011), they should allow reusing existing skills and become transparent (Blake, 2012). Since there are many studies on NUIs and touchless interaction, the S-phrase operators selection was made by carrying out a bibliographic review. We think this selection may be culture independent because the participants of the selected papers were people from different countries and continents. Therefore, we consider the selected gestures are a starting point toward a "general" gesture set for NUIs based on THG.

Despite the selected gestures may be used to design and implement several application functionalities, there are other gestures that have not been included in the model yet. The gesture "wave" is an example, which may be used for several tasks (e.g., to identify users (Hayashi, et al., 2014)). This is not a major limitation because the model is extensible. For this aim, the time it takes to perform the new stroke should be estimated—or find the formulas to compute it instead—and included as a new S-phrase operator. This means that stroke time can be computed in different ways, hence the estimate proposed here is only an alternative (formulas 5.3 and 5.4). Something similar happens with the H-phrase operators (formulas 5.3 and 5.5), in which case only one operator has been defined.

We can also consider the case of drawing gestures to illustrate the capability of THGLM to be extended (though this idea is developed in Chapter 7). The previous chapter studied some models for predicting the stroke times of THDG. Although more than one of these models can be used, all the candidates have the same limitation: they only predict production times, and hence, they do not take into account other aspects needed to perform a task. For instance, users should prepare mentally and physically to produce gestures, and they may do it in various ways (e.g., with an open or closed hand; see Chapter 3). THGLM addresses these limitations in some way by incorporating one of those models. Consequently, THGLM enables the inclusion of those other aspects in the analysis because it provides a way to analyze tasks by computing their total times instead of just computing gesture-stroke times.

Other simplifications were made for the movement operators. On the one hand, a constant time, computed from Zeng's et al. work (Zeng, et al., 2012), was used for the ***P*** operator for performing the model evaluation. However, keeping in mind that both the use of Fitts' Law on touchless interaction has been verified by other research (Zeng, et al., 2012; Polacek, et al., 2012; Sambrooks & Wilkinson, 2013; Schwaller & Lalanne, 2013; Jude, et al., 2014a; Pino, et al., 2013) and Mackenzie's suggestion (MacKenzie, 2013), Fitts' Law could be used for this operator. On the other hand, an average time was estimated for ***Pr*** and ***Re*** operators, although they could be computed in other ways and used in formulas 5.2 and 5.3. Nevertheless, as preparation and retraction are optional phases, ***Pr*** and ***Re*** may be omitted and still get acceptable results (take care when this choice is made because worse results could be obtained depending on every case).

As it was mentioned above, a deliberate simplification was made for the ***M*** operator similar to Card et al. (Card, et al., 1980). Although it is expected the estimated value will allow getting acceptable results in the model validations, the mental operator should be revised and maybe "replaced with a set of operators for the diverse interactions requiring user attention and cognition" (MacKenzie, 2013). Also, the heuristics to place ***M*** operators and several additional recommendations have been revised.

Lastly, we consider necessary to mention some details concerning the experiments. First, in general, we could observe a relatively high variation of gesture production times. (The average coefficient of variation among all operators is 30%—and 25% discarding the ***Sp*** operators.) Second, different results may be obtained for users with different characteristics (e.g., impaired people) Third, the gestures used in the experiments were performed using only the dominant hand (the right hand in most cases), but the used hand could have an effect on time (e.g., (Zeng, et al., 2012)), or designers may assume that users can learn with one hand and perform gestures with the other (similar to (Annett & Bischof, 2013)) without problems. Finally, other subjects' characteristics—such as gender or age—might have a significant impact on the results. Consequently, further research is needed to generalize the model.

5.7 Conclusions

We have described in this chapter a model that was named THGLM (Section 5.1), which is the first attempt to formulate a comprehensive model to quantitatively evaluate tasks on NUIs based on THG. This model allows estimating performance time a user needs to do a task using his/her dominant hand. It has been built following the proposals to analyze hand gestures looking at their temporal structure (McNeill, 1992; Kendon, 2004; Kita, et al., 1998) and the KLM methodology (Card, et al., 1980). The first one means NUIs based on THG are analyzed by gesture units. The second one entailed the definition of some operators which correspond to gestures commonly used as part of NUIs. The operator times (Table 5.3) were estimated by user studies in most cases, and using related works when it was needed. However, these values may change depending on users' characteristics. Furthermore, the model has not been validated yet, and other operators, such as Mentally Prepare (***M***), should be further studied. The described heuristics (Figure 5.4) and the procedure to use the model (Figure 5.5) should be also used in the model validation. The following chapters address these issues.

Chapter 6

Validation of the Model

The previous chapter described the development of the THGLM, but an inevitable question arises after building it: Are the predictions made using the model acceptable? In other words, we want to know whether the estimated values of the time to perform tasks on NUIs based on THG are close to the corresponding ones observed when users perform such tasks. An *empirical validation* (Card, et al., 1980) is needed to know it. This validation allows determining the model performance, but other options should be evaluated as well. These options are the use of the model to compare several designs and ask UI designers to utilize the model.

Given this landscape, we hypothesize that (*H3*) THGLM predicts performance time with acceptable quality; (*H4*) the model allows analyzing UI designs and comparing two or more design options; (*H5*) model predictions remain stable when they are computed by independent designers. If these hypotheses are validated, then THGLM should allow designers of NUIs based on THG to carry out usability assessments without users' participation. Therefore, the following sections of this chapter[9] describe the studies conducted to verify the validity of these three hypotheses.

6.1 Empirical Validation

This section explains the four experiments performed trying to verify hypothesis *H3*. Six applications were employed to observe users performing various tasks. Forty seven participants accomplished a total of nineteen tasks, bearing in mind that each participant took part only in one experiment. Then, the required time to accomplish each task was computed and compared against the value previously calculated using the model. It allowed calculating the metrics values to determine whether the predictions are acceptable or not as stated in Chapter 2 (Section 2.3.3). The details on the experiments are explained below.

[9] Parts of this chapter have been used in the writing of publications (Erazo & Pino, 2015) and (Erazo & Pino, 2016).

6.1.1 Method

Participants

The persons who volunteered to participate in the experiments were healthy young adults with basic to no experience in touchless interaction. Undergraduate and graduate students were recruited from our campus. They were invited by several ways such as email and social networks. Users self-declared their experience in THG and other demographic characteristics (age, gender, etc.) in final questionnaires. Their experience was basic and consisted of using Kinect and/or Wii for playing games in most cases. Table 6.1 provides the specific participants' characteristics for each experiment (labeled as E1 to E4).

Table 6.1. Participants' characteristics by experiment.

Experiment	Number	Av. Age (SD)	Right-handed	Female	Experienced with gestures
E1	18	19 (2)	17	8	13
E2	12	18 (0.4)	9	1	10
E3	8	32 (4)	8	1	3
E4	9	19 (1)	8	4	6
Total	47		42	14	32

Apparatuses

Two apparatuses were used in order to collect data of participants interacting with applications. The first one was the same apparatus employed in the "user study for time measurements" described in Chapter 5, which consisted of a notebook, a projected display, and a Kinect sensor. The applications used in this case were NUIPy, OctaNUI, Interaction Gallery[10], and Kinect Paint[11] (Figure 6.1a-d).

We developed NUIPy and OctaNUI using Microsoft Visual C# and Kinect for Windows SDK V1.8 on Windows 7, whereas the other two are third-party applications. NUIPy (Figure 6.1a) allows solving a kind of puzzles that represent basic programs written in Python. Each puzzle part corresponds to one statement shown in a rectangle box. Users have to perform a gesture (tapping or gripping) to select it. If a user selects a statement in the right order, it is executed in Python IDLE[12]. Swipe gestures are used for browsing between programs (puzzles) and tap gestures for selecting the buttons. The second application, OctaNUI (Figure 6.1b), permits interacting with GNU Octave[13] to run basic commands. Users have to select and execute

[10] Interaction Gallery is a WPF C# sample distributed with Kinect for Windows SDK (http://msdn.microsoft.com/en-us/library/jj663801.aspx).
[11] Kinect Paint is available at http://paint.codeplex.com/
[12] IDLE is the Python IDE (https://www.python.org/)
[13] "GNU Octave is a high-level interpreted language, primarily intended for numerical computations. It is quite similar to Matlab." (http://www.gnu.org/software/octave/)

a command with a dataset. OctaNUI employs two types of gestures, tap and horizontal swipe, for selecting buttons and browsing respectively. Interaction Gallery (Figure 6.1c) contains several pages in which a user can view images, reproduce a video, scroll text and browse a map by means of tap, grip and release gestures. Finally, Kinect Paint (Figure 6.1d) allows painting on a canvas and selecting buttons or options using holding gestures.

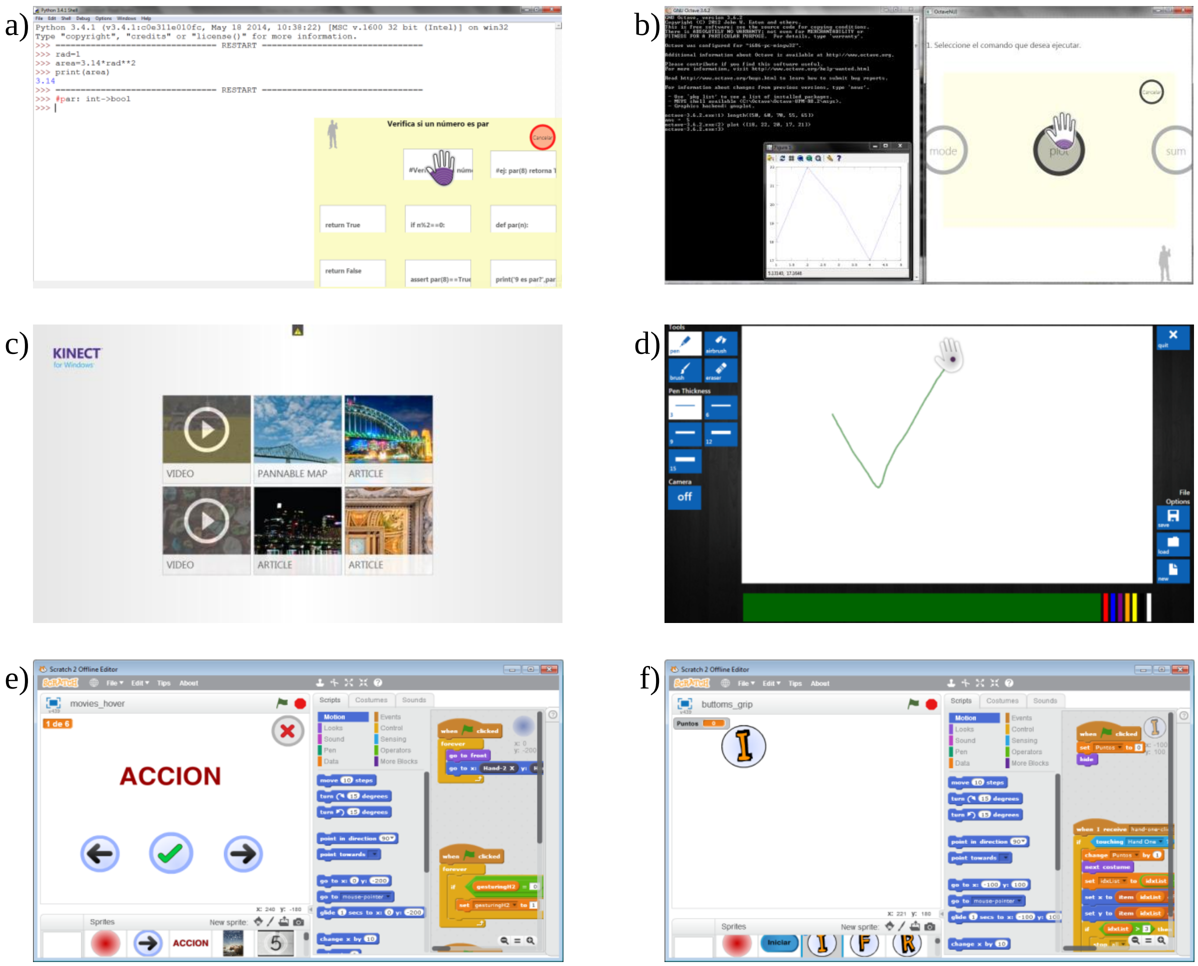


Figure 6.1. Applications used in the experiments: a) NUIPy, b) OctaNUI, c) Interaction Gallery, d) Kinect Paint, e) Bmog, f) Gester.

On the other hand, the second hardware setup 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. 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.

Two applications were developed to be used with the second setup, which were named *Bmog* and *Gester*. Bmog (Figure 6.1e) 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 (Figure 6.1f) in turn 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[14] with the aim of performing the tests with a different platform from previous experiments and encouraging participants (i.e., the undergraduate students) 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.

Table 6.2 shows several features of the applications used in each experiment. The table contains the number of tasks performed with each application, the gestures employed by each application, and the devices used to acquire those gestures.

Table 6.2. Summary of software characteristics by experiment. T = Tap, G = Grip, R = Release, S = Swipe, H = Hover, D = Drawing.

