Data input and content exploration in scenarios with restrictions

Diogo de Carvalho Pedrosa

Data input and content exploration in scenarios with restrictions

Diogo de Carvalho Pedrosa

Advisor: **Profa. Dra. Maria da Graça Campos Pimentel**
Co-advisor: **Dr. Pablo Santiago César Garcia**

Doctoral dissertation submitted to the *Instituto de Ciências Matemáticas e de Computação* - ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*.

USP – São Carlos
January 2015

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

P372d

Pedrosa, Diogo de Carvalho
Data input and content exploration in scenarios with restrictions / Diogo de Carvalho Pedrosa; orientadora Maria da Graça Campos Pimentel; co-orientador Pablo Santiago César Garcia. -- São Carlos, 2015.
134 p.

Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2015.

1. Text entry. 2. Eye-typing. 3. Assistive technologies. 4. Interactive tabletop. 5. Interactive digital TV. I. Campos Pimentel, Maria da Graça, orient. II. Santiago César Garcia, Pablo, co-orient. III. Título.

Entrada de dados e exploração de conteúdo em cenários com restrições

Diogo de Carvalho Pedrosa

Orientadora: **Profa. Dra. Maria da Graça Campos Pimentel**
Co-orientador: **Dr. Pablo Santiago César Garcia**

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências - Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA.*

USP – São Carlos
Janeiro de 2015

Acknowledgments

This work would not have been possible without the contributions and the great company of many people, who I acknowledge here. Some of the names will not be remembered now, but may not be less important.

First, I would like to thank my advisor Graça. The results reported here were possible not only because of her research experience and dedication, but also because of all the doors that she opened for me, specially articulating the two international internships that I made. Her pleasure in offering opportunities to the larger number of students possible is something that I want to bring with me. I am also grateful to the detailed and personalized orientation and the great moments spent with my two co-advisors: Pablo, who I had the pleasure to include officially in the records, and Khai, whose name is not in the cover due to the restricted graduate program rules.

I would like to thank all the colleagues from the Intermídia lab, the SEN5 group, and the HCILab that shared the their working spaces with me. From the Intermídia: Didier, José Augusto, Diogo Martins, Bruna, Olibário, Raíza, Raoni, Omar, Rodolfo, Lílian, Cássio, Alan, Cristiane, Marcos, Renata, David, Eduardo, William, Johana, Rudinei, Tiago, Jorge, Danilo, Dilvan, João Paulo, Kléberson, Adriano, Flor, Silvio, Manzato, Edson, Sadao, Douglas, Kifayat, and many others. From the SEN5 group: Rodrigo, Jack, Bo, Chen, Fons, Kees, Marwin, Rufael, Simon, Dick, Ivan, and Steven. From the HCILab: Mingming, Alex, Michael, Vikash, Berto, Matt, Jun, Erik, Kaz, Stephanie, Lilla, Bill, Celine, Heather, Mary Lou, and David. Thank also to my colleagues from the Lince lab—César, Erick, Caio, Kamila, and Tiago—, Amy Wrigth, a partner in my last study, and Pedro, for creating the illustration of the scenarios.

Thank to all the participants of the many experiments that I conducted and the respondents of the many forms that I applied. Thank to CNPq, RNP, CAPES, CWI, and FAPESP (grant #2012/01510-0 and #2013/04306-7) for the financial support.

As I am not from São Carlos, I had the pleasure to create a big and incredible new family during the last almost six years. A very special thank to Alê, Flávio, Claudinha, Van, David, Aline, Deca, Guga, João, Antônio, Sarinha, Sadao, Vini, Lígia Nunes, Lu, Lígia Paschoal, Fazoca, Marcelo, Vívian, Oripes, Ju, Nat, and Felipe. My jogging group, forcing me to leave the lab almost every night, are also great! Thank to my coach Vitor all the others!

A very special thank to my dear distant friends Márcio, Suzana, Tarcísio and Ritinha, who I

could meet from time to time in some of the travels that I made these years.

Of course, I could not forget the great support of my family. My father Ivo, my mother Bel, my sister Andrea and my brother Marcelo, my brother in law Wolfgang and my nephew Arthurzinho are truly responsible for who I am today. Thank also to my cousins, uncles, aunts and people who also became part of my family: Lara, Paulo, Bia, Victor, Marcus, Stê, Airtinho, Inaê, Tiê, Moema, Leda, and Airton.

Lastly, I would like to specially thank the most important person in this journey, the first to hear all my complains, who give me happiness, peace and security, who is gestating my little Amora, my love, Uaiana.

Abstract

As technology evolves, new devices and interaction techniques are developed. These transformations create several challenges in terms of usability and user experience. Our research faces some challenges for data input or content exploration in scenarios with restrictions. It is not our intention to investigate all possible scenarios, but we deeply explore a broad range of devices and restrictions. We start with a discussion about the use of an interactive coffee table for exploration of personal photos and videos, also considering a TV set as an additional screen. In a second scenario, we present an architecture that offers to interactive digital TV (iDTV) applications the possibility of receiving multimodal data from multiple devices. Our third scenario concentrates on supporting text input for iDTV applications using a remote control, and presents an interface model based on multiple input modes as a solution. In the last two scenarios, we continued investigating better ways to provide text entry; however, our restriction becomes not using the hands, which is the kind of challenge faced by severely motor-disabled individuals. First, we present a text entry method based on two input symbols and an interaction technique based on detecting internal and external heel rotations using an accelerometer, for those who keep at least a partial movement of a leg and a foot. In the following scenario, only the eyes are required. We present an eye-typing technique that recognizes the intended word by weighting length and frequency of all possible words formed by filtering extra letters from the sequence of letters gazed by the user. The exploration of each scenario in depth was important to achieve the relevant results and contributions. On the other hand, the wide scope of this dissertation allowed the student to learn about several technologies and techniques.

Keywords: Text entry, eye-typing, assistive technologies, motor disabilities, interactive tabletop, additional screen, multimodal data, interactive digital TV

Resumo

Com a evolução da tecnologia, novos dispositivos e técnicas de interação são desenvolvidas. Essas transformações criam desafios em termos de usabilidade e experiência do usuário. Essa pesquisa enfrenta alguns desafios para a entrada de dados e exploração de conteúdo em cenários com restrições. Não foi intenção da pesquisa investigar todos os possíveis cenários, mas sim a exploração em profundidade de uma ampla gama de dispositivos e restrições. Ao todo cinco cenários são investigados. Primeiramente é apresentada uma discussão sobre o uso de uma mesa de centro interativa para a exploração de fotos e vídeos pessoais, a qual também considera um aparelho de TV como tela adicional. Com base no segundo cenário, uma arquitetura que oferece a aplicações de TV digital interativa (TVDI) a possibilidade de receber dados multimodais de múltiplos dispositivos é apresentada. O terceiro cenário se concentra no suporte a entrada de texto para aplicações de TVDI usando o controle remoto, resultando na apresentação de um modelo de interface baseado em múltiplos modos de entrada como solução. Os dois últimos cenários permitem continuar a investigação por melhores formas de entrada de texto, porém, a restrição se torna a impossibilidade de usar as mãos, um dos desafios enfrentados por indivíduos com deficiência motora severa. No primeiro deles, são apresentados um método de entrada de texto baseado em dois símbolos de entrada e uma técnica de interação baseada na detecção de rotações do pé apoiado sobre o calcanhar usando acelerômetro, para aqueles que mantêm pelo menos um movimento parcial de uma perna e um pé. No senário seguinte, apenas os movimentos dos olhos são exigidos. Foi apresentada uma técnica de escrita com o olho que reconhece a palavra desejada ponderando o comprimento de a frequência de ocorrência de todas as palavras que podem ser formadas filtrando letras excedentes da lista de letras olhadas pelo usuário. A exploração de cada cenário em profundidade foi importante para a obtenção de resultados e contribuições relevantes. Por outro lado, o amplo escopo da dissertação permitiu ao estudante o aprendizado de diversas técnicas e tecnologias.

Palavras-chave: Entrada de texto, digitação com olhar, tecnologias assistivas, deficiências motoras, mesa interativa, tela adicional, dados multimodais, TV digital interativa.