Experiment	Application name	No. performed tasks	Used gestures	Acquisition device
E1	NUIPy	6	T, G, R, S	Kinect
E2	OctaNUI	4	T, S	
E3	Interaction Gallery	4	T, G, R, S	
	Kinect Paint	3	H, D	
E4	Bmog	1	H	LeapMotion
	Gester	1	G, R	

Procedure and Tasks

Overall the experiments consisted of performing first a practice, and then, several experimental tasks. Participants received verbal instructions from the experimenter to carry out the tasks (except for experiment E2). They were instructed for doing the tasks using their dominant hand (the right hand in most cases). Moreover, a within-subjects design and balanced Latin squares for determining the order of tasks were used. Additionally, the experimenter observed users' interactions and took notes on wrong tasks. Further details of the experiments and differences between them are described in the following.

Participants interacted with NUIPy in experiment E1. They performed four practice and six experimental tasks (i.e., they solved ten puzzles): five using tapping gestures and five using gripping gestures for selecting program statements. The used puzzles had 3, 4 and 5 statements. If a statement was selected in wrong order (i.e., the participant made a mistake), s/he had to repeat the whole puzzle.

[14] Scratch is a free educational visual programming language that allows learning some programming concepts easily.

OctaNUI was employed to carry out the second experiment (E2). Participants received written instructions for performing one practice and four experimental tasks. The tasks consisted of selecting and executing a command with a dataset. If participants made a mistake, they could cancel the task to repeat it. The duration time of each task was automatically logged.

Participants used two applications in experiment E3. Interaction Gallery was used to perform four tasks: (1) play and pause a video; (2) display the image of an article; (3) scroll the text of an article, and (4) navigate a map. The tasks started and finished in the main page/window. On the other hand, three tasks were executed with Kinect Paint. Participants had to (1) draw a check after changing several format options (tool, thickness and color); (2) start a new drawing before drawing a question mark, and (3) close the program without saving the drawing.

Experiment E4 consisted of a practice session and two experimental tasks (one with each application). 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.

Data collection was carried out by video recording the whole interactions. These videos were segmented manually by using VirtualDub to compute task duration times. Additionally, observed values that corresponded to task instances with significant errors or in which participants did not follow the prescribed method were discarded in agreement with (Card, et al., 1980). In other words, the task executions with mistakes were not utilized to compute the metric values.

6.1.2 Results

Using the six aforementioned applications, the times to accomplish the tasks were observed and used to analyze and confirm how well the model works; i.e., verify hypothesis *H3*. Task times were estimated using the formulas 5.1 to 5.6 (see Chapter 5), the proposed operator times (Table 5.3), and the heuristics for placing ***M*** operators (Section 5.5.1). These values were compared against the observed ones to compute the metric values.

Figure 6.2 shows a comparison of observed and predicted times for the nineteen tasks. The reached root-mean-square error (RMSE) is 12.1 %, while the average of the absolute prediction error—given as percentage of the estimated time— is 10.1 % (min: -19.1 %, 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 in next chapter). Likewise, Figure 6.3 plots the correlation between estimated and observed execution times. We measure the strength of the relationship between estimated and observed times to be $R^2 = 0.917$.

The values computed for the utilized metrics allow concluding that THGLM forecasts performance time in an acceptable way. These values are in agreement with the baseline established in Chapter 2. Therefore, we can take the next step which is verifying the model utility to compare interface designs.

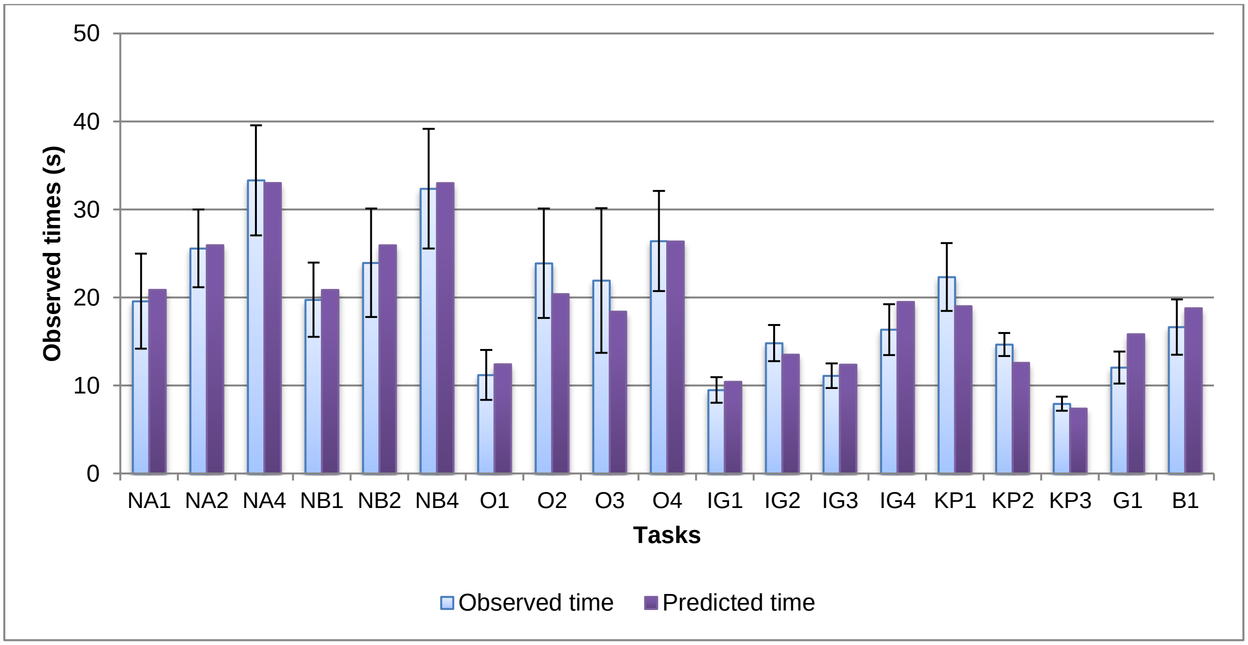


Figure 6.2. Task execution times: observed vs. predicted. Error bars indicate 1 SD. Task notation: N = NUIPy, using tapping (A) and gripping (B) gestures for selecting puzzle parts; O = OctaNUI; IG = Interaction Gallery; KP = Kinect Paint; G = Gester, B = Bmog.

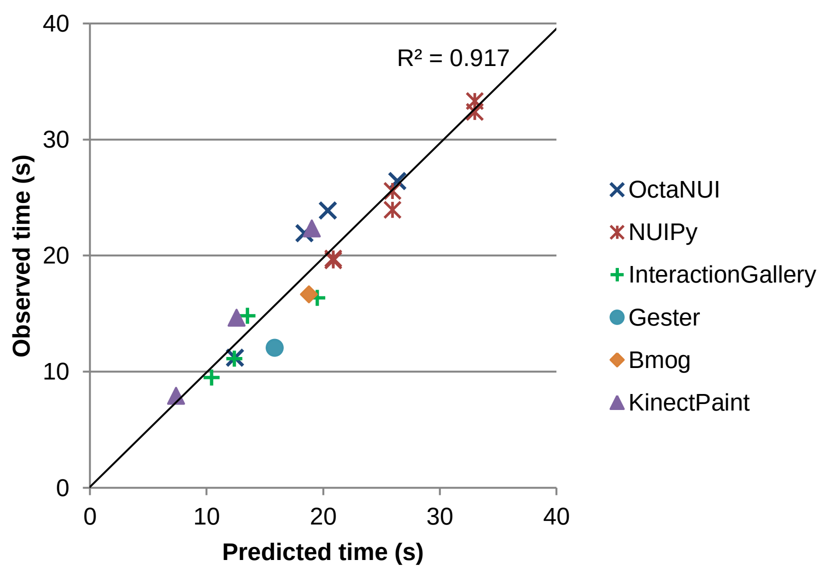


Figure 6.3. THGLM prediction

6.2 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 (i.e., verify hypothesis *H4*). The chosen application should allow a user to take a photo after selecting the desired background/wallpaper picture. The purpose of the study, then, is to mimic the kind of challenges the UI designers will have to face when designing THG user interfaces. Of course, other functionalities could be included as part of this application, but we consider the proposed ones are enough for the goals of this work, i.e., in a sense, this study also models the real scenario.

The study consisted in analyzing three design options for the given application to select the best one (Figure 6.4). 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.

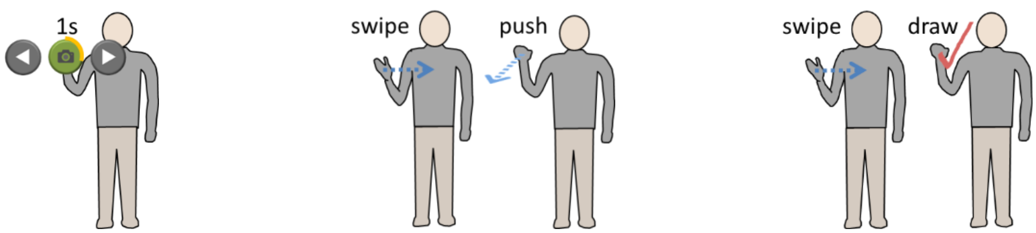


Figure 6.4. Design options used in the study: D1, D2, and D3.

6.2.1 Method

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 employed in experiment E4 (see Section 6.1.1), 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.

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.

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.

6.2.2 Results

The three design options were compared following the same procedure to analyze the model quality described above, which included discarding about 25% of wrong task instances. Figure 6.5 shows this comparison. 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. Therefore, these results support hypothesis *H4*.

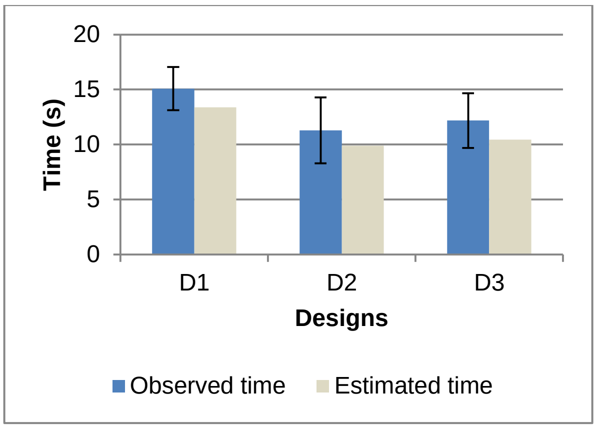


Figure 6.5. Observed and predicted times for the three design options (D). Error bars indicate 1 SD.

6.3 Validation with Designers

We have shown in the previous sections 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 the researcher. 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 researcher should be consistent with the ones predicted by one or more designers to conclude the model is valid or not (Stanton & Young, 2003).

It is arguable that THGLM estimations are consistent across predictions given that it is based on KLM. A large body of research demonstrates the original KLM is a well validated model as mentioned before (see Chapter 2). Moreover, several researchers have verified the validity of KLM when numeric predictions are produced by independent designers including novices (Stanton & 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 not just by the researchers on behalf of hypothetical UI designers, but by real designers themselves.

An empirical study was conducted in order to assess the model predictions with designers' participation. Specifically, this study aims to verify whether THGLM predictions are consistent when computed by independent designers to forecast performance time on UIs based on THG, conforming to hypothesis *H5*.

The study to corroborate this hypothesis 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 the researcher and designers using the model, and observed values, were compared to finally determine the model consistency. This study and results are explained in further detail in this section.

6.3.1 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 (Figure 6.6) and the needed files (explained below).

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.

Procedure and Tasks

After the designers accepted to take part in the study, the researcher emailed them the study instructions and the needed files (five files in total) to accomplish the tasks. The students worked out following the instructions 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 5.1, and Section 5.4), a procedure to apply it (similar to Section 5.5), and several examples of the use of the model. These examples were also provided in a 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).

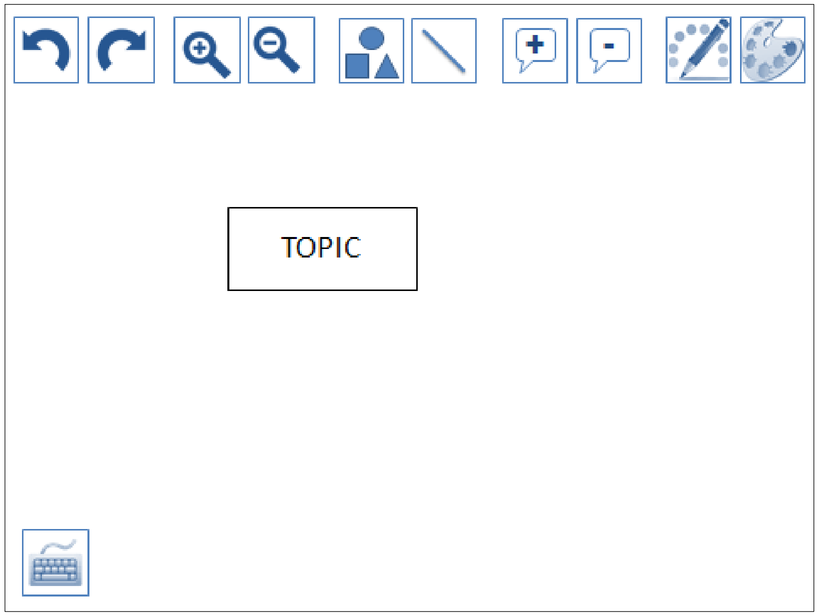


Figure 6.6. User interface utilized in the study.

We provided the initial UI (Figure 6.6) 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.)

As a final task, 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 total disagreement to high or full agreement respectively) regarding model explanation, heuristics for mental operations, procedure to apply the model, examples, and general evaluation (see Appendix D.2).

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.

6.3.2 Part 2: Observed Values

Apparatus

The application prototype used the interface described in the previous section (Figure 6.6) 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 (LM) 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 cm 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.

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, written informed consent was obtained from all participants.

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.

6.3.3 Results

Figure 6.7 shows a comparison between observed times and the times calculated by both the 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 the 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. Figure 6.7 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.

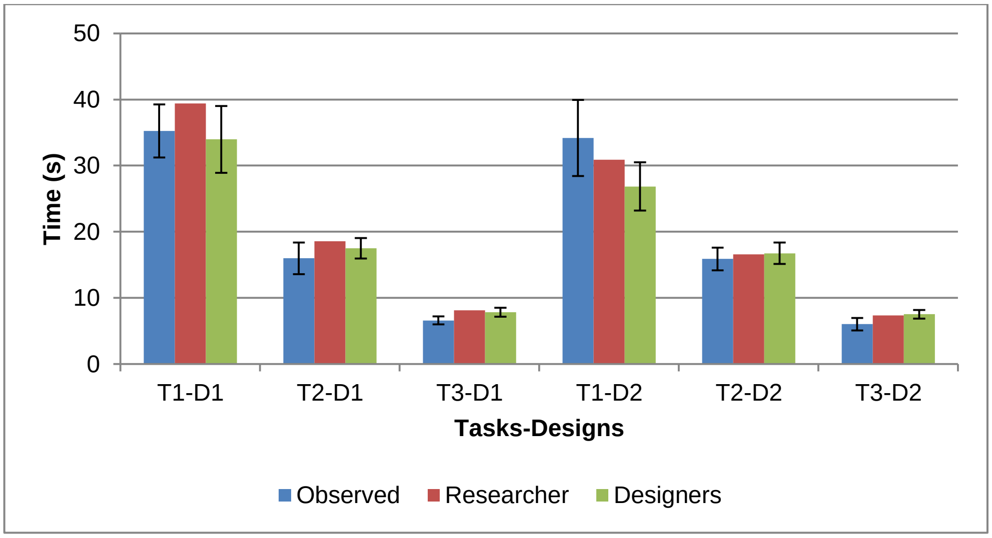


Figure 6.7. Comparison of observed times and times predicted by a researcher and designers. T = Tasks, D = Designs. Error bars indicate 1 SD.

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.

Figure 6.7 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 Figure 6.7). 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. Figure 6.8 shows the obtained scores of the five questions to evaluate the clearness and 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 minutes ($\sigma = 40$, 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).

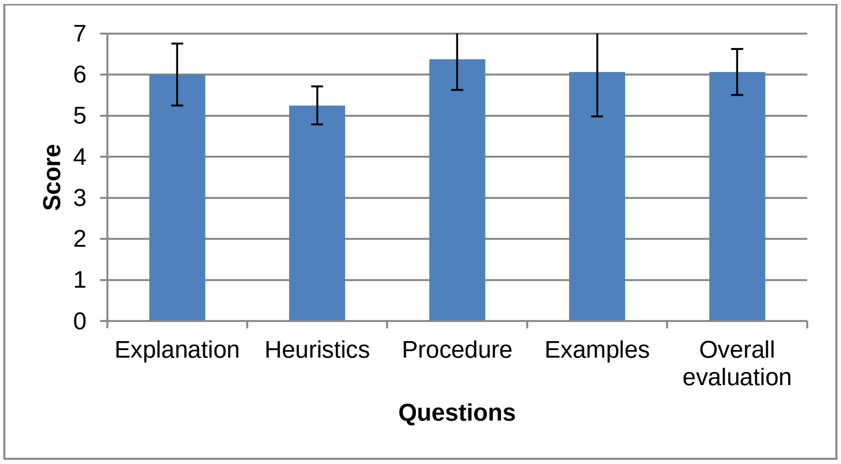


Figure 6.8. Designers' opinions about the procedure to apply THGLM. Error bars indicate 1 SD.

6.3.4 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 *H5*.

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 requires judgment (Kieras, 2001) as discussed before. 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).

6.4 Conclusions

We have described several studies in this chapter in order to verify both whether THGLM makes good predictions and whether it can be used as a design tool. Precisely, the results from the empirical validation confirm the model has an acceptable quality. The reached RMSE is lower than the baseline (21% Card et al.'s error (Card, et al., 1980)), and there was high positive correlation between predicted and observed times. These results validate hypothesis *H3*. Similarly, we took the next step based on these metric values, which is use the model to compare several design options. The results support the hypothesis (*H4*) that the model can be used to compare two or more UI designs and choose the best one. Furthermore the model utility has been confirmed by independent designers (hypothesis *H5*) who were invited to participate in a study. Consequently, the validity of these three hypotheses confirms the general hypothesis that THGLM can be used to assess or analyze NUIs based on THG.

Chapter 7

Extending the Model

THGLM has an acceptable performance according to the empirical evaluation, but it does not include all operators that may be used to analyze interface designs. However, it is not a major limitation because the model is extensible following the KLM inspiration. If a designer or researcher thinks a new operator to the model has to be incorporated, then s/he has to estimate the time value of that operator or find the equation(s) to compute its value. The *Drawing* operator is an example of a new S-phrase operator that was added to THGLM—in Chapter 5—based on a formula verified in Chapter 4. Probably most new operators will belong to the expressive category, but operators for the other groups can also be included as demonstrated below.

Though finding all candidate operators that may be incorporated to the model is beyond the scope of this dissertation, this chapter[15] introduces several operators that may be useful in the analysis of NUIs based on THG. The chapter starts presenting the selected operators, which have been in fact mentioned or suggested before, implicitly or explicitly. The utility of these operators is illustrated after computing their values. Also, other operators that may be included in the future are suggested.

7.1 Candidate New 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), but with the limitation of being a "deliberate simplification" (Card, et al., 1980). Due to this and other reasons, MacKenzie (MacKenzie, 2013) advocates updating the ***M*** operator by replacing it with a set of operators. It must be noted, however, that this operator has been successfully used in other works using the original value (e.g., (Lee, et al., 2015; Holleis, et al., 2007; Luo & John, 2005)).

[15] Most part of this chapter has been used in (Erazo & Pino, 2016).

Based on (Card, et al., 1983), MacKenzie (MacKenzie, 2013)(pp. 272-274) proposes to use five $\boldsymbol{M}$ 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 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 (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 $\boldsymbol{M}$ 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 & 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., 2014a). 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 ***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 & 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 (see Chapter 5), 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, we have only proposed the values for some S-phrase operators that may be used to analyze some application functionalities, but 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 NUIs 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

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 the list of THGLM operators and in consonance with (Walter, et al., 2014)[16]. The same gestures were performed in the second phase (P2), but participants used both hands; that is, they randomly selected the hand

[16] 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., holding requires a constant time to be recognized, and its recognition starts when the cursor is over the button, and hence, reaction time is not observable.

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. (see Appendix D.3).

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 know 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 seconds 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 minutes on the average.

Apparatus

The hardware setup was the same used in the experiment to validate the model with UI designers described in the previous chapter (Section 6.3); i.e., it consisted of a desktop computer, a display and a gesture acquisition device. A new custom application was developed to be employed with this setup. 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., 2014a)) 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.

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 hours per week, and thirteen had some basic experience on gesture 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.

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 & 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 ***Hp*** operator value; and the ***U*** operator value was obtained using the times of pulling gestures. Table 7.1 summarizes the obtained times.

Table 7.1. 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
$\boldsymbol{M_S}$, Simple reaction	General	0.375	0.076
$\boldsymbol{M_P}$, Physical matching	General	0.388	0.065
Hp, Hand preference	Movement	11%*	N/A

The first operator in Table 7.1, *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 toward 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 ***P*** operator and not to the ***T***, ***G***, ***R***, and ***U*** operators. Nevertheless, we did not analyzed MT, and hence, it is necessary to use some related work. Specifically, Jude et al. found the degradation in MT to be about 11% (Jude, et al., 2014a). This value could be 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[17] using the constant value suggested in Table 5.3).

Finally, we made a comparison between the current and the previously proposed times (see Table 5.3) 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 Figure 7.1 (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.

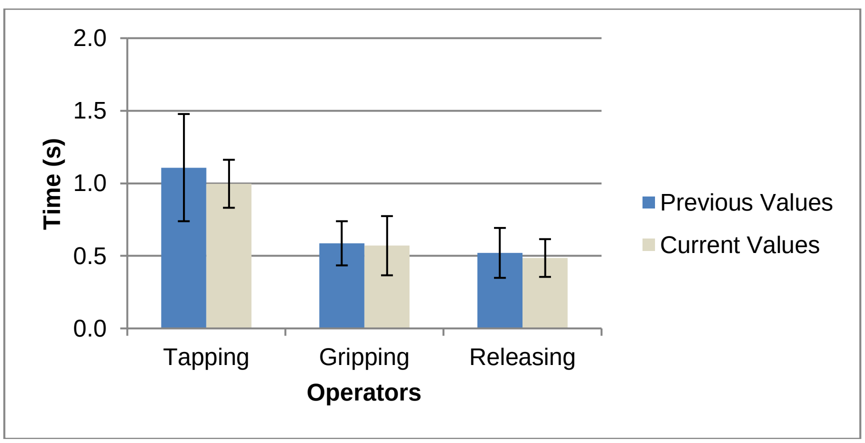


Figure 7.1. Comparison of current and previous times for three operators. Error bars indicate 1 SD.

7.3 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 (described in Chapter 6), 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

[17] This value is lower than the average value reported in (Jude, et al., 2014a), but it falls within the computed intervals.

using the NDH. On the other hand, the same procedure applied in Chapter 6 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 in Section 6.1, 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 ($\boldsymbol{M_S}$ 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 $\boldsymbol{M_S}$ 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*** operator should be updated according to MacKenzie's suggestion (MacKenzie, 2013).

7.4 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 this suggestion in mind, 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 5.3 includes an operator for swiping (as well as the corresponding ones to prepare for performing a swipe) according to Chapter 5. 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, we suggest following the same idea as shown in Table 5.3 (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 before. Although a constant value has been used in the experiments to validate THGLM, Fitts' Law can be used to estimate the time to reach a target as demonstrated in several works (Zeng, et al., 2012; Polacek, et al., 2012; Sambrooks & Wilkinson, 2013; Schwaller & Lalanne, 2013; Jude, et al., 2014a; Pino, et al., 2013). Steering Law (Accot & Zhai, 1997) is another model that may be evaluated using THG and possibly be extended. The potential HCI applications of this model are device comparison and menu design as explained in Chapter 2. As a consequence, if the use of this model is verified and/or adapted to be included as an operator in THGLM, then it could be used to forecast performance to navigate vertical or horizontal menus, drag an element through a "tunnel", etc.

Regarding the $\boldsymbol{M}$ 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 thesis. However, it is worth mentioning 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 (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 he 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.

7.5 Conclusions

This chapter has demonstrated our model is extensible by adding and using several new operators. On the one hand, an improvement on the mentally prepare operator has been proposed. It consists of two mental operators for simple decisions tasks. The performed test indicates that one of these operators worked well because the predictions were improved. However, the analysis involved only one task which is insufficient to generalize the results. In any case, these results provide evidence in favor of studying in further detail the $\boldsymbol{M}$ operator. On the other hand, two new operators have been added to represent pull gestures and hand preference. The first one is an expressive operator to represent other kind of strokes. The other operator, which belongs to movement operators, enables modeling interactions performed with either the dominant or the non-dominant hand. Nonetheless, it should be improved to predict time of bimanual tasks. All in all, these new operators expand the possibilities offered by THGLM, and other operators can be included in the future as needed[18].

[18] Appendix B contains the final list of the model operators after including the ones described in this chapter.

Chapter 8

Conclusions

This thesis has addressed the use of model-based evaluation with interfaces based on gestures. Given that current models were insufficient to assess or analyze NUIs based on THG, we proposed to develop a predictive model that allows estimating required time to execute a simple task using an interface of this type in an acceptable way. This objective has been achieved, as well as the proposed four specific objectives.

This chapter concludes this dissertation providing some final considerations. Overall the chapter presents the conclusions and discusses briefly several advantages and limitations, as well as some suggestions of applications of the model. The chapter contains four parts that are in connection to the objectives stated in Chapter 1. Next, the chapter summarizes the contributions of this thesis. Finally, some possible future directions to address the model limitations are provided.

8.1 Extending Existing Models

We analyzed the feasibility of extending/adapting existing models to encompass THDG in agreement with specific objective O1. Three models (CLC (Cao & Zhai, 2007), Isokoski's (Isokoski, 2001), and the drawing operator of KLM (Card, et al., 1980)), which were developed for other interface types, were analyzed following the original proposals and doing some changes. Three experiments were carried out to build, evaluate and compare the models using the selected two metrics. The results of these experiments confirmed that the three models can be used because acceptable values for both metrics were obtained ($R^2 \geq 0.88$, RMSE $\leq$ 26 %). Consequently, these models enable UI designers to quantitatively analyze THDG by forecasting production time of the selected gestures.