Contents

List of Tables

List of Figures

List of Listings

List of Algorithms

Chapter 1

Introduction

As devices for data entry evolve, user-interaction alternatives present both challenges and opportunities for designers of interactive applications. In the last years, interactive TVs have reached a stable maturity level, smartphones and tablets have become popular, and interactive tabletops and eye-trackers have become affordable. These transformations create several challenges in terms of usability and user experience. Our research faces some of them.

Each device type offers a different way for people to interact with digital data. Sometimes, the same information can be accessed using different device types, as it is the case of web pages rendered by web browsers. However, the form factor and input and output mechanisms of the device create advantages and disadvantages for their use in different contexts. As an example, research has demonstrated that direct manipulation of multi-touch interfaces fosters collaboration, as it allows adjacent users to easily understand the actions of others [Buisine et al., 2012; Soro et al., 2011]. If a group of people wants to engage in a collaborative task of creating a presentation from a set of photos and videos, an interactive tabletop seams appropriate. However, they would be restricted to perform this activity in the room where the heavy device is installed. If they decide to do this in a place where no tabletop is available, they might need to use a laptop and overcome the personal-oriented nature of it.

Even if the most appropriate device is used in a given context, the accomplishment of a complementary task may become a challenge. This is because the nature of the primary task, which determines the device in use, may be different form the nature of the secondary task. Continuing with the example of collaboratively creating a presentation, individuals may need to explore the available content by themselves before suggesting an specific piece to the group. This secondary task would be better supported by a laptop. However, as the primary task is to collaboratively create a presentation, sometimes people will have to face the challenge of also using the tabletop for individually exploring content. Designers must not neglect the need to provide personal areas [Apted et al., 2006], personal controls and screen orientation [Shen et al., 2003] in a tabletop. The group as a whole may also need from time to time to watch individual videos or to review the current version of the presentation. For that, a vertical screen would bring

more comfort.

As another example, let us consider a television show in which guests are engaged in a quiz while the remote audience is invited to play along [Luyten et al., 2006]. Even though it is hard for the viewer to use a regular remote control to give textual answers to open questions, the use of a TV set would still be the best option for the primary task of watching the show. Text input using remote control is one of the restricted scenarios discussed in this dissertation.

Another type of restriction emerges from the diversity of people. According to the Convention on the Rights of Persons with Disabilities [United Nations, 2006], products, environments, programmes and services should be designed "*to be usable by all people, to the greatest extent possible, without the need for adaptation or specialized design.*" In this definition of "universal design", the convention also acknowledge that some particular groups of persons with disabilities may also depend assistive devices, where needed. The challenges faced by those groups are numerous. Individuals with severe motor disabilities, for instance, depend on assistive technologies that replace the regular keyboard to be able to type.

In this dissertation, we deal with challenges that arise when interactive tabletops, interactive television sets, mobile devices, and personal computers are used for data input or content exploration in scenarios with restrictions. For each scenario, we present the methodology used and discuss a possible solution.

1.1 Scenarios with restrictions

In our **first scenario**, which we call **Table**, we are interested in the exploration of home media in family gatherings. Nowadays, laptops and tablets (together with televisions) are used, but they impose restrictions on the level of social interaction they allow. The collaborative nature of passing photos around and the possibility of keeping a photo in your hands for enjoying its details may be recovered by interactive tabletops. Their large size and high resolution make them a good alternative for enabling, as a coffee table in the living room, old practices around media.

In order to help us understanding the requirements demanded by a media sharing application, we applied an informal questionnaire with potential users searching for desired functionalities. The responses to the questionnaire guided the implementation of a prototype application that allowed us to test different concepts with real users. In the experiments, we aimed at better understanding three main challenges for effectively supporting social interactions around media in the living room:

1. Metaphor: is it desirable to have a digital interface that uses the metaphor of physical photographs placed on the table?
2. Digital ecosystem: how helpful/disturbing is to use additional screens (e.g., TV) for media exploration?
3. Level of control: what is the adequate level of control that should be provided to the users?

The use of the metaphor of physical photos was considered fun and intuitive. However, the

need of some software support regarding the alignment and distribution of media items was explicitly indicated by some of the participants. The desired level of control turned out to be dependent on the task at hand. Storytelling scenarios and explorative scenarios have different requirements. Most of the participants preferred to use the tabletop when they could count on an auxiliary display. The use of a TV helps creating an environment that supports several users and improves the experience by providing a better image quality and comfort.

A television set is useful not only as an output device for interactions occurring in a tabletop but also as a larger screen for tablets and smartphones. In fact, this potentiality is explored by Chromecast[1], a small media streaming device (dongle) release in mid-2013. It is plugged into the HDMI port of a TV, allowing smartphones, tablets and computers to stream content to the big screen. Another way of leveraging device functionalities by interconnecting them is to use portable devices as input devices for interactive TV applications. In our **second scenario**, called **Multimodal**, we propose a component architecture that allows richer applications for interactive digital television (iDTV) to be developed, offering them the possibility of receiving multimodal data from different devices. Applications for iDTV offer few features because they must be easy to use, even if used only with the remote control as an input device. The restriction on the remote control—compared with a full keyboard and a direct manipulation device such as a mouse, that are common on personal computers—limits the usability of iDTV applications [Carmichael et al., 2006] and the interactive potential of an iDTV system [Roibás et al., 2005]. Cortez et al. [2012] say that the limitation of TV's input modality "*is a burden on anyone trying to build rich cinematic Internet experiences to the TV*".

For this scenario, after motivating the reader, we present an architecture that offers to applications running on a set-top box (STB) the possibility of receiving multimodal data (audio, video, image, ink, accelerometer data, text, voice and customized data) from multiple devices (such as smartphones, tablet, laptops or even desktops). To validate the architecture, we implemented a corresponding multimodal interaction component which extends a Digital TV middleware and we built applications that use the component.

The purpose of one of the applications is to provide communication between users, allowing exchange of text messages and file sharing. Our **third scenario**, called **Remote**, also deals with the challenge of entering text in an interactive TV application. However, instead of dealing with different devices, we focus on the input through a remote control. We present a model of a software interface based on multiple input modes: a virtual keyboard mode, a phone keypad mode, and a speech mode.

We designed the component according to the user-centered design (UCD) methodology. We first carried out a literature review with respect to text input methods used in TV systems. Next, we interviewed four experts in the iDTV domain. A third activity involved the application of questionnaires to 153 TV users, aiming at identifying a profile that corresponds to users who use text entry mechanisms. After developing a first version of the prototype, we conducted usability

[1] `http://www.google.com/chrome/devices/chromecast`, accessed 30/Sep/2014.

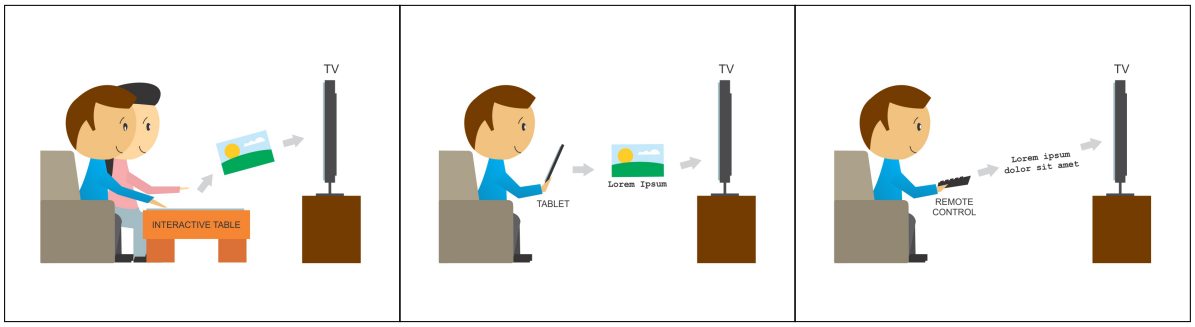
tests using the think aloud protocol, and usability inspections using the heuristic evaluation and cognitive walkthrough techniques. The evaluations allowed the detection of a number of problems, which could be dealt with in intermediary versions. The evaluations also pointed to several opportunities of design improvement. In particular, they highlight the importance of using complementary text input modes in order to satisfy the needs of different users.