The results also showed that the best model is a variant of one of the KLM operators (i.e., the drawing operator). This model (D_c) asserts production time is a linear function of the number of segments (n_D), the total length of all segments (l_D), and the numbers of corners (n_C) of the gesture. Additionally, it assumes gestures are performed within a square of side about 60 cm which roughly corresponds to the gesture space. This model should be preferred to forecast production time of THDG according to the computed metric values ($R^2 \geq 0.90$, RMSE ≤ 21 %).

It is worth noticing that the D_c model only predicts production times, and hence, it does not include other aspects needed to perform a task. For example, users prepare their bodies and minds before drawing gestures, and they may do it in various ways (e.g., closing the hand, using a finger, using the other hand as a signal to start, etc.). In conclusion, the D_c model is useful just to compute gesture-stroke times and not the total time to do a task. The other studied models have the same limitation. This fact led us to search for a way to include those other aspects in the analysis of tasks.

8.2 Formulating a New Model

Reached this point, we were able to analyze pointing tasks and drawing gestures using Fitts' Law (Fitts, 1954) (not studied here) and the D_c model respectively. However, these models have individual limitations as discussed before. We addressed those limitations by subsuming both models and providing a way to analyze tasks in a comprehensive manner. In other words, we assumed that it was possible to develop a comprehensive predictive model that allows estimating the time a user takes to accomplish a simple task on a NUI based on THG. Verifying this supposition required dividing the work in two steps: formulation and evaluation of the model.

The proposed model, that was named THGLM, was developed based on existing basic theory, that is to say, using gesture-units (Kendon, 2004; McNeill, 1992; Kita, et al., 1998) and the KLM methodology (Card, et al., 1980). THGLM 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. It 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, and g-unit time in turn is computed by summing up all g-phrases plus an optional retraction time. The number of g-units and g-phrases also depends on the complexity of tasks and interface designs. Besides, it was necessary to make some assumptions so that the model is restricted to gestures performed by young adults in normal health conditions, with basic to no experience with touchless interactions, and using the whole hand (fingers are not considered).

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.

Referring to the model parameters, THGLM uses a set of operators that are different than the corresponding ones used by KLM because gestures are more complex than keystrokes. 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 we decided to perform a systematic bibliographic review to find the candidate gestures to be included as operators. We chose the gestures most frequently used in related works to be included as operators. One of these operators corresponds to the D_c model that was included afterwards to support drawing gestures. Thus, the current model contains fourteen operators (plus the variants of some of them) with their estimated times (see Appendix B for a whole overview).

Two additional steps were required as part of the model formulation. On the one hand, we updated the heuristics rules and recommendations for including mental operators as THGLM is based on KLM that uses those rules. On the other hand, a procedure to apply the model was generated in accordance with objective O4. We studied the model performance in detail using these enhancements as well.

8.3 Validating the Model

We proposed three specific hypotheses to study THGLM performance as stated in Chapter 6. They refer to the quality of predictions (*H3*), the comparison of design options (*H4*), and the use of the model by designers (*H5*). The verification of hypotheses *H3* and *H4* allowed achieving objective O3. (The next section refers to *H5*.)

Firstly, the empirical validation confirms the quality of the model to forecast performance time. THGLM 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 supported by the high relationship between estimated and observed times ($R^2 > 0.9$). These results validate hypothesis *H3*.

Meanwhile, the results from the study to analyze UI designs validate hypothesis *H4*. 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.

Even though these results confirm the validity of the model, it is important to mention some additional details. Given that a limitation of THGLM lies in being constrained to error-free execution, we proceeded during the model validation 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”. Actually, the model performance will decrease if tasks with significant errors are 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., to introduce an operator for age). Other aspect refers to the variability of the measured times. We reached relatively high coefficients of variation for observed task times. The average coefficient of variation remained near 20%. However, these values are comparable to previously reported ones (e.g., 31% in (Card, et al., 1980)). Beyond these details, THGLM has gotten an acceptable quality making the required assumptions.

It should be also noted that we have used two gesture acquisition devices to carry out the experiments: Kinect and LM. The justification for this decision is twofold. On the one hand, both devices enable recognition of gestures included in THGLM as operators, but depending on the specific case/need, LM has greater precision whereas Kinect allows recognizing full body gestures. On the other hand, and from a broad perspective, our emphasis in on modelling performance of gesture interfaces, and thus, the study of it could make use of any capturing device, as opposed to studying the capabilities of a specific device.

8.4 Applying the Model

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. The participants followed the generated procedure to apply the model and forecast task times. Notably, designers' predictions stayed stable in comparison to the researcher's ones, confirming hypothesis *H5*. 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 an easy way.

We also observe that our general assumption is validated for the three specific hypotheses (*H3* to *H5*). 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 in order to develop good software products using THG.

In addition, there is another aspect that may be needed when applying the model; it is the necessity of including new operators. Although we have not included all possible operators in the model formulation and validation, we have demonstrated the model is extensible by adding new operators. All the proposed operators (Appendix B) may be used to analyze various application functionalities, as well as more operators can be easily included by estimating the corresponding times, or finding the equations to compute them instead.

Summing up, the models discussed here could be useful tools to analyze, assess, select and/or improve designs of interfaces based on gestures. On the one hand, if UI designers need visual aids to understand THG conception and production, then they can use a descriptive model (e.g., the one proposed in Chapter 3). On the other hand, designers can be also interested on performing quantitative analyses based on performance time. In this case, if they need to select gestures to increase the efficiency of UIs, then they can use the D_c model (Chapter 4) to compare and choose gestures before and/or without testing with real users. Also, if their designs include only pointing tasks, then they can evaluate or compare designs using Fitts' Law (widely studied according to related work), and perhaps, also some of the THGLM operators for expanding options. However, designers may prefer analyzing tasks comprehensibly using THGLM (Chapters 5 and 6), which can also be extended by including new operators for additional gestures (Chapter 7). All in all, these models should allow UI designers generate suitable gesture-based designs for several applications.

8.5 Contributions

In conclusion, this thesis makes the following main contributions to the field of Human-Computer Interaction:

- A quantitative model that is the first comprehensive model to forecast performance time on NUIs based on THG. Software designers and/or researches would appreciate to have available a model that could be used in a relatively easy way without undertaking time-consuming and resource-intensive ad-hoc experiments. Thus, the main contribution of this thesis is a valid and usable model for predicting execution time of NUIs based on THG by adult novice users.

- A list of hand gestures commonly used as part of NUIs or touchless interfaces (Appendix C) with estimated strokes times (Appendix B, S-phrase operators). Additionally, we think these gestures may be culture independent. Therefore, we consider the selected gestures are a starting point towards defining a "standard" gesture set for common tasks on NUIs based on THG.

- A model to estimate the required time to perform THDG. Consequently, software designers or researchers interested in analyzing performance of THDG will have a model to estimate production time of the selected gestures. They may also need to make comparisons among gestures, or they may want to design complete gesture sets.

- A qualitative and quantitative analysis of gesture articulation variability. This contribution is derived from the preliminary steps we took with the goal of better understanding production of touchless gestures. Actually, the performed study also made the following contributions: (1) a descriptive model on gestures conception and production (pp. 30-32, Chapter 3); (2) an embodied taxonomy of touchless gestures (pp. 32-33, Chapter 3); (3) implications for designing applications based on touchless gestures.

8.6 Future Work

Finally, THGLM has several constraints in spite of the good results and advantages. 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 support other kind of gestures and/or bimanual interactions.

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 support bimanual interactions. For example, one hand may 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).

On the other hand, there are several specific scenarios 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 be also 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). A second option is the use of interfaces based on THG in classrooms (Erazo, et al., 2016). In this case, for example, the model may provide reference values that could be used to design applications and/or plan the tasks that students would do during a class. 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, and particularly using model-based evaluation (MacKenzie, 2013), there are other dimensions of performance. Fatigue is one of these dimensions that should be considered to design UIs based on THG. The fact that gestures are performed in the air produces fatigue and a feeling of heaviness in the upper limbs (it is called gorilla-arm effect). Moreover, some gestures and regions of gesture spaces may be more prone to fatigue effect (Hincapié-Ramos, et al., 2014). Likewise, the number of errors users make could depend on the performed gesture and/or the interface design. Of course, other aspects related to user performance (e.g., learning, recall, etc.) may play an important role. Unfortunately, there exist little research addressing these aspects, and hence, it is unfeasible to forecast the corresponding values. All in all, 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.

Appendices

Appendix A: Models for Touchless Hand Drawing Gestures (THDG)

A.1 Applicability

Drawing gestures is an approach different than and/or complementary to pointing tasks that can be analyzed using one of the predictive models studied in Chapter 4. These gestures may be drawing figures of shapes, letters, numbers, etc. in the air. As an example of their utility we can select a gesture to be used as a shortcut of an application. We would need to compare two or more gestures, and hence, we could apply a model to compute the production time of each one, and then choose the one with the lowest value. Following a similar procedure, UI designers could also define the set of gestures that the application will recognize; i.e., find the specific gestures for a most efficient interface. Therefore, the proposed models for THDG enable the comparison of gestures to select the best one or create a complete gesture vocabulary.

A.2 Description

We have adapted three models, that were formulated for mouse and pen interactions, to estimate the production times of THDG: Isokoski's (Isokoski, 2001), CLC (Cao & Zhai, 2007), and KLM (***D*** operator) (Card, et al., 1980). Each model was analyzed as it was originally proposed and modified accordingly. Table A.1 contains the best version for each of these modified models.

The model that reached the best performance in the experiments is D_c, which a variant of one of the KLM operators (i.e., the drawing operator). It is a linear model that allows computing production time based on the number of segments (n_D), the total length of all segments (l_D, in meters), and the numbers of corners (n_C) of the gesture. As its first parameter is the number of segments, the curves of the gestures must be represented using straight lines in order to count those segments. The procedure to achieve this representation is as follows: "if the angle α inscribed by an arc was greater than 270°, then use 3 segments; if $\alpha < 120°$, then use 1 segment; otherwise use 2 segments" (Vatavu, et al., 2011). Figure A.1 illustrates this procedure for the graffiti gestures "D" and "E" utilized in Chapter 5 (see Figure 5.3).

On the other hand, CLC asserts that the production time of a gesture is equal to the sum of all lines and curves that compose the gesture as shown in Table A.1. For example, the gesture "D" (Figure A.1) comprises one line segment and one curve. The suggested formula for lines is a function of the length of the line segment (L, in meters), whereas the formula for curves counts the angle (α, in radians) and the radius (r, in meters) of the curve.

Isokoski's model in turn computes the time by multiplying the number of segments (see Figure A.1) by a constant time. Table A.1 provides this value but Table 4.2 (Chapter 4) contains further options.

Table A.1. Best version of each analyzed model ordered from best to worst.

Model name	Formulas	Constants
KLM (D_c)	$D_c(n_D, l_D, n_C) = a\,n_D + b\,l_D + c\,n_C$	a = 0.223 b = 0.297 c = 0.173
CLC	$T = \sum T(line) + \sum T(curve)$ $T(line) = aL + b$ $T(curve) = \frac{\alpha^a}{K} r^{1-\beta}$	For *line*: a = 0.486 b = 0.345 For *curve*: a = 0.615 K = 1.249 β = 0.711
Isokoski's	$T = \#\text{segments} * \text{constant\_time}$	constant_time = 0.544

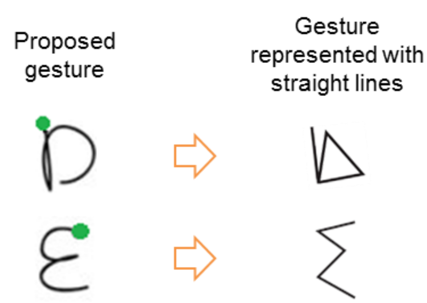


Figure A.1. Gesture "D" and "E" represented with straight lines.

A.3 Limitations

Although the results presented in Chapter 4 confirm that the aforementioned models (Table A.1) have acceptable performance, the three ones have several limitations (the same ones in all cases). The main limitation is that these models only forecast production times of THDG and not the time to accomplish a complete task as it is the case of THGLM (Appendix B). However, the model with the best scores (D_c) is part of THGLM to give more opportunities in usability analyses (discussion in Chapter 5). A second limitation refers to the gesture space in which it is assumed that gestures would be performed. The gesture space used in the studies roughly corresponds to a 0.6 m square (see Figures 4.1 and 2.5). Finally, gestures are assumed to be executed by novice young users in normal health conditions using the dominant hand.

Appendix B: Touchless Hand Gesture Level Model (THGLM)

B.1. Applicability

As described in Chapter 5, THGLM is a predictive model based on KLM (Card, et al., 1980) and gesture units (g-units, Figure B.1) (McNeill, 1992; Kendon, 2004; Kita, et al., 1998). It is an easy to use model that enables the analysis and/or evaluation of the design of NUIs based on touchless hand gestures. For this purpose, THGLM allows estimating the execution time of a task given a method. Then, this value can be used as a metric to compare design options and select the best one, analyze changes on the interface design or identify possible problems. This approach should be followed especially at early design stages or as a previous step before testing the actual interface with real users.

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

Figure B.1. G-units, g-phrases and phases in the context of THGLM (based on (McNeill, 1992; Kendon, 2004; Kita, et al., 1998)).

B.2. Description

According to THGLM, the time to execute a task is equal to the sum of all g-units needed to describe that task (formula B.1 below). 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. 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 B.2). 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 B.3; bear in mind two g-phrases are distinguished). H-phrase time is equal to the sum of feedback time plus exit time (formula B.4 and according to (Müller-Tomfelde, 2007)). (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 B.5); i.e., summing up the times of each of the needed operators.

$$T_{execute} = \sum_{i=1}^{m} T_{\mathrm{Gunit}_i} \qquad (B.1)$$

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

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

$$H = feedback\_time + exit\_time \tag{B.4}$$

$$T_{\text{stroke}} = \sum_{op \in OP} n * op \tag{B.5}$$

Where OP is the set of operators (Table B.1), and n is the number of occurrences of each operator.

THGLM contains three groups of operators derived from the previous model formulation: expressive (S-phrase and H-phrase), movement and general operators. Table B.1 summarizes the operators that belong to each category with the corresponding time values or formulas (see Chapters 5 and 6 for further details). This list contains the operators studied in this thesis, but other operators could be added in the future as demonstrated in Chapter 7.

Table B.1. Final list of the proposed operators with the corresponding values. [a] This value corresponds to the total time of holding (i.e., 1 second). [b] According to Chapter 4.

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
		U, Pulling		Moving the hand toward the back.	0.941	0.121
		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, $D_c(n_D, l_D, n_C)$ [b]		"Drawing" shapes, numbers, etc. in the air. 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.	$a\, n_D + b\, l_D + c\, n_C$	N/A

<table>
<tr><th colspan="3">Operators</th><th>Description</th><th>Time (s)</th><th>SD (s)</th></tr>
<tr><td rowspan="7">Movement</td><td colspan="2">Pr, Preparation</td><td>(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.</td><td>0.452</td><td>0.103</td></tr>
<tr><td colspan="2">Re, Retraction</td><td>(Optional) Moving the hand from the position where a stroke or hold finishes to a resting position.</td><td>0.746</td><td>0.106</td></tr>
<tr><td rowspan="3">Sp, Swipe preparation</td><td>Mean</td><td rowspan="3">Preparing the hand for next swipe.</td><td>0.624</td><td>0.325</td></tr>
<tr><td>Horizontal</td><td>0.562</td><td>0.361</td></tr>
<tr><td>Vertical</td><td>0.685</td><td>0.274</td></tr>
<tr><td colspan="2">P, Pointing</td><td>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.</td><td>1.046</td><td>N/A</td></tr>
<tr><td colspan="2">Hp, Hand preference</td><td>A factor that is applied when the non-dominant hand is used. It is used only with P operator.</td><td>11%</td><td>N/A</td></tr>
<tr><td rowspan="4">General</td><td rowspan="3">M, Mentally prepare</td><td>General</td><td rowspan="3">Mentally preparing to execute subsequent physical operations.</td><td>0.927</td><td>0.116</td></tr>
<tr><td>Simple reaction ($\boldsymbol{M_S}$)</td><td>0.375</td><td>0.076</td></tr>
<tr><td>Physical matching ($\boldsymbol{M_P}$)</td><td>0.388</td><td>0.065</td></tr>
<tr><td colspan="2">SR(t), Response Time</td><td>The time the system needs to respond to user input.</td><td>t</td><td>N/A</td></tr>
</table>

In addition, THGLM needs a set of heuristics rules for placing ***M*** operators, which are reproduced below (based on (Card, et al., 1980; Kieras, 2001), see Section 5.5.1). The procedure to apply THGLM is also reproduced for ease of reference in the following (based on (Kieras, 2001; Holleis, 2009), see Section 5.5.2).

Set of updated heuristics for placing M operators:

- **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.

Procedure to apply THGLM:

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.

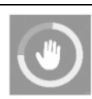
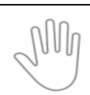
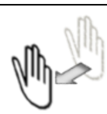
B.3. Limitations

THGLM has an acceptable quality according to the performed evaluation (Chapter 6), but there are some constraints that should be taken into account when using it. First, the method to execute the task must be known and completely specified at the level of THG using a set of operators (Table B.1). This method is assumed to be executed without errors. Second, THGLM models gestures performed by young adults in normal health conditions, and hence, the model performance may decrease depending on users' characteristics. Moreover, the target users are people with basic or no experience with touchless interaction because the model was developed with the participation of novice users. (Some recommendations are also provided in Section 5.5 to tackle this limitation.) Third, the gestures to be used as part of the intended UI have to be represented using operators. Although we provide a set of these operators, the list is not exhaustive. Other operators can be included as needed after estimating the corresponding values (Chapter 7). Fourth, only one-handed gestures (i.e., whole hand gestures without considering fingers) can be used with the current model version. Finally, though THGLM forecasts performance time acceptably, other aspects (e.g., fatigue and errors) should be considered in the design and evaluation of NUIs based on THG.

Appendix C: Touchless Hand Gestures Commonly Used in NUIs

The following table (C.1) presents details on the systematic literature review described in Chapter 5 (Section 5.2). It contains the list of gestures extracted from selected published papers. It therefore gives an idea of the most common gestures. The papers are classified in two groups (listed in Additional Bibliography). Group 1 comprises the articles that describe software based on THG for specific applications, whereas Group 2 includes the papers that study specific gestures, for general or particular purposes. Gestures with few occurrences were discarded, like the case of “rotate” that was used in just three studies. The number of papers in Group 1 is 27 and the number of papers in Group 2 is 25.

Table C.1. Touchless hand gestures commonly used in NUIs or touchless interaction according to the performed literature review. Gestures are ordered by the total number of papers.

Gesture			Number of papers		
Name	**Depiction**	**Explanation**	**Total**	**Group 1**	**Group 2**
Tap/push		Move the hand toward the front.	24	12	12
Horizontal swipe		Move the hand from right to left or vice versa, one time and returning to the starting position.	20	13	7
Dwell/ hold		Hold the hand on a target a pre-set time.	16	9	7
Vertical swipe		Move the hand from top to bottom or vice versa, one time and returning to the starting position.	15	9	6
Grab/grip		Close the hand.	13	7	6
Release		Open the hand.	8	5	3
Draw		Draw/trace figures of shapes, letters, etc. in the air.	8	4	4
Pull		Move the hand toward the back.	7	5	2
Wave		Wave the hand (i.e., the movement of the hand that people use to say hello or goodbye).	7	4	3

Appendix D: Questionnaires used for the Studies

D.1 Example of Demographics Questionnaire

Personal data			
Name:			
Age:			
Sex:	☐ Male	☐ Female	
Occupation:			
Nationality:			
Dominant hand:	☐ Left	☐ Right	

Expertise in using computers	
Main tasks (describe)	

Expertise in using touchless interaction	
Devices (select one or more)	☐ MS-Kinect ☐ Leap Motion ☐ Intel Real Sense ☐ Others: ___________
Main tasks or experience using the selected device(s) (describe)	

Thanks for your participation!

D.2 Questionnaire for the study described in Section 6.3

Questionnaire used to evaluate the document "Using the Touchless Hand Gesture Level Model to Estimate Execution Times"

You are asked below to provide information and your opinion on the document entitled "Using the Touchless Hand Gesture Level Model to Estimate Execution Times" that you used to analyze the designs of a hypothetical application for group work which is based on touchless hand gestures. Your answers will not be wrong in any case, and your opinion will be a contribution to improve the work done so far. We would appreciate your cooperation.

Part 1. Time

Indicate the time needed to perform each of the requested points.

a) Time required for reading and grasping the model with the procedure; i.e., understanding the document.

minutes

b) Time required to accomplish each of the proposed tasks:

Task	**Time (minutes)**	
	P3	**P4**
A		
B		
C		

Part 2. Document Evaluation

Evaluate each of the following statements using a scale from 1 to 7, where 1 means "total disagreement", and 7 means "full agreement". Moreover, provide the remarks (problems, suggestions, etc.) that you consider appropriate for each statement.

a) The explanation of the model (Section 2) is clear and understandable.

Score	

Remarks:

b) The heuristic and recommendations (Section 3) to use the mentally prepare operator (***M***) are clearly explained and could be used.

Score	

Remarks:

c) The procedure for using the model (section 4) is clear, understandable and you were able to apply it to the proposed designs.

Score	

Remarks:

d) The examples are clearly explained and helpful in understanding the use of the model and applying it to the proposed designs.

Score	

Remarks:

Part 3. Overall Evaluation

Evaluate in a general/comprehensive way the document that describes the model together with the procedure to use it. Use a scale from 1 to 7, where 1 is insufficient and 7 is satisfactory.

Score	

Remarks:

Thanks for your valuable contribution!!

D.3 Questionnaire for the study described in Chapter 7

Questionnaire on preferences in the use of hands as part of the investigation "A Predictive Model for User Performance Time with Natural User Interfaces based on Touchless Hand Gestures" [19]

Please indicate your preference in the use of hands in the following activities marking the boxes as follows:

x	Preference (one box)
xx	Strong preference (two boxes)
blank	No preference

		Left		Right	
1	Writing				
2	Drawing				
3	Throwing				
4	Scissors				
5	Toothbrush				
6	Knife (without fork)				
7	Spoon				
8	Broom (upper hand)				
9	Striking a match				
10	Opening box (lid)				

[19] This questionnaire is based on (Oldfield, 1971; MacKenzie, 2013).

Glossary

Articulations — the ways in which users produce gestures in the air.

Descriptive model — a kind of model used in HCI to approach usability problems. It provides a framework for designers to delineate and/or reflect on the problems usually based on depictions or verbal descriptions.

Fitts' law — a predictive model to estimate the time it takes to point at a target, based on the object size and distance.

GCP model — a descriptive model that helps in understanding gestures conception and production.

Gesture — a user's body motion that conveys information to the application with which s/he is interacting.

Gesture phrase — (g-phrase) a unit of visible bodily action that occurs within a gesture unit.

Gesture space — the input physical area/space where users perform the gestures.

Gesture unit — (g-unit) an entire "excursion" between successive rests of the limbs that begins the moment the limb begins to move and ends when it has reached a resting position again.

Gesture vocabulary — the term used to refer to the set of gestures utilized to interact with the application.

Hold — a single meaningful still phase that is used for static gestures instead of a stroke.

Keystroke Level Model (KLM) — a predictive model that allows forecasting the execution part of a task using a set of operators, which in most cases are assumed to take a constant time.

Mental act — the mental preparation needed to perform an action. KLM and THGLM use a generic operator, ***M***, that subsumes the corresponding cognitive processes.

Natural user interface (NUI) — a UI that may appear "natural" to users because little to no training is needed to use it. Users interact with it through touchless hand gestures.

Performance time — the period a user needs to accomplish a set of tasks using a system.

Predictive model — a kind of model employed in HCI to approach usability problems. Predictive models can be used to objectively estimate the required time, errors, etc. for performing a set of user interactions by using formalisms or equations.

Preparation — an optional phase in which the body parts are moved to the starting position of a stroke.

$\boldsymbol{R^2}$ — a metric of model validity that reflects the strength of the relationship between predicted and observed times.

Reaction time — the delay between a fixed (or reaction) stimulus and the initiation of a response (e.g., a detectable movement).

Recovery — the final and optional phase of a g-unit in which the utilized body parts return to a resting position (the original or another one).

Retraction — see Recovery.

RMSE (root mean square error) — a metric of model validity that shows the percentage difference between predicted and observed values.

Stroke — the phase (of a gesture phrase) in which the peak of effort of a gesture is expressed.

Stroke time — the time required to produce the stroke of a gesture.

Touchless Gesture (TG) — a gesture performed in the air with one or more body parts and without haptic contact.

Touchless Hand Drawing Gesture (THDG) — a touchless hand gesture that consist in tracing/drawing figures of shapes, letters, numbers, etc. in the air.

Touchless Hand Gesture (THG) — a touchless gesture performed with one or both hands.

Touchless Hand Gesture Level Model (THGLM) — a predictive model that allows forecasting the time to execute a task given a method (expressed using gesture-units and THG-level actions) and using the corresponding formulas.

Wizard of Oz — a kind of study in which participants believe they are interacting “normally” with a system that provides the results or information but responses are actually given by a human operator (the “wizard”).

Bibliography

Accot, J., & Zhai, S. (1997). Beyond Fitts' Law: Models for Trajectory-Based HCI Tasks. *Proceedings of CHI 1997, pp. 295-302. ACM Press.*

Accot, J., & Zhai, S. (1999). Performance Evaluation of Input Devices in Trajectory-based Tasks: An Application of the Steering Law. *Proceedings of CHI 1999, pp. 466-472. ACM Press.*

Aigner, R., Wigdor, D., Benko, H., Haller, M., Lindbauer, D., Ion, A., Zhao, S., & Koh, J. (2012). *Understanding mid-air hand gestures: A study of human preferences in usage of gesture types for hci.* Microsoft Research TechReport MSR-TR-2012-111. https://www.microsoft.com/en-us/research/publication/understanding-mid-air-hand-gestures-a-study-of-human-preferences-in-usage-of-gesture-types-for-hci/.

Anderson, F., & Bischof, W. F. (2013). Learning and Performance with Gesture Guides. *Proceedings of CHI 2013, pp. 1109-1118. ACM Press*, (pp. 1109-1118).