Be it for TV applications or personal computers, text entry is an important skill in our lives. People who have been affected by a motor neuron disease or any disorder that affects voluntary muscle activity may be deprived of this routine means of communication. Examples of diseases are Amyotrophic Lateral Sclerosis (ALS) and Duchenne Muscular Dystrophy (DMD). The motivation of our last two scenarios is to provide text entry interaction techniques for those with severe motor disabilities.

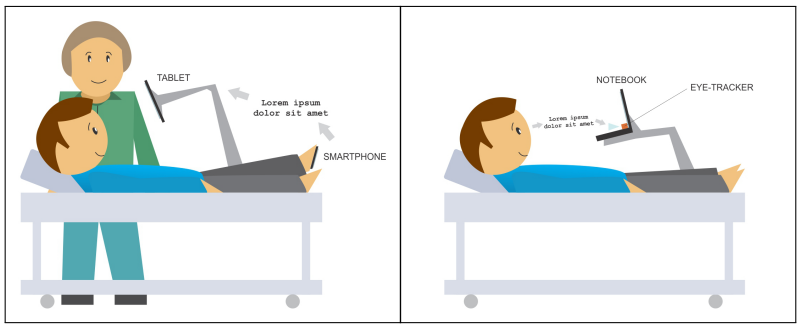
In the **fourth scenario**, which we call **Foot**, we present an interaction technique and a text entry method designed for people with a severe motor disability who, for some time, keep partial movement of a leg and a foot. The idea is to attach in one of the user's feet an accelerometer-equipped device, which detects and transmits the movements to a second device located in front of the user's face. The method interprets the movements as characters according to a Morse-based codification. Our design is informed by the motor capabilities of a male, in his 60s, with a motor neuron disease. Our first contact with him and his family was used to understand his needs and to specify the requirements for a prototype. In a second meeting, we were able to test a working prototype with him and understand the adjustments that had to be done. A second version of the prototype were developed and tested with 15 able-bodied users in order to establish a baseline that allows comparisons with other methods and to easily detect further potential improvements.

The test results were not completely satisfactory. We realized that leveraging foot movement was not any better than exploring the potential of extraocular muscles, a strong and efficient set of six muscles that control the movement of the eye. They remain unaffected until late stages of ALS. Thus, in the **last scenario**, called **Eye**, we explore typing with the movements of the eye. In eye-typing, a virtual keyboard is shown on the screen and the user serially gazes at the intended keys. We propose and evaluate an eye-typing technique that is based on filtering out letters from the sequence of letters looked by the user. Experiments with able-bodied participants reveal that our key filtering-based approach, Filteryedping, is a fast technique, allowing an average of 15.95 words per minute after 100 minutes of typing. An iterative design and evaluation with individuals with severe motor disabilities helped us to improve the technique by creating parameters that allow it to be adapted to different users.

Figure 1.1 illustrates a simplified version of each scenario. We present each illustration again in the beginning of the chapter that discuss its respective scenario.



(a) Table (b) Multimodal (c) Remote



(d) Foot (e) Eye

Figure 1.1: Illustration of the scenarios.

1.2 Contributions and organization of the dissertation

It was not our intention to explore all possible scenarios in which data input or content exploration happens with some kind of restriction. However, as discussed in the previous section, we have explored a broad range of devices and restrictions. The main contributions of this dissertation are:

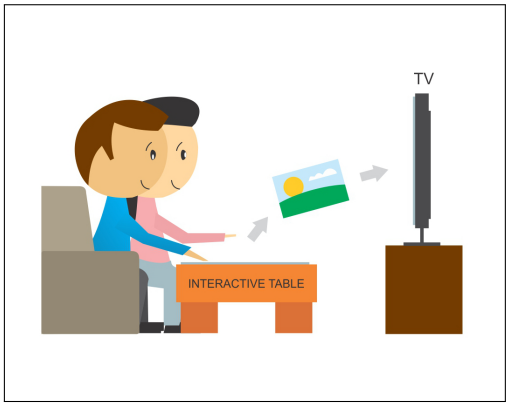
- **Table Scenario** – A discussion about three research questions relative to the use of an interactive coffee table for exploration of personal photos and videos, supported by the vignettes and quotes from couples that used the tabletop and vertical screen set-up (Chapter 2, based on [Pedrosa et al., 2013]);
- **Multimodal Scenario** – An architecture for a component that offers to applications running on a set-top box the possibility of receiving multimodal data from multiple devices (Chapter 3, based on [Pedrosa et al., 2010a; 2011]);
- **Remote Scenario** – An interface model based on multiple input modes to deal with limitations of text input in iDTV using a remote control (Chapter 4, based on [Pedrosa et al., 2010b; Vega-Oliveros et al., 2010]);
- **Foot Scenario** – A text entry method, based on two input symbols, and an interaction technique, based on detecting internal and external heel rotations using an accelerometer, for people with a severe motor disability (Chapter 5, based on [Pedrosa and Pimentel, 2014]);
- **Eye Scenario** – An eye-typing technique that recognizes the intended word by weighting

length and frequency of all possible words, formed by filtering extra letters from the sequence of letters gazed by the user (Chapter 6, based on [Pedrosa et al., 2014; 2015]).

Chapter 7 discusses the contributions and limitations of the dissertation, pointing out directions for future research.

Chapter 2

Exploration of personal photos and videos using an interactive coffee table



Interactive tabletops offer a unique opportunity for exploring home videos and photos. Nevertheless, there are still a number of unexplored challenges for effectively providing support for collocated group interaction around media. This chapter[1] reports on a user study involving 24 users, intended to better understanding the challenges ahead. Our volunteers—in couples—evaluated our media sharing application prototype, providing valuable feedback with regards to three key challenges: metaphor, digital ecosystem, and level of control. First, users appreciated the selected metaphor of physical photos, but without relinquishing software support, such as alignment and distribution of media items. Second, vertical auxiliary screens helped in supporting a bigger number of users and providing more comfort and a better viewing angle and stance. Third, the nature of the task (either storytelling or random exploration) had a strong influence on the control capabilities to be provided by the application. Fourth, personal spaces within the tabletop were useful for allowing independent navigation. We consider these results as relevant for the future developments of home media sharing applications for the living room, as they guide designers to overcome restrictions related

[1]This chapter is based on the following paper:

- **D. Pedrosa**, R. L. Guimarães, M. G. C. Pimentel, D. C. A. Bulterman, and P. Cesar. Interactive coffee table for exploration of personal photos and videos. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, pages 967-974, New York, NY, USA, 2013. ACM. doi: 10.1145/2480362.2480548. URL http://doi.acm.org/10.1145/2480362.2480548.

to the collaborative exploration of personal media using tabletops.

2.1 Research questions on media exploration in family gatherings

Families and friends used to gather in the living room to watch printed photographs. Nowadays, laptops and tablets (together with televisions) are used for the same purpose, but they impose restrictions on the level of social interaction they allow. The collaborative nature of passing photos around and the possibility of keeping a photo in your hands for enjoying its details are somewhat lost.

Current advances in technology and cost effective manufacturing processes are making interactive tabletops affordable, filtering them into everyday use. Its large size and high resolution make them a good alternative for enabling, in the living room, old practices around media. Research has demonstrated that direct manipulation of multi-touch interfaces fosters collaboration, as it allows adjacent users to easily understand the actions of others [Buisine et al., 2012; Soro et al., 2011]. One effort towards creating an interactive coffee table prototype is presented by Grammenos et al. [2010].

In our research, we are interested in better understanding how interactive tabletops can be used for exploring and viewing home media in family gatherings. With this goal in mind, we place an interactive tabletop in the middle of the living room, so that it can be used as a normal coffee table in front of the television set (see Figure 2.1). We consider this location as a stimulus for group engagement, as it is a place typically used during leisure moments, and where old shoeboxes of photos were usually placed. In particular, we aim at better understanding what are the challenges ahead for effectively supporting social interactions around media in the living room. For that, we conducted a number of experiments considering three main challenges:

1. Metaphor: is it desirable to have a digital interface that uses the metaphor of physical photographs placed on the table?
2. Digital ecosystem: how helpful/disturbing is to use additional screens (e.g., TV) for media exploration?
3. Level of control: what is the adequate level of control that should be provided to the users?