Annett, M., & Bischof, W. (2013). Your Left Hand Can Do It Too! Investigating Intermanual, Symmetric Gesture Transfer on Touchscreens. *Proceedings of CHI 2013, pp. 1119-1128. ACM Press.*

Anthony, L., & Wobbrock, J. O. (2010). A Lightweight Multistroke Recognizer for User Interface Prototypes. *Proceedings of Graphics Interface 2010, pp. 245-252. ACM Press.*

Argelaguet, F., & Andujar, C. (2013). A survey of 3D object selection techniques for virtual environments. *Computers & Graphics, 37*(3), 121-136.

Bachynskyi, M., Oulasvirta, A., Palmas, G., & Weinkauf, T. (2014). Is Motion-Capture-Based Biomechanical Simulation Valid for HCI Studies? Study and Implications. *Proceedings of CHI 2014, pp. 3215-3224. ACM Press.*

Bachynskyi, M., Palmas, G., Oulasvirta, A., & Weinkauf, T. (2015). Informing the design of novel input methods with muscle coactivation clustering. *ACM Transactions on Computer-Human Interaction (TOCHI), 21*(6), 30:1-30:25.

Ball, R., & Hourcade, J. P. (2011). Rethinking Reading for Age From Paper and Computers. *International Journal of Human-Computer Interaction, 27*(11), 1066-1082.

Barclay, K., Wei, D., Lutteroth, C., & Sheehan, R. (2011). A Quantitative Quality Model for Gesture Based User Interfaces. *Proceedings of OzCHI 2011, pp. 31-39. ACM Press.*

Baudel, T., & Beaudouin-Lafon, M. (1993). Charade: Remote Control of Objects Using Free-Hand Gestures. *Communications of the ACM, 36*(7), 28-35.

Blake, J. (2012). The natural user interface revolution. In *Natural User Interfaces in .Net* (pp. 1-43). Manning.

Blum, T., Kleeberger, V., Bichlmeier, C., & Navab, N. (2012). mirracle: An Augmented Reality Magic Mirror System for Anatomy Education. *IEEE Virtual Reality Short Papers and Posters, pp. 115-116. IEEE Press.*

Bolt, R. A. (1980). "Put-that-there": Voice and gesture at the graphics interface. *Proceedings of SIGGRAPH 1980, pp. 262-270. ACM Press.*

Boulos, M., Blanchard, B., Walker, C., Montero, J., Tripathy, A., & Gutierrez-Osuna, R. (2011). Web GIS in practice X: a Microsoft Kinect natural user interface for Google Earth navigation. *International Journal of Health Geographics, 10*(45).

Buxton, W. (1990). A Three State Model of Graphical Input. *D. Diaper et al. (Eds), Human-Computer Interaction - INTERACT 1990, pp. 449-456. Elsevier Science Publishers.*

Cao, X., & Zhai, S. (2007). Modeling Human Performance of Pen Stroke Gestures. *Proceedings of CHI 2007, pp.1495-1504. ACM Press*, (pp. 1495-1504).

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

Card, S., Moran, T., & Newell, A. (1983). *The Psychology of Human-Computer Interaction.* L. Erlbaum Associates.

Choi, E., Kim, H., & Chung, M. K. (2014). A taxonomy and notation method for three-dimensional hand gestures. *International Journal of Industrial Ergonomics, 44*(1), 171-188.

Cockburn, A., Gutwin, C., & Greenberg, S. (2007). A Predictive Model of Menu Performance. *Proceedings of CHI 2007, pp. 627-636. ACM Press.*

Cohn, G., Morris, D., Patel, S., & Tan, D. (2012). Humantenna: Using the Body as an Antenna for Real-Time Whole-Body Interaction. *Proceedings of CHI 2012, pp. 1901-1910. ACM Press.*

Dahlbäck, N., Jönsson, A., & Ahrenberg, L. (1993). Wizard of Oz studies: why and how. *Proceedings of IUI 1993, pp. 193-200. ACM Press.*

Erazo, O., & Pino, J. (2014). Estimating the Difficulty of Touchless Hand Gestures. *IEEE Latin America Transactions, 12*(1), 17-22.

Erazo, O., & Pino, J. A. (2013). Analyzing Touchless Hand Gestures Performance. *Proceedings of ChileCHI 2013, pp. 14-17. ACM Press.*

Erazo, O., & Pino, J. A. (2015). Predicting Task Execution Time on Natural User Interfaces based on Touchless Hand Gestures. *Proceedings of IUI 2015, pp. 97-109. ACM Press.*

Erazo, O., & Pino, J. A. (2016). Predicting User Performance Time for Hand Gesture Interfaces. *Submitted (under evaluation) to International Journal of Industrial Ergonomics, 2016*.

Erazo, O., Baloian, N., Pino, J. A., & Zurita, G. (2016). Easing Students' Participation in Class with Hand Gesture Interfaces. *Accepted paper for UCAMI conference, Las Palmas de Gran Canaria, Spain, 2016.*

Erazo, O., Pino, J. A., & Antunes, P. (2015). Estimating Production Time of Touchless Hand Drawing Gestures. *J. Abascal et al. (Eds.): INTERACT 2015, Part III, LNCS, vol. 9298, pp. 552–569. Springer, Heidelberg.*

Erazo, O., Pino, J. A., Pino, R., & Fernández, C. (2014). Magic Mirror for Neurorehabilitation of People with Upper Limb Dysfunction Using Kinect. *Proceedings of HICSS 2014, pp. 2607-2615. IEEE Press.*

Ferreira, A, Antunes, P., & Pino, J.A. (2009). Evaluating Shared Workspace Performance using Human Information Processing Models. *Information Research*, *14*(1), paper 388.

Fikkert, W., Van Der Vet, P., van der Veer, G., & Nijholt, A. (2009). Gestures for Large Display Control. *S. Kopp and I. Wachsmuth (Eds.): GW 2009, LNAI, vol. 5934, pp. 245-256. Springer, Heidelberg.*

Fitts, P. M. (1954, June). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology, 47*(6), 381-391.

Fothergill, S., Mentis, H., Kohli, P., & Nowozin, S. (2012). Instructing People for Training Gestural Interactive Systems. *Proceedings of CHI 2012, pp. 1737-1746. ACM Press.*

Frigg, R., & Hartmann, S. (2012). Models in Science. In *The Stanford Encyclopedia of Philosophy (Fall 2012 Edition), Edward N. Zalta (ed.). URL=http://plato.stanford.edu/archives/fall2012/entries/models-science.*

Gallo, L., Placitelli, A. P., & Ciampi, M. (2011). Controller-free exploration of medical image data: Experiencing the Kinect. *Proceedings of CBMS 2011, pp. 1-6. IEEE Press.*

Gibbon, D., Gut, U., Hell, B., Looks, K., & Thies, A. T. (2003). A computational model of arm gestures in conversation. *INTERSPEECH.*

Giovanni, S., Choi, Y. C., Huang, J., Khoo, E. T., & Yin, K. (2012). Virtual Try-On Using Kinect and HD Camera. *M. Kallmann and K. Bekris (Eds.): MIG 2012, LNCS, vol. 7660, pp. 55–65. Springer, Heidelberg*.

Giuliani, M., & Knoll, A. (2008). MultiML – A General Purpose Representation Language for Multimodal Human Utterances. *Proceedings of ICMI 2008, pp. 165-172. ACM Press.*

Guiard, Y. (1987). Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. *Journal of Motor Behaviour, 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.

Guo, B., Sun, J., Wei, Y. C., Meekhof, C., & Leyvand, T. (2011). Kinect identity: Technology and Experience. *Computer, 4*, 94-96.

Gupta, S., Chen, K. Y., Reynolds, M. S., & Patel, S. N. (2011). LightWave: Using Compact Fluorescent Lights as Sensors. *Proceedings of UbiComp 2011, pp. 65-74. ACM Press.*

Gupta, S., Morris, D., Patel, S., & Tan, D. (2012). Soundwave: using the doppler effect to sense gestures. *Proceedings of CHI 2012, pp. 1911-1914. ACM Press.*

Hayashi, E., Maas, M., & Hong, J. I. (2014). Wave to Me: User Identification Using Body Lengths and Natural Gestures. *Proceedings of CHI 2014, pp. 3453-3462. ACM Press.*

Henschke, M., Gedeon, T., & Jones, R. (2015). Touchless Gestural Interaction with Wizard-of-Oz: Analysing User Behaviour. *Proceedings of OzCHI 2015, pp. 207-211. ACM Press.*

Herrera-Acuña, R., Argyriou, V., & Velastin, S. A. (2015). A Kinect-based 3D hand-gesture interface for 3D databases. *Journal on Multimodal User Interfaces, 9*(2), 121-139.

Hespanhol, L., Tomitsch, M., Grace, K., Collins, A., & Kay, J. (2012). Investigating Intuitiveness and Effectiveness of Gestures for Free Spatial Interaction with Large Displays. *Proceedings of PerDis 2012. ACM Press.*

Hick, W. E. (1952). On the rate of gain of information. *Quarterly Journal of Experimental Psychology, 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. *Proceedings of CHI 2014, pp.1063-1072. ACM Press* (pp. 1063-1072). ACM.

Hoff, L., Hornecker, E., & Bertel, S. (2016). Modifying Gesture Elicitation: Do Kinaesthetic Priming and Increased Production Reduce Legacy Bias? *Proceedings of TEI 2016, pp. 86-91. ACM Press.*

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

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 forAdvanced Mobile Phone Interaction. *Proceedings of CHI 2007, pp. 1505-1514. ACM Press.*

Höysniemi, J., Hämäläinen, P., & Turkki, L. (2004). Wizard of Oz Prototyping of Computer Vision Based Action Games for Children. *Proceedings of IDC 2004, pp. 27-34. ACM Press.*

Hsu, H. M. (2011). The Potential of Kinect in Education. *International Journal of Information and Education Technology, 1*(5), 365-370.

Hyman, R. (1953). Stimulus information as a determinant of reaction time. *Journal of experimental psychology, 45*(3), 188-196.

Ismair, S., Wagner, J., Selker, T., & Butz, A. (2015). Mime: Teaching mid-air pose-command mappings. *Proceedings of MobileHCI 2015, pp. 199-206. ACM Press.*

Isokoski, P. (2001). Model for Unistroke Writing Time. *Proceedings of CHI 2011, pp. 357-364. ACM Press.*

Jagodziński, P., & Wolski, R. (2014). Assessment of Application Technology of Natural User Interfaces in the Creation of a Virtual Chemical Laboratory. *Journal of Science Education and Technology, 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. *Proceedings of CHI 2004, pp. 455-462. ACM Press,* (pp. 455-462).

Jude, A., Poor, G. M., & Guinness, D. (2014a). An Evaluation of Touchless Hand Gestural Interaction for Pointing Tasks with Preferred and Non-preferred Hands. *Proceedings of NordiCHI 2014, pp. 668-676. ACM Press.*

Jude, A., Poor, G. M., & Guinness, D. (2014b). Personal Space: User Defined Gesture Space for GUI Interaction. *Extended Abstracts of CHI 2014, pp. 1615-1620. ACM Press.*

Kabbash, P., MacKenzie, I. S., & Buxton, W. (1993). Human Performance Using Computer Input Devices in the Preferred and Non-Preferred Hands. *Proceedings of INTERACT 1993 and CHI 1993, pp. 474-481. ACM Press*, (pp. 474-481).

Kamal, A., Li, Y., & Lank, E. (2014). Teaching Motion Gestures via Recognizer Feedback. *Proceedings of IUI 2014, pp. 73-82. ACM Press.*

Karam, M., & Schraefel, M. (2005). *A taxonomy of Gestures in Human Computer Interactions.* University of Southampton.

Kendon, A. (2004). Gesture units, gesture phrases and speech. In *Gesture: Visible action as utterance* (pp. 108-126). Cambridge University Press.

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

Kieras, D. (2003). Model-based Evaluation. In *The Human-Computer Interaction Handbook* (pp. 1191-1208). New Jersey: Lawrence Erlbaum Associates.

Kim, K., & Ren, X. (2014). Assisting Visually Impaired People to Acquire Targets on a Large Wall-Mounted Display. *Journal of Computer Science and Technology, 29*(5), 825-836.

Kipp, M. (2001). ANVIL A Generic Annotation Tool for Multimodal Dialogue. *Proceedings of Eurospeech, pp. 1367-1370.*

Kistler, F., & André, E. (2013). User-Defined Body Gestures for an Interactive Storytelling Scenario. *P. Kotzé et al. (Eds.): INTERACT 2013, Part II, LNCS, vol. 8118, pp. 264-281. Springer, Heidelberg.*

Kita, S., Van Gijn, I., & Van der Hulst, H. (1998). Movement Phases in Signs and Co-Speech Gestures, and their Transcription by Human Coders. *I. Wachsmuth and M. Fröhlich (Eds.): Gesture and Sign Language in Human-Computer Interaction. LNCS, vol. 1371, pp. 23-35. Springer, Heidelberg.* Springer Berlin Heidelberg.

Kitchenham, B. (2004). *Procedures for Performing Systematic Reviews.* Joint Technical Report, Computer Science Department, Keele University (TR/SE-0401) and National ICT Australia Ltd. (0400011T.1).

Kohli, L. (2013). Warping Virtual Space for Low-Cost Haptic Feedback. *Proceedings of SIGGRAPH Symposium on I3D 2013, pp. 195. ACM Press.*

Kölsch, M., Beall, A. C., & Turk, M. (2003). The postural comfort zone for reaching gestures. In S. Publications (Ed.), *Proceedings of the human factors and ergonomics society annual meeting, vol. 47(4), pp. 787-791. SAGE Publications* (pp. 787-791). SAGE Publications.

Koperski, J. (2016). Models. In *The Internet Encyclopedia of Philosophy. URL=http://www.iep.utm.edu/models/.*

Kosinski, B., & Cummings, J. (1999). *The Scientific Method: An Introduction Using Reaction Time.* Tested studies for laboratory teaching (ABLE Proceedings) 25.

Koutsabasis, P., & Domouzis, C. K. (2016). Mid-Air Browsing and Selection in Image Collections. *Proceedings of AVI 2016, pp. 21-27. ACM Press.*

Kratz, S., & Rohs, M. (2011). Protractor3D: A Closed-Form Solution to Rotation-Invariant 3D Gestures. *Proceedings of IUI 2011, pp. 371-374. ACM Press.*

Kurtenbach, G., & Hulteen, E. A. (1990). Gestures in human-computer communication. In *The art of human-computer interface design* (pp. 309-317). Addison-Wesley.

Lausberg, H., & Sloetjes, H. (2009). Coding gestural behavior with the NEUROGES-ELAN system. *Behavior research methods, 41*(3), 841-849.

LaViola, J. J. (2013). 3D Gestural Interaction: The State of the Field. *International Scholarly Research Notices Artificial Intelligence, Article No. 514641*.

Law, A. M., Kelton, W. D., & Kelton, W. D. (1991). Basic Simulation Modeling. In *Simulation Modeling and Analysis* (pp. 1-6). New York: McGraw-Hill.

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

Lisboa, L. B., Garcia, V. C., Lucrédio, D., de Almeida, E. S., de Lemos Meira, S. R., & de Mattos Fortes, R. P. (2010). A systematic review of domain analysis tools. *Information and Software Technology, 52*(1), 1-13.

Livingston, M. A., Sebastian, J., Ai, Z., & Decker, J. W. (2012). Performance Measurements for the Microsoft Kinect Skeleton. *Proceedings of Virtual Reality Workshops 2012, pp. 119-120. IEEE Press* .

Lubos, P., Bruder, G., & Steinicke, F. (2014). Analysis of direct selection in head-mounted display environments. *Proceedings of 3DUI, pp. 11-18. IEEE Press.*

Luo, L., & John, B. E. (2005). Predicting Task Execution Time on Handheld Devices Using the Keystroke-Level Model. *Extended Abstracts of CHI 2005, pp. 1605-1608. ACM Press.*

MacKenzie, I. S. (2003). Motor Behavior Models for Human-Computer Interaction. In *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science* (pp. 27-54). San Francisco: Morgan Kaufmann.

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. *Proceedings of NordiCHI 2012, pp. 568-577. ACM Press.*

Marin, G., Dominio, F., & Zanuttigh, P. (2015). Hand gesture recognition with jointly calibrated Leap Motion and depth sensor. *Multimedia Tools and Applications*.

McNeill, D. (1992). Guide to Gesture Classification, Transcription, and Distribution. In *Hand and Mind: What Gestures Reveal about Thought* (pp. 75-104). The University of Chicago Press.

Mehler, A., Lücking, A., & Abrami, G. (2014). WikiNect: Image schemata as a basis of gestural writing for kinetic museum wikis. *Universal Access in the Information Society, 14*(3), 333-349.

Microsoft Corporation. (2013). *Kinect for Windows Human Interface Guidelines v1.8.0.* Retrieved from http://msdn.microsoft.com/en-us/library/jj663791.aspx

Morris, M. R., Danielescu, A., Drucker, S., Fisher, D., Lee, B., & Wobbrock, J. O. (2014). Reducing Legacy Bias in Gesture Elicitation Studies. *Interactions, 21*(3), 40-45.

Morton, A., & Suárez, M. (2001). Kinds of Models. In *Model Validation in Hydrological Science* (pp. 11-21). Eds. P. Bates and M. Anderson. New York: John Wiley Press.

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

Mulligan, M., & Wainwright, J. (2004). Modelling and Model Building. In *Environmental Modelling: Finding Simplicity in Complexity* (pp. 7-17). John Wiley & Sons.

Murata, A., & Iwase, H. (2001). Extending Fitts' law to a three-dimensional pointing task. *Human movement science, 20*(6), 791–805.

Nacenta, M. A., Kamber, Y., Qiang, Y., & Kristensson, P. O. (2013). Memorability of Pre-designed and User-defined Gesture Sets. *Proceedings of CHI 2013, pp. 1099-1108. ACM Press.*

Nancel, M., Wagner, J., Pietriga, E., Chapuis, O., & Mackay, W. (2011). Mid-air Pan-and-Zoom on Wall-sized Displays. *Proceedings of CHI 2011, pp. 177-186. ACM Press.*

Neff, M., Kipp, M., Albrecht, I., & Seidel, H. P. (2008). Gesture Modeling and Animation Based on a Probabilistic Recreation of Speaker Style. *ACM Transactions on Graphics (TOG), 27*(1).

Nielsen, M., Moeslund, T., Störring, M., & Granum, E. (2008). Gesture Interfaces. In *HCI beyond the GUI: Design for haptic, speech, olfactory, and other nontraditional interfaces* (pp. 75-106). Morgan Kaufmann.

Nielsen, M., Störring, M., Moeslund, T. B., & Granum, E. (2004). A Procedure for Developing Intuitive and Ergonomic Gesture Interfaces for HCI. *A. Camurri and G. Volpe (Eds.): GW 2003, LNAI, vol. 2915, pp. 409–420. Springer, Heidelberg, 2915.*

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.*

Obaid, M., Häring, M., Kistler, F., Bühling, R., & André, E. (2012). User-Defined Body Gestures for Navigational Control of a Humanoid Robot. *S.S. Ge et al. (Eds.): ICSR 2012, LNAI, vol. 7621, pp. 367-377. Springer, Heidelberg.*

Oh, U., & Findlater, L. (2013). The Challenges and Potential of End-user Gesture Customization. *Proceedings of CHI 2013, 1129–1138. ACM Press.*

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

Panger, G. (2012). Kinect in the Kitchen: Testing Depth Camera Interactions in Practical Home Environments. *Extended Abstracts of CHI 2012, pp. 1985-1990. ACM Press.*

Pavlovic, V. I., Sharma, R., & Huang, T. S. (1997). Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(7), 677-695.

Peters, M., & Ivanoff, J. (1999). Performance Asymmetries in Computer Mouse Control of Right-Handers, and Left-Handers With Left-and Right-Handed Mouse Experience. *Journal of Motor Behavior, 31*(1), 86-94.

Pettitt, M., Burnett, G., & Stevens, A. (2007). An Extended Keystroke Level Model (KLM) for Predicting the Visual Demand of In-Vehicle Information Systems. *Proceedings of CHI 2007, pp. 1515-1524. ACM Press.*

Pino, A., Tzemis, E., Ioannou, N., & Kouroupetroglou, G. (2013). Using Kinect for 2D and 3D Pointing Tasks: Performance Evaluation. *M. Kurosu (Ed.): HCII 2013, Part IV, LNCS, vol. 8007, pp. 358-367. Springer, Heidelberg.*

Piumsomboon, T., Clark, A., Billinghurst, M., & Cockburn, A. (2013). User-Defined Gestures for Augmented Reality. *P. Kotzé et al. (Eds.): INTERACT 2013, Part II, LNCS, vol. 8118, pp. 282-299. Springer, Heidelberg.*

Polacek, O., Klíma, M., Sporka, A. J., Zak, P., Hradis, M., Zemcik, P., & Procházka, V. (2012). A Comparative Study on Distant Free-Hand Pointing. *Proceedings of EuroiTV 2012, pp. 139-142. ACM Press.*

Quek, F., McNeill, D., Bryll, R., Duncan, S., Ma, X., Kirbas, C., McCullough, K.E., & Ansari, R. (2002). Multimodal Human Discourse: Gesture and Speech. *ACM Transactions on Computer-Human Interaction (TOCHI), 9*(3), 171-193.

Rekik, Y., Grisoni, L., & Roussel, N. (2013). Towards Many Gestures to One Command: User Study for Tabletops. *P. Kotzé et al. (Eds.): INTERACT 2013, Part II, LNCS, vol. 8118, pp. 246-263. Springer, Heidelberg.*

Rekik, Y., Vatavu, R., & Grisoni, L. (2014a). Match-up & Conquer: A Two-step Technique for Recognizing Unconstrained Bimanual and Multi-finger Touch Input. *Proceedings of AVI 2014, pp. 201–208. ACM Press.*

Rekik, Y., Vatavu, R., & Grisoni, L. (2014b). Understanding Users' Perceived Difficulty of Multi-Touch Gesture Articulation. *Proceedings of ICMI 2014, pp. 232–239. ACM Press.*

Ren, G., Li, C., O'Neill, E., & Willis, P. (2013). 3D Freehand Gestural Navigation for Interactive Public Displays. *Computer Graphics and Applications, 33*(2), 47-55.

Rico, J., & Brewster, S. (2010). Usable Gestures for Mobile Interfaces: Evaluating Social Acceptability. *Proceedings of CHI 2010, pp. 887-896. ACM Press.*

Rosenbaum, D. A. (2009). *Human Motor Control.* San Diego: Academic Press, Elsevier.

Ruiz, J., Bunt, A., & Lank, E. (2008). A Model of Non-Preferred Hand Mode Switching. *Proceedings of Graphics Interface 2008, pp. 49-56. ACM Press.*

Sakoe, H., & Chiba, S. (1978). Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 26*(1), 43-49.

Sambrooks, L., & Wilkinson, B. (2013). Comparison of gestural, touch, and mouse interaction with Fitts' law. *Proceedings of OzCHI 2013, pp. 119-122. ACM Press.*

Schwaller, M., & Lalanne, D. (2013). Pointing in the Air: Measuring the Effect of Hand Selection Strategies on Performance and Effort. *A. Holzinger et al. (Eds.): SouthCHI 2013, LNCS, vol. 7946, pp. 732–747. Springer, Heidelberg, 7946.*

Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., & Moore, R. (2013). Real-Time Human Pose Recognition in Parts from Single Depth Images. *Communications of the ACM, 56*(1), 116-124.

Song, K., Kim, J., Cho, Y. H., Lee, A., Ryu, H., Choi, J. W., & Lee, Y. J. (2013). The Fingerstroke-Level Model Strikes Back: A modified Keystroke-Level Model in developing a gaming UI for 4G networks. *Extended Abstracts of CHI 2013, pp. 2359-2362. ACM Press.*

Spano, L. D., Cisternino, A., & Paternò, F. (2012). A Compositional Model for Gesture Definition. *M. Winckler et al. (Eds.): HCSE 2012, LNCS, vol. 7623, pp. 34-52. Springer, Heidelberg.*

Sridhar, S., Feit, A. M., Theobalt, C., & Oulasvirta, A. (2015). Investigating the Dexterity of Multi-Finger Input for Mid-Air Text Entry. *Proceedings of CHI 2015, pp. 3643-3652. ACM Press,* (pp. 3643-3652).

Stanton, N. A., & Young, M. S. (2003). Giving ergonomics away? The application of ergonomics methods by novices. *Applied Ergonomics, 34*(5), 479-490.

Sturman, M. M., Vaillancourt, D. E., & Corcos, D. M. (2005). Effects of Aging on the Regularity of Physiological Tremor. *Journal of Neurophysiology, 93*(6), 3064-3074.

Sutter, C., Müsseler, J., Bardos, L., Ballagas, R., & Borchers, J. (2008). The impact of gain change on perceiving one's own actions. *Mensch & computer*, 147-156.

Tu, H., Ren, X., & Zhai, S. (2012). A Comparative Evaluation of Finger and Pen Stroke Gestures. *Proceedings of CHI 2012, pp. 1287-1296. ACM Press.*

Vatavu, R. D. (2012). User-Defined Gestures for Free-Hand TV Control. *Proceedings of EuroiTV 2012, pp. 45-48. ACM Press.*

Vatavu, R. D., Casiez, G., & Grisoni, L. (2013). Small, Medium, or Large? Estimating the User-Perceived Scale of Stroke Gestures. *Proceedings of CHI 2013, pp. 277-280. ACM Press.*

Vatavu, R., Vogel, D., Casiez, G., & Grisoni, L. (2011). Estimating the Perceived Difficulty of Pen Gestures. *P. Campos et al. (Ed.): INTERACT 2011, Part II, LNCS, vol. 6947, pp. 89-106. Springer, Heidelberg.*

Wachs, J. P., Kölsch, M., Stern, H., & Edan, Y. (2011). Vision-based hand-gesture applications. *Communications of the ACM, 54*(2), 60-71.

Walter, R., Bailly, G., & Müller, J. (2013). StrikeAPose: Revealing Mid-Air Gestures on Public Displays. *Proceedings of CHI 2013, pp. 841-850. ACM Press.*

Walter, R., Bailly, G., Valkanova, N., & Müller, J. (2014). Cuenesics: Using Mid-Air Gestures to Select Items on Interactive Public Displays. *Proceedings of MobileHCI 2014, pp. 299-308. ACM Press.*

Webb, J., & Ashley, J. (2012). NUI. In *Beginning Kinect Programming with the Microsoft Kinect SDK* (pp. 170-172). Apress.