The main contribution of this chapter is the discussion about these three research questions, supported by the vignettes and quotes from the couples using the tabletop and vertical screen set-up.

Figure 2.1: Interactive coffee table for exploration of personal photos and videos.

2.2 Related work on social interactions around media in the living room

We present works relevant to the issues we investigate. Although they share some of our concerns, they do not provide a comprehensive discussion about the challenges of supporting social interactions around media in the living room, as we do.

2.2.1 The metaphor of physical photographs

The process of looking photos may be structured in grids, stacks or lists. It can also be unstructured, as when several photos are kept together in a container, usually a shoebox, and can be spread over a table or handled individually and passed from hand to hand. Kirk et al. [2012] explored the shoebox metaphor in a media management application for an interactive tabletop, which offers a space in which virtual shoebox of photos can be stored and another space where the photos can be manipulated. As in our application, media objects are under physical-based constraints, providing users with an environment that is closer to the real-world dynamics. They also offer a way of arranging photos in a grid in order to allow a quick overview. The authors were not concerned about providing a wide range of functionalities, but to analyze how families interact with tabletops in a domestic setting.

Apted et al. [2006] describes the design of a collaborative digital photograph exploration application for a tabletop. Its design is strongly influenced by the metaphor of physical photographs placed on the table in order to achieve an easy to learn and easy to remember interface,

mainly for elderly. They offer several photo editing functionalities, but no storytelling support was given.

Chen et al. [2010] argue that some dynamic and unstructured activities with media collections, such as casual browsing and searching or storytelling, are poorly supported by today's interfaces. They present a prototype that uses a magnet metaphor in addition to a tree-view in order to support a flexible combination of structured and unstructured photo browsing and searching. Unlike ours, their prototype was designed for a desktop controlled by a mouse.

Hilliges and Kirk [2009] show how easily users of their tabletop photo-sharing application get sidetracked during a photo-talk, and call the readers' attention to the implications of this to the design of photo exploration applications. As we are going to show, the prototype we have developed have 3 different modes of exploration, which offer both the possibility to follow a structured presentation sequence or to get inspired by a "messy" arrangement of "physical" media items.

Kristensson et al. [2008] also report on a design for browsing of photo collections using a tabletop, however they were concerned with the bi-manual manipulation of the collections' tag cloud.

2.2.2 Additional screen

Although interactive tabletops usually offer a large screen size, the use of a vertical additional screen is often explored. Wigdor et al. [2006] identify various design requirements for the implementation of a system intended to support the integration of an interactive tabletop into an environment with multiple screens. Yoshimoto et al. [2011] use the combination of an interactive tabletop and a TV screen to show different views while navigating maps. It shows the Google Maps interface in the tabletop while the additional screen shows Google Earth and Google Street View. Those two works are not focused on a domestic use. Chiu et al. [2008] present, without further discussion, a video demo of tabletop system for browsing photo and video collections and use a flat panel display in the room to show the full-size items.

To help users benefit from a multi-device environment, researchers investigate ways to inform people in a room of which devices can be connected, what content can be transfered, and how it can be done. Marquardt et al. [2012] present a design pattern called gradual engagement, which is based on three stages: 1) awareness of device presence, 2) reveal of exchangeable content, and 3) interaction methods for transferring content. Fei et al. [2013] introduce some interaction techniques that use an array of NCF tags on the edges of a tabletop. That way, the spacial relationship between people and surfaces help users understand how to interact. In our design, the relative position between the coffee table and an additional TV screen is also considered in the interaction.

2.2.3 Shared controls

Seifried et al. [2009] present a system which interface is shown on an interactive coffee table and acts as a universal control of digital devices. Although the authors argue that it "*encourages social interaction with friends and family*", the initial evaluation did not focus on group tasks. The interface of the system uses a video image of the environment, captured by a camera located on the ceiling. When used by multiple users, the whole tabletop surface acts as a single control unit. In our work, we investigated a scenario which provides multiple controls to avoid bottlenecks. Their system can be used to start the playback of a movie by dragging a DVD from the shelf to the TV (in the interface). In our system, in order to watch one of the personal movies on the TV, the user drags it in the direction of the edge closest to the TV, as if the video would go out of the table and reach the TV.

The application presented by Shen et al. [2003] describe an interactive tabletop system for image visualization that allows addition, removal and displacement of control panels, so that each user can have its own control panel in front of herself. However, no detail is given regarding these functionalities in the brief description presented about the conducted user tests.

People usually partition tabletop interfaces into personal, group and storage spaces [Scott et al., 2004]. The work of Apted et al. [2006] also includes the personal space concept. In their application, when a photo is in a user's personal space, no other user may select it. Thus, it imposes a stronger constraint to the interaction, when compared to the personal spaces of our application.

Klinkhammer et al. [2011] implement a tabletop-integrated tracking system that is capable of detecting users' location for assigning a display space to each user. The interface allows visitors of a museum to explore information in a self-directed way. However, the system was designed only for parallel information exploration, which means that its main purpose is to divide a big table by smaller interaction areas. Group tasks were left to future work.

In the study of Sugimoto et al. [2004], a system that allows users to collaborate in various tasks is presented. A board is used as a shared space and PDAs are used as personal spaces. The two spaces are linked, allowing work results to be moved from one to the other. They were not concerned with offering personal and shared spaces in the same device.

Möllers and Borchers [2011] investigate shared and personal spaces in the perspective of privacy. Their main concern was to create tangible widgets that can be placed over interactive tabletops to provide privacy control over part of the contents displayed. This study is not concerned with the privacy aspect of personal spaces.

2.3 Requirements elicitation

In order to help us understanding the requirements demanded by our media sharing application, we applied an informal questionnaire with potential users searching for desired functionalities. It

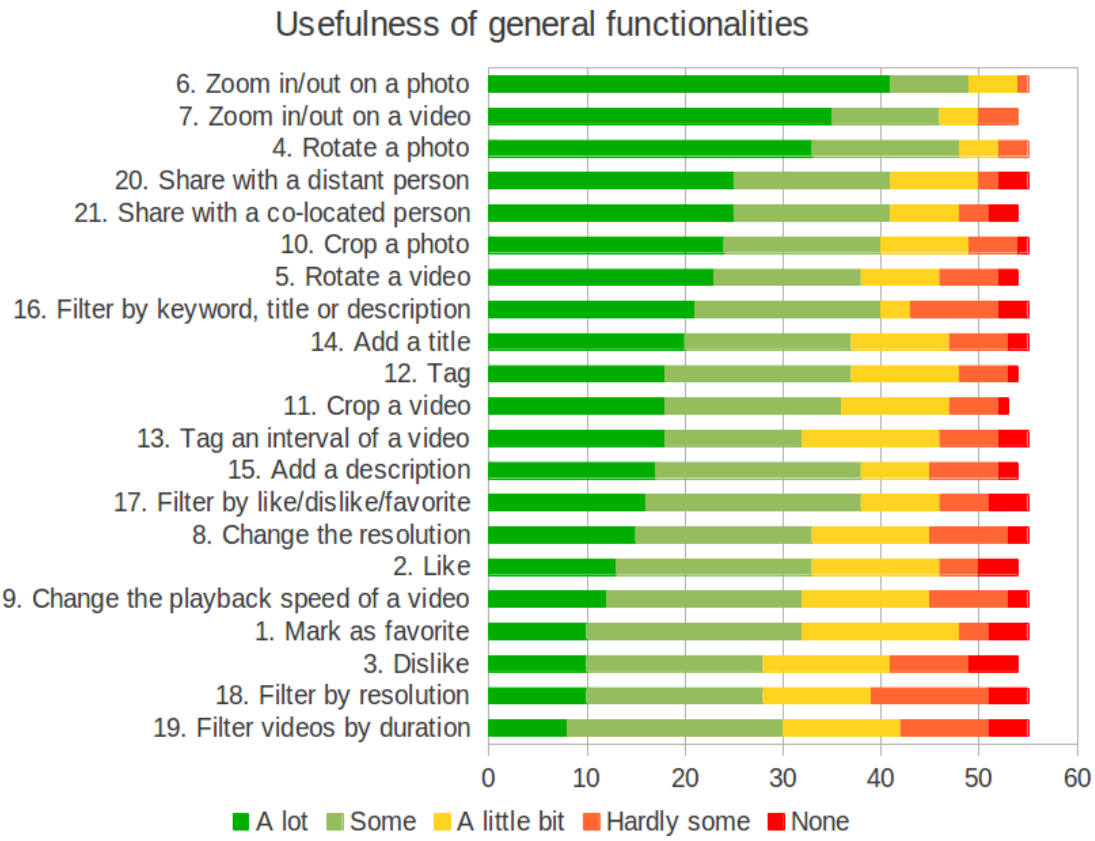


Figure 2.2: Questionnaire for requirement elicitation: Usefulness of general functionalities.

was administered through a web system and it took each of the 55 respondents about 15 minutes to complete. There were 40 males and 15 females. There were 43 people aged between 20 and 49 and 12 people aged between 50 and 69. The majority (36) never used an interactive tabletop before. Other 16 respondents declared to rarely use it. Only one declared to use it sometimes. Two people did not answer this question.

A section of the questionnaire asked what general functionalities are considered the most useful. 21 functionalities were rated by the respondents in a scale ranging from none to a lot. Figure 2.2 shows the aggregated answers sorted by relevance.

Another section asked in particular about collaborative functionalities. Figure 2.3 shows ten rated functionalities sorted by relevance. The responses indicate that users are interested in having different presentation modes and being able to reach distant areas of the tabletop surface, either giving or bringing back an item to someone seated in the other side of the tabletop. The term “Implicit division” (question 2 in Figure 2.3) refers to the use of a different background color to separate the table area between users, allowing photos and videos to move from one area to another. An interesting suggestion given in the open questions was the use of a physical support that allows people to switch the device between a vertical and horizontal position. We also asked “How useful it is to have a TV near the interactive table as a vertical auxiliary screen?” Answers –None (0), Hardly some (3), A little bit (7), Some (14), A lot (23)– show that the TV as an auxiliary device brings high expectations.

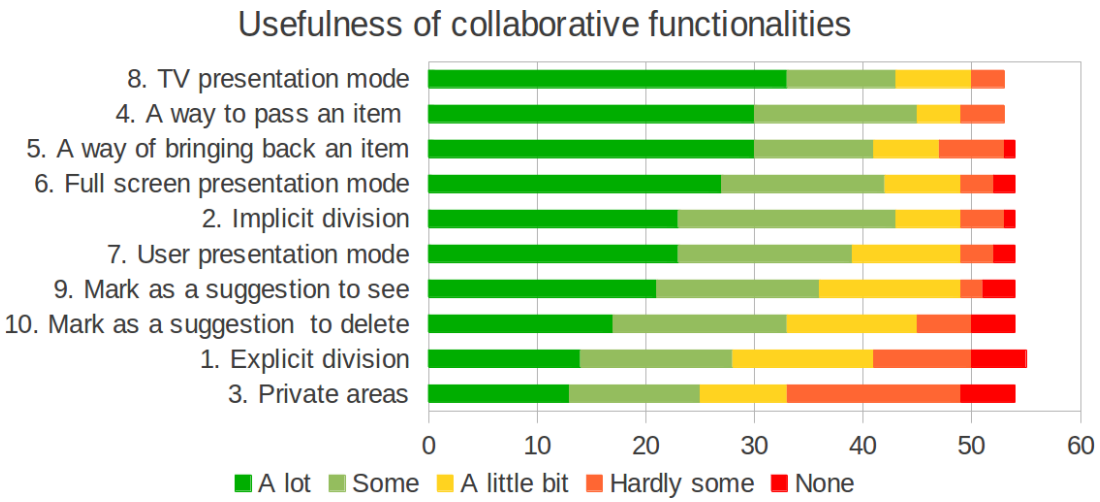


Figure 2.3: Questionnaire for requirement elicitation: Usefulness of collaborative functionalities to explore photos and videos in an interactive tabletop.

2.4 The prototype application

The responses to the questionnaire guided the implementation of a prototype application that allowed us to test different concepts with real users. In particular, the top three general functionalities and the top five collaborative functionalities were included, which lead to three modes of use described in the following subsections. We implemented the tabletop application in C++, using the Cornerstone Software Development Kit (SDK)[2].

2.4.1 Free Exploration Mode

The interface shown after the initialization of the application uses the metaphor of physical photographs placed on the table. In this mode, media items can be moved, resized and rotated by means of touch gestures. A move may be performed using a single finger, while resize and rotate are performed with two fingers.

A single tap can be used to bring an item to the front and to start or to pause the playback of videos. Media items respect the principle of inertia in terms of slowing after being thrown in any direction, and of bouncing when reaching a border.

The free exploration mode also shows a personal space for each user, indicated by a different (lighter) background color. Figure 2.4 (top-left) shows the interface supporting two users seated at the same side of the tabletop.

2.4.2 Presentation Mode

For the whole surface to show a single media item, the prototype offers a presentation mode (Figure 2.4 (bottom-left)), which can be activated in two ways: 1) increasing the size of a media

[2] `https://cornerstone.multitouch.fi`, accessed 25/Sep/2014.

Figure 2.4: Exploration mode (top-left), presentation mode (bottom-left), and TV mode (right) interfaces.

item until its height reaches 60% of the height of the display; or 2) tapping a media item using two or more fingers at the same time. In addition to the image in an increased size, this mode also presents a thumbnail bar with all media items in the top of the screen. The bar can be scrolled if there are more items being explored than what can be shown. Tapping a thumbnail replaces the current selected media item. Another way of navigating the collection is making a swipe gesture over the active item to the left or to the right. A stop button allows the user to go back to the free exploration mode.

2.4.3 TV Mode

The interfaces of the application are all oriented towards one of the long sides of the table. One can criticize this design decision by arguing that users in the other side may see the upside-down media items. This is what motivates Shen et al. [2003] to use a circular design, providing a continuous desktop display orientation for multiple people. However, our interfaces were designed for a tabletop used as a coffee table, located in the center of the living room, where usually there is a couch set facing a television display and users sat at one of the sides of the tabletop (in front of the TV).

This arrangement is thought to integrate the tabletop with the TV set. The TV mode (Figure 2.4 (right)) allows users in the couch to see photos and videos in a proper angle of view. In order to activate this mode, users simply flick the media item in the direction of the edge closest to

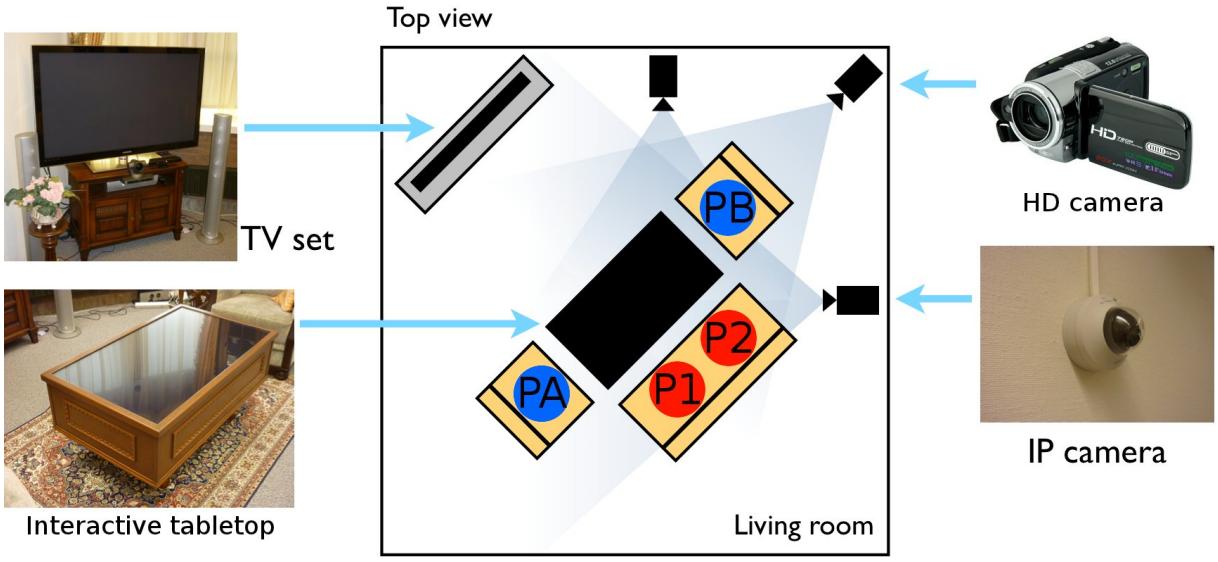


Figure 2.5: Sketch of the room setup and participants (P_i) layout.

the TV. This hides the media item from the tabletop and shows it on the TV display. It also shows a thumbnail bar—as the one shown in the presentation mode, but this time in the bottom part of the tabletop surface—and two control panels that allow the navigation between media items and the playback control for the videos. The stop button of the control panel exits the TV mode and brings the media item back to table. Another way of bringing an item back to the tabletop is sending another item to the TV. The communication between the tabletop and the TV is done through a server with WebSocket technology, using the Pusher service[3]. The TV shows a browser in full screen mode.

2.5 User experiments

We were able to conduct user experiments for answering our three research questions. The experiments were conducted in a lab that simulates a living room. The lab has an interactive tabletop (MT467 MultiTouch Cell 46" Advanced Full HD LCD), a LED 55" television, three video cameras, a three-seat couch and two armchairs (Figure 2.5).

The 24 volunteers were grouped in pairs, resulting in 12 experiment sessions. All pairs knew each other well, except one. Each session lasted about 45 minutes and was divided in five parts. In the first four parts, users had the opportunity to use the application while discussing some issues with the moderator(s). In the last part, they answered a post-test questionnaire. About half of the participants used their own media during the experiments. The set of photos and videos used in the experiments was such that it favored engagement of both users in a collaborative exploration (in 5 pairs, each participant knew the used media as much as his/her partner) or the presentation of media items (storytelling from the person who was in the event to the one who

[3] `http://http://pusher.com`, accessed 25/Sep/2014.

Table 2.1: Characterization of participants. The number of answers do not totalize 24 because two of the volunteers did not answer the post-test questionnaire (it was not mandatory).

Gender distribution	
Males	14 (58%)
Females	8 (33%)

Age distribution	
13-19 years	1 (4%)
20-49 years	11 (46%)
50-69 years	9 (37%)
$\geq$ *70 years*	1 (4%)

Device frequency of use			
	Smartphone	*Tablet*	*Interactive table*
Never	4 (17%)	6 (25%)	14 (58%)
Rarely	1 (4%)	3 (12%)	7 (32%)
Sometimes	3 (12%)	5 (21%)	1 (4%)
Often	3 (12%)	5 (21%)	0 (0%)
Always	10 (42%)	3 (12%)	0 (0%)
Total	21 (87%)	22 (92%)	22 (92%)

did not know the media items – 7 pairs). In all cases, the number of items in the set was around 25. Table 2.1 presents a short characterization of the participants.

In the first three parts of each experiment, users were seated side by side in front of the TV (Figure 2.5, participants P1 and P2). The three first parts lasted around 20 minutes altogether. The first part was used to let participants familiarize themselves with the sensitivity of the tabletop screen using the free exploration mode. They were stimulated to move, resize and rotate the media items, while exploring the media collection. The second part was used to present and inquire users about the presentation mode. The moderator explained and demonstrated the two ways of accessing it (zooming or tapping with two or more fingers). This was followed by a discussion about relevant topics related to the presentation mode, such as the ways of accessing it, the position and functionalities of the thumbnail bar and how to navigate the collection. Then, the experiment entered in the third part, in which the TV mode was introduced. At this point, the moderator turned on the TV and explained how to send items back and forth to and the TV. Again, participants had the opportunity to play around with the functionality just presented, and they discussed with the moderator issues, such as the use of multiple control panels, the more comfortable posture propitiated by the vertical screen, and the drawbacks of having to split the attention between two screens.

The last hands-on section was preceded by the participants being asked to move to the armchairs parallel to the TV (Figure 2.5, participants PA and PB). At this point, we disabled the presentation and TV modes, and the application instructed each user how to divide the media items in two groups. The objective was to let users face the challenge of reaching distant items or having to place items in a distant area of the tabletop. They were asked to move what they considered the “best items” to one of the sides and to move the “worst items” to the other. In some cases, participants suggested and used different criteria to split the media items. This way, both users had to collaborate by giving items to each other, and asking for items. This part lasted about 5 minutes.

Figure 2.6: Participants using the free exploration mode for the last hands-on section.

Figure 2.6 shows the interface of the application during this part of the experiment. It is worth noting that the personal spaces were now placed at the small sides of the tabletop: in this scenario, as a media item enters a personal space, it is automatically rotated towards the user and, as it leaves a personal space, it is automatically rotated towards the couch. We used this as an opportunity to discuss the role of personal and shared spaces in a tabletop application. In this part of the experiment we also discussed some alternative techniques to facilitate the access of different areas of the surface and how collaboration between users can complement the techniques.

2.6 User habits

The videos recorded during the experiments and the answers to the post-test questionnaire were analyzed. Each recording was carefully watched and notes were taken. The grouping of related notes helped us to analyze the results as reported in this and the following section. In this section, we present the current habits regarding media consumption, as we believe they are useful for better understanding the results, and we discuss the importance of offering support for both storytelling and random exploration tasks.

2.6.1 Media consumption

As expected, current habits related to the consumption of personal media are diverse. Most people see and show photos and videos in their personal computer. Only some of them connect it to the TV (either by a USB cable or wireless) in order to benefit from advantages, such as screen

size, quality and comfort. It is still possible to find people who keep the habit of showing albums or boxes of physical photos, but looks like it is increasingly restricted to old photos. One of the volunteers said nostalgically:

> "Ten years ago, it was nice to sit around the table, with friends and to look at photos from holidays [...] But at the moment we don't do it. [...] This [application] is more familiar." (P10)

Some people still keep the habit of printing photos, either to give them as a gift, or to use in picture frames or photo boards. Swan and Taylor [2008] present a detailed examination of these habits using examples from a study undertaken with six families. On the other hand, we also found among the volunteers people who now use their mobile device, such as smartphones and tablets, by passing it hand-to-hand, or by connecting it to the TV, what is consistent with practices reported in other studies [Lucero et al., 2011; Stelmaszewska et al., 2008].

Media items are usually organized in folders, separated by events. The exploration is usually restricted to a specific event, which can also be a holiday of several days. The most common order used to explore event items is the chronological. Some volunteers also search photos by person, by GPS location (rarely), or by user-defined tags, such as "moon", for instance.

2.6.2 Story telling or random exploration?

As explained, five pairs of participants used a set of media items that stimulated a collaborative exploration and seven pairs used a set that propitiated storytelling. The use of those two different scenarios helped us understand the importance of offering both the free exploration mode and the presentation mode. In general, users recognized the utility of each mode, as in these excerpts:

> "If I... let's say... I have a party, I would just put the pictures unorganized there, people would start looking at them and talking. But in the moment where I really want to show somebody, I would like some functionality that go back to some ordering." (P5)

> "If you show pictures to your grandmother [...] it would be nice to have ordered [...]. In a big group, and to his friends, chatting [...], this way of showing pictures would be nice." (P7)

> "When you see pictures from a vacation from someone, you look, and he, and you... (referring to the free exploration mode) or when you tell everybody the same story, then it's better to have one picture... (referring to the presentation mode)" (P18)

This last quote is very meaningful, because it illustrates that people want to freely look at pictures from others, in their own pace; but at the same time, they want to show their own pictures

having the full control of the presentation. A participant commented that with digital photos we are *"back to the horrible seeing of family slides"*, where someone forces the others to listen to long and boring stories for each photo. Another participant explained that when he is with the family, usually one person controls the presentation, but the others influence the presentation by asking him/her to move the presentation forward or backward or by stretching the arm and taking control.

Another task that could benefit from the free exploration mode is the selection of a subset of media items, as suggested by one of the participants, who once a year selects a set of the best pictures to his mother.

2.7 Discussion

In this section, we try to answer the three research questions relative to: metaphor, digital ecosystem, and level of control. The discussion regarding the level of control is divided into a section about multiple control panels and another about personal spaces.

2.7.1 Physical photos metaphor, but with software support!

The discussions regarding the use of the metaphor of physical photos clearly indicate that users approve it. It was considered fun and intuitive. People liked the idea of sitting around a table and passing photos to friends. However, the need of some software support regarding the alignment and distribution of media items in the free exploration mode was explicitly indicated by some of the participants. It was also suggested by the attitude of users who placed items side by side for creating a grid that allows the visualization of several items at the same time without overlapping.

A volunteer commented that sometimes he looks at photos together with others using the thumbnail mode of his photo application, because it helps people to choose the photos they want to see enlarged. However, he says that it is not as good as the tabletop, because the thumbnails are too small. He suggested that the application should recognize a simple gesture—hitting the tabletop with a flat hand, for instance—to temporarily organize items in a grid. He emphasizes that items should not stay fixed; users should be able to freely move them afterwards. Other volunteer suggested the free exploration mode should never let an item overlap others, which means that moving or resizing an item would also rearrange others automatically. Other two suggestions related to this topic were the use of a shift key while handling an item in order to force it to be in a straight orientation and to move in a right line, and a predefined grid where users could place items.

Some situations and talks indicate other positive aspects of the metaphor of physical photos. Satisfied with the experience, a participant said:

> "The fundamental difference between here and how it was before, where you had photos to manipulate by hand, is that [in the past] after [use] you had to arrange

everything, to store away. Here you just need to turn off." (P15, authors' translation)

An interesting situation was observed while a user was searching items to show and comment with her partner. She naturally flicked some items to the borders saying *"Let me throw this away"*, in order to avoid being bothered by uninteresting items. This would be a perfect situation in which users could benefit from the "Black hole", a software support offered by the application presented by Apted et al. [2006], which allows a gradual and reversible deletion of items.

The free exploration mode was appreciated by the possibility it gives to multitasking. One could pass an item to a friend and let her see it while looking for another interesting item to comment on. A participant said:

> "It's already much better than, you know, on the small screen or in full screen, because you can watch some stuff and I can watch other stuff..." (P3)

However, it was also criticized by the lack of support to search. The same participant suggested the use of tap and hold to allow the activation of different search mechanism, and gives as an example, searching for items with a specific person.

The last hands-on part of the experiment helped us understanding that it is important to provide users with means to access distant items. Although participants achieved good results by using verbal requests to each other during the accomplishment of the task, users still missed software support. A participant remarked:

> "I don't like to be dependent on her. I feel helpless. [...] I felt like the cooperation was very good, but it was also necessary." (P5)

Some interaction techniques were proposed by participants in order to overcome this. The most welcome solution is to allow users to move the "tablecloth" (by dragging the background), and thus users would be able to bring all the items closer to themselves at the same time. This would be a temporary state. After releasing the "tablecloth", it returns to its original position. Participants were concerned about how other people around the tabletop would react in this situation. The application should have some mechanism to avoid that someone gets disturbed by the movement of the "tablecloth", such as locking this action while someone is touching a media item, for instance. A participant suggested that the "tablecloth" would be useful when someone is alone. Other solutions proposed were the use of voice commands and the use of a virtual rope to pull items.

Figure 2.7 shows the aggregated answers, sorted by relevance, to the questions related to the usefulness of implemented functionalities. The two most highly rated functionalities are related to the metaphor of physical photos.

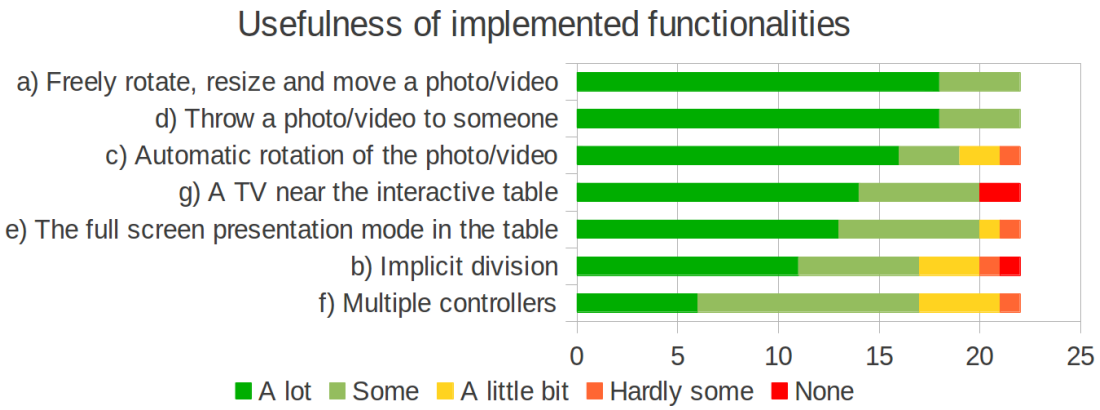


Figure 2.7: Post-test questionnaire: Usefulness of the implemented functionalities to explore photos and videos in an interactive tabletop.

2.7.2 Advantages and disadvantages of an additional vertical screen

Most of the participants preferred to use the tabletop when they could count on an auxiliary display. The scenario combining the tabletop and the TV is *"more open"*, said P17. A sexagenarian participant praised, saying that it is *"almost as good as old fashion slides"* (P9). In general, the idea is that the TV suggests an environment that supports a bigger number of people interested in seeing media items:

> "I think if you are with two people [...] the table is fine, but if you are with 10 [people]... then I think it's better to see it on the TV." (P14)

When comparing the tabletop + TV scenario with a tablet + TV scenario, the former was seen as *"much more family, group oriented"* (P3). Another volunteer said that, using the tabletop, everybody sees the context from where the photos are coming from. It gives an idea of the amount of photos available, and what are the previous and the next items to be seen. Someone also suggested that the tabletop could be used in a scenario involving remote people.

A relevant factor for the participants when indicating the preference to see items in the TV was that it shows brighter colors. Only one participant saw this as a disadvantage, arguing that the colors on the TV are not as natural as on the tabletop. However, the great majority liked it, using adjectives, such as *"exceptional"* (P15), *"clear"* (P16), and *"phenomenal"* (P22). Other factors that may have influenced this preference were the less reflective surface of the TV, and the *"cool factor"* (P18), as commented by an adult female participant.

Without any doubt, however, the most determinant factors were the comfort, viewing angle and stance propitiated by the use of the TV. During the experiments, many participants complained about their stance while interacting with the interactive coffee table:

> "I don't think I would be able to keep it up for very long. I think my back would start to hurt." (P11)

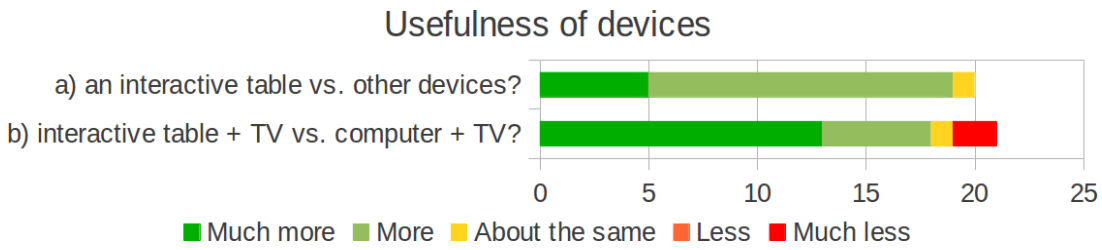


Figure 2.8: Post-test questionnaire: To COLLABORATIVELY explore photos and videos, how useful is... a) an interactive table when compared to a computer / tablet / smartphone? b) an interactive table connected to a TV when compared to a computer connected to a TV?

> "That's something you could do for a very short period of time [...] Fine, we are young, right? [...] I wouldn't say adults would do this [use for long period]. Children maybe, adolescents maybe." (P23)

> "There is nothing comfortable about being bent over the table." (P15, authors' translation)

We also observed that some users stood up during the experiment in order to better see the media on the tabletop. A solution presented independently by two of the participants was to offer means to incline the tabletop, so that they could sit back in a more comfortable way while still being able to interact with it. The use of a lever was suggested, as the table is heavy. A tabletop with a changeable screen shape have already been suggested to allow different spatial arrangements of users around the table [Takashima et al., 2013]. A solution that tilts the tabletop could make the TV superfluous and would avoid the need to move the head in order to split the attention between two devices. However, it would be applicable only in cases of very small groups, as it would restrict even further the usable space around the tabletop. A different solution would be to use new technologies that make the tabletop thinner, such as presented by Jackson et al. [2009], to let people put their legs under it. But in this case it would no longer be a coffee table.

Lastly, two other things were also suggested: a mode in which the TV works as a mirror of the tabletop, so that people could comfortably follow the interaction of others with the tabletop, and an automatic presentation mode, so that people could *"sit back and relax"* (P9).

Figure 2.8 shows the aggregated answers of two questions of the post-test questionnaire comparing different devices and combinations of devices. It can be seen that participants see an interactive tabletop as a useful tool for group tasks.

2.7.3 Multiple control panels

Both in the presentation mode and in the TV mode, the application shows two control panels, one in each lower corner of the tabletop. Participants did not have a clear opinion about it, but in general they tended to believe that its suitability depends on the number of people around the table and the task being performed.

Some suggested that a single control panel could be used, as long as it is movable so that users can pass it to someone else. We believe that this is the model with which they are used to when using TV remote controls; however, it is not necessarily the best option to be used in a tabletop. When the performed task has a presentation nature, users tend to see a single control panel as more appropriate. Still, a user saw the multiple control panels as a tool to interrupt boring presentations. When the interference from others really has to be avoided, the control panel can be moved to a personal device. Considering a task with a more explorative nature, seems that the best configuration would be to provide a control panel for each two or three users, so that everybody has a control nearby. Moreover, using less than one control for each user helps promoting awareness of actions performed by others and saves screen space.

A concern related to the use of multiple control panels is that it may become source of conflicts, especially among children. However, there was a volunteer that claimed that there are many other aspects that may cause conflicts; the use of two or more control panels is not the worst one. Others consider that users would talk and respect each other. When asked if adults would fight because of this, P10 answered: "We are too old to fight". Some interaction conflicts, not only related to the use of two control panels, could be witnessed during the experiments, but they were easily solved by a user asking the other to stop interacting for a moment. We could also see a participant using one of his hands to hold the partner's hands in order to perform an action without interruption.

2.7.4 Personal spaces

Our application makes a moderate use of the personal spaces. However, several suggested functionalities are indirectly related to this concept. One suggestion is that people should be able to use a restricted area of the tabletop in order to enlarge a media item as much as they want without disturbing others. Participants were frustrated by not being able to expand a media item. When they had the chance to expand items indefinitely (while the presentation mode was disabled) they clearly demonstrated satisfaction: *"Beautiful!"* (P8), *"There is no limit!"* (P17). The personal space should be used as this restricted area. When zooming an item inside the personal space, users should be able to indefinitely enlarge it, without extrapolating the personal space border. Although intuitive, the activation of the presentation mode by zooming caused a lot of complains. One of them is that if it is triggered by mistake, it can startle or disturb other users. *"This one is really bad for multiuser"*, said P5. One of the suggestions given to avoid this is the use of a highlight around the entire screen as a hint that if the user releases the fingers, the activation will be triggered. This is also a problem with the activation by a two-finger tap, but it is less likely to happen by mistake. Techniques for adding new user's interface object on a shared device while preserving mutual awareness of the participants without disturbing them are discussed by Belatar and Coldefy [2010]. Another suggestion was to offer a small presentation mode inside personal areas, so that users could navigate media items in a structured way without

disturbing others.

If the tabletop technology allows the identification of the user that touches a certain point—by means of hand orientation, for instance—personal spaces can be used as a no touch area for the others, thus providing a more individualized space, as in the work of Apted et al. [2006]. A private space to visualize confidential media items was asked by one of the participants.

Participants approved the use of the personal space boundary as a trigger to the automatic rotation of the media items. However, participants suggested that an item should be rotated only when entering a personal space, not when leaving it.

2.8 Summary and future directions

In order to investigate issues related to the use of an interactive tabletop as a coffee table that allows the exploration of personal photos and videos, we have conducted a user study involving 24 volunteers. The development of the prototype application used in the experiments was guided by a requirement elicitation questionnaire answered by 55 people. More specifically, we aimed to answer three research questions related to our scenario. Although our research has some limitations, such as the use of small media collections and a hardware offering low sensitivity, we believe we have managed to achieve our goals.

The first question aimed to identify whether a digital interface that uses the metaphor of physical photographs placed on the table is desirable. Our analysis show that people liked the idea of sitting around a table and passing photos to family members and friends, as in the old times—however, it is necessary to complement the experience by offering some software support regarding the alignment and distribution of media items on the tabletop.

The second question investigated the combined use of the tabletop with an additional screen. Our results suggest that the use of a TV helps creating an environment that supports several users and improves the experience by providing a better image quality and comfort.

The question related to the desired level of interaction can be divided into two parts: usefulness of multiple control panels and personal areas within the tabletop. Regarding multiple control panels, the answer is dependent on the task at hand. For storytelling scenarios, it seems that a single control panel is more appropriate. In a more explorative scenario, the use of a control panel for each two or three users may be the optimal configuration. Finally, we obtained strong evidences that personal spaces would be very useful as a space that allows zooming and navigation capabilities without disturbing other users. Private spaces that allow confidentiality on the tabletop should not be prioritized in applications for media exploration. Lastly, the use of the personal space boundary as a trigger to the automatic rotation of the media items is appropriate, but abrupt rotations should be avoided.

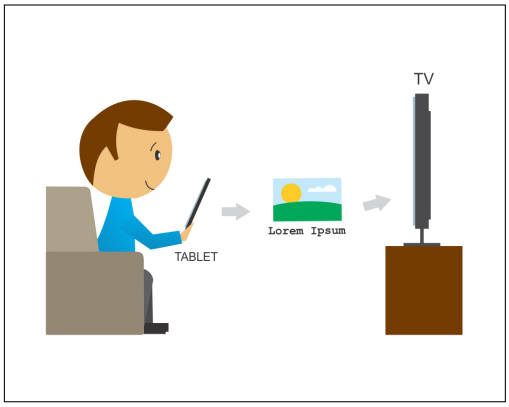
As future work, we suggest the development and evaluation of a tabletop hardware that allows it to be tilted, potentially making superfluous the use of a secondary screen. Regarding the challenge of level of control, more specific experiments using a prototype version that provides

flexible use of control panels would be useful to confirm our findings. Also, a more flexible use of personal spaces is still to be investigated. Some of the comments given indicate that it would be convenient to allow a dynamic creation and resize of personal spaces. This would allow the use of a "personal" space by a subgroup of users, or the use of multiple personal spaces by the same user, while performing a classification task, for instance.

In the next chapter, we concentrate on the benefits of connecting several devices with a television set. Instead of using the TV set as an additional screen for a tabletop, as we do while investigating the second research question of this chapter, we switch the focus to the TV. In Chapter 3, we are interested in using additional devices to provide input for iDTV applications.

Chapter 3

Multimodal interaction component for iDTV



In most current interactive DTV applications user interaction is restricted to keys presses on a remote control. For simple applications this type of interaction is sufficient; however, as interactive applications become more popular new input devices are demanded. After discussing motivating scenarios, this chapter[1] presents an architecture that offers to applications running on a set-top box the possibility of receiving multimodal data (audio, video, image, ink, accelerometer, text, voice and customized data) from multiple devices (such as smartphones, tablet, laptops or even desktops). We validate the architecture by implementing a corresponding multimodal interaction component that extends the Brazilian Digital TV middleware, and by building applications that use the component. Thus, we show that the proposed component facilitates the development of innovative applications, since it helps to overcome one of the main restrictions in the development of iDTV application, which is to rely solely on inputs from the remote control.

[1]This chapter is based on the following papers:

- **D. Pedrosa**, J. A. C. Martins Jr., E. L. Melo, and C. A. C. Teixeira. A