Webster, J., & Watson, R. T. (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Quarterly, 26*(2), 13-23.

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.

Weisberg, M. (2003). *When less is more: Tradeoffs and idealization in model building.* Doctoral Dissertation, Stanford University.

Weisberg, M. (2007). Who is a Modeler? *British Journal for the Philosophy of Science, 58*(2), 207-233.

Whelan, R. (2008). Effective Analysis of Reaction Time Data. *The Psychological Record, 58*(3), 475-482.

Wigdor, D., & Wixon, D. (2011). The Natural User Interface. En *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture* (págs. 9-14). Boston: Morgan Kaufmann.

Wobbrock, J. O., Wilson, A. D., & Li, Y. (2007). Gestures without Libraries, Toolkits or Training: a $1 Recognizer for User Interface Prototypes. *Proceedings of UIST 2007, pp. 159-168*, (pp. 159-168).

Wobbrock, J., Morris, M., & Wilson, A. (2009). User-Defined Gestures for Surface Computing. *Proceedings of CHI 2009, pp. 1083-1092. ACM Press.*

Wong, A. L., Haith, A. M., & Krakauer, J. W. (2015). Motor Planning. *The Neuroscientist, 21*(4), 385-398.

Yoo, S., Parker, C., Kay, J., & Tomitsch, M. (2015). To Dwell or Not to Dwell: An Evaluation of Mid-Air Gestures for Large Information Displays. *Proceedings of OzCHI 2015, pp. 187-191. ACM Press.*

Zeigler, B. P., Praehofer, H., & Kim, T. G. (2000). Framework for Modeling and Simulation. In *Theory of Modeling and Simulation* (pp. 25-36). Academic Press.

Zeng, X., Hedge, A., & Guimbretiere, F. (2012). Fitts' Law in 3D Space with Coordinated Hand Movements. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting, pp. 990-994. SAGE Publications, 56.*

Zhou, Y., David, B., & Chalon, R. (2012). PlayAllAround: Wearable One-hand Gesture Input and Scalable Projected Interfaces. *Proceedings of Ergo'IHM 2012, pp. 105-112. ACM Press.*

Additional Bibliography[20]

Arif, A. S., Stuerzlinger, W., & Gordynski, A. (2014). Error Behaviours in an Unreliable In-air Gesture Recognizer. *Extended Abstracts CHI 2014, pp. 1603-1608. ACM Press.*

Aumi, M. T. I., & Kratz, S. (2014). AirAuth: Towards Attack-Resilient Biometric Authentication Using In-Air Gestures. *Extended Abstracts CHI 2014, pp. 1585-1590. ACM Press.*

Bossavit, B., Marzo, A., Ardaiz, O., & Pina, A. (2013). Hierarchical Menu Selection with a Body-Centered Remote Interface. Interact. Comput.

Brown, M. A., & Stuerzlinger, W. (2014). The Performance of Un-Instrumented In-Air Pointing. *Proceedings of GI 2014, pp. 59-66. ACM Press.*

Chattopadhyay, D., & Bolchini, D. (2014). Touchless Circular Menus: Toward an Intuitive UI for Touchless Interactions with Large Displays. *Proceedings of AVI 2014, pp. 33-40. ACM Press.*

Cockburn, A., Quinn, P., Gutwin, C., Ramos, G., & Looser, J. (2011). Air pointing: Design and evaluation of spatial target acquisition with and without visual feedback. *Int. J. of Human-Computer Studies, 69*(6), 401-414.

Cohen, L., Haliyo, S., Chetouani, M., & Régnier, S. (2014). Intention prediction approach to interact naturally with the microworld. *Proceedings of AIM 2014*. IEEE.

Dias, T., Variz, M., Jorge, P., & Jesus, R. (2013). Gesture Interaction System for Social Web Applications on Smart TVs. *Proceedings of OAIR 2013, pp. 225-226. CID.*

Drossis, G., Grammenos, D., Birliraki, C., & Stephanidis, C. (2013). MAGIC: Developing a Multimedia Gallery Supporting mid-Air Gesture-Based Interaction and Control. *Posters' Extended Abstracts HCII 2013, pp. 303-307. Springer.*

Garzotto, F., & Valoriani, M. (2013). Touchless Gestural Interaction with Small Displays: A Case Study. *Proceedings of CHItaly 2013. ACM Press.*

Gatto, I., & Pittarello, F. (2012). Prototyping a Gestural Interface for Selecting and Buying Goods in a Public Environment. *Proceedings of AVI 2012, pp. 784-785. ACM Press.*

Giovanni, S., Choi, Y. C., Huang, J., Khoo, E. T., & Yin, K. (2012). Virtual Try-On Using Kinect and HD Camera. *Proceedings of MIG 2012, pp. 55-65. Springer.*

[20] This section contains the references of the papers selected from the systematic literature review described in Chapter 5 (Section 5.2) and Appendix C.

Hartmann, F., & Schlaefer, A. (2013). Feasibility of touch-less control of operating room lights. *Int. J. CARS, 8*(2), 259-268.

Hayashi, E., Maas, M., & Hong, J. I. (2014). Wave to Me: User Identification Using Body Lengths and Natural Gestures. *Proceedings of CHI 2014, pp. 3453-3462. ACM Press.*

Hespanhol, L., Tomitsch, M., Grace, K., Collins, A., & Kay, J. (2012). Investigating Intuitiveness and Effectiveness of Gestures for Free Spatial Interaction with Large Displays. *Proceedings of PerDis 2012. ACM Press.*

Ho, K., & Weng, H. (2013). Favoured Attributes of In-Air Gestures in the Home Environment. *Proceedings of OzCHI 2013, pp. 171-174. ACM Press.*

Hosseini, M., Vallotton, P., Bednarz, T., & Sowmya, A. (2013). A study of touchless versus touch-based interactions with bacterial biofilm images. *Proceedings of VRCAI 2013, pp. 285-290. ACM Press.*

Hötker, A. M., Pitton, M. B., Mildenberger, P., & Düber, C. (2013). Speech and motion control for interventional radiology: requirements and feasibility. *Int. J. CARS, 8*(6), 997-1002.

Jacob, M. G., & Wachs, J. P. (2014). Context-based hand gesture recognition for the operating room. *Pattern Recogn. Lett. 36*, 196-203.

Jalaliniya, S., Smith, J., Sousa, M., Büthe, L., & Pederson, T. (2013). Touch-less interaction with medical images using hand & foot gestures. *Proceedings of UbiComp 2013, pp. 1265-1274. ACM Press.*

Kim, Y., Sim, S., Cho, S., Lee, W. W., Jeong, Y. S., Cho, K., & Um, K. (2014). Intuitive NUI for Controlling Virtual Objects Based on Hand Movements. *Future Information Technology, 457-461. Springer.*

Lee, Y. H., Wu, S. K., & Liu, Y. P. (2013). Performance of remote target pointing hand movements in a 3D environment. *Hum. Movement Sci., 32*(3), 511-526.

Lim, C. J., & Jung, Y. G. (2013). A Study on the Usability Testing of Gesture Tracking-Based Natural User Interface. *Posters' Extended Abstracts HCII 2013, pp. 139-143. Springer.*

Löcken, A., Hesselmann, T., Pielot, M., Henze, N., & Boll, S. (2012). User-centred process for the definition of free-hand gestures applied to controlling music playback. *Multimedia systems 18*(1), 15-31.

Loehmann, S., Knobel, M., Lamara, M., & Butz, A. (2013). Culturally Independent Gestures for In-Car Interactions. *Proceedings of Interact 2013, pp. 538-545. Springer.*

Markussen, A., Jakobsen, M. R., & Hornbæk, K. (2013). Selection-Based Mid-Air Text Entry on Large Displays. *Proceedings of Interact 2013, pp. 401-418. Springer.*

Mehler, A., vor der Brück, T., & Lücking, A. (2014). Comparing Hand Gesture Vocabularies for HCI. *Proceedings of HCII 2014, pp. 81-92. Springer.*

Meneses, A., & Hernández, E. (2013). Kinect©, as Interaction Device with a Tiled Display. *Proceedings of HCII 2013, pp. 301-311. Springer.*

Nancel, M., Wagner, J., Pietriga, E., Chapuis, O., & Mackay, W. (2011). Mid-air Pan-and-Zoom on Wall-sized Displays. *Proceedings of CHI 2011, pp. 177-186. ACM Press.*

Panger, G. (2012). Kinect in the Kitchen: Testing Depth Camera Interactions in Practical Home Environments. *Extended Abstracts CHI 2012, pp. 1985-1990. ACM Press.*

Pino, A., Tzemis, E., Ioannou, N., & Kouroupetroglou, G. (2013). Using Kinect for 2D and 3D Pointing Tasks: Performance Evaluation. *Proceedings of HCII 2013, pp. 358-367. Springer.*

Polacek, O., Klíma, M., Sporka, A. J., Zak, P., Hradis, M., Zemcik, P., & Procházka, V. A. (2012). Comparative Study on Distant Free-Hand Pointing. *Proceedings of EuroiTV 2012, pp. 139-142. ACM Press.*

Re, G. M., & Bordegoni, M. (2014). A Natural User Interface for Navigating in Organized 3D Virtual Contents. *Proceedings of VAMR 2014, pp. 93-104. Springer.*

Ren, G., & O'Neill, E. (2013a). 3D selection with freehand gesture. *Comput. Grap.*, *37*(3), 101-120.

Ren, G., & O'Neill, E. (2013b). Freehand Gestural Text Entry for Interactive TV. *Proceedings of EuroiTV 2013, pp. 121-130. ACM Press.*

Rovelo Ruiz, G. A., Vanacken, D., Luyten, K., Abad, F., & Camahort, E. (2014). Multi-Viewer Gesture-Based Interaction for Omni-Directional Video. *Proceedings of CHI 2014, pp. 4077-4086. ACM Press.*

Rümelin, S., Marouane, C., & Butz, A. (2013). Free-hand Pointing for Identification and Interaction with Distant Objects. *Proceedings of AutomotiveUI 2013, pp. 40-47. ACM Press.*

Salvador, R., Romão, T., & Centieiro, P. (2012). A Gesture Interface Game for Energy Consumption Awareness. *Proceedings of ACE 2012, pp. 352-367. Springer.*

Sambrooks, L., & Wilkinson, B. (2013). Comparison of gestural, touch, and mouse interaction with Fitts' law. *Proceedings of OzCHI 2013, pp. 119-122. ACM Press.*

Sousa, T., Cardoso, I., Parracho, J., Dias, P., & Santos, B. S. (2014). DETI-Interact: Interaction with Large Displays in Public Spaces Using the Kinect. *Proceedings of DAPI 2014, pp. 196-206. Springer.*

Tan, J. H., Chao, C., Zawaideh, M., Roberts, A. C., & Kinney, T. B. (2013). Informatics in Radiology: Developing a Touchless User Interface for Intraoperative Image Control during Interventional Radiology Procedures. *Radiographics, 33(2)*, E61-E70.

Tang, J. C., Xiao, R., Hoff, A., Venolia, G., Therien, P., & Roseway, A. (2013). HomeProxy: Exploring a Physical Proxy for Video Communication in the Home. *Proceedings of CHI 2013, pp. 1339-1342. ACM Press.*

Tian, J., Qu, C., Xu, W., & Wang, S. (2013). KinWrite: Handwriting-Based Authentication Using Kinect. *Proceedings of Annual Network & Distributed System Security Symposium 2013.*

Tuntakurn, A., Thongvigitmanee, S. S., Sa-Ing, V., Hasegawa, S., & Makhanov, S. S. (2013). Natural Interactive 3D Medical Image Viewer Based on Finger and Arm Gestures. *Proceedings of BMEiCON 2013. IEEE.*

Vaidyanathan, V., & Rosenberg, D. (2014). “Will Use It, Because I Want to Look Cool” A Comparative Study of Simple Computer Interactions Using Touchscreen and In-Air Hand Gestures. *Proceedings of HCII 2014, pp. 170-181. Springer.*

Van de Camp, F., Schick, A., & Stiefelhagen, R. (2013). How to Click in Mid-Air. *Proceedings of DAPI/HCII 2013, pp. 78-86. Springer.*

Vatavu, R. D. (2013). A comparative study of user-defined handheld vs. freehand gestures for home entertainment environments. *J. of Ambient Intelligence and Smart Env.*, *5*(2), 187–211.

Vatavu, R. D. (2012). User-Defined Gestures for Free-Hand TV Control. *Proceedings of EuroiTV 2012, pp. 45-48. ACM Press.*

Vatavu, R. D., & Zaiţi, I. (2014). Leap Gestures for TV: Insights from an Elicitation Study. *Proceedings of TVX 2014, pp. 131-138. ACM Press.*

Wu, H., & Wang, J. (2012). User-Defined Body Gestures for TV-based Applications. *Proceedings of ICDH 2012, pp. 415-420. IEEE.*

Yim, J., Lee, K., & Kim, H. (2014) Implementation of a Preliminary Natural User Interface for Video on Demand Systems. *Future Information Technology, 475-480. Springer.*

Zarzuela, M. M., Pernas, F. J. D., Calzón, S. M., Ortega, D. G., & Rodríguez, M. A. (2013). Educational Tourism Through a Virtual Reality Platform. *Proceedings of Computer Science 25, pp. 382-388. Elsevier.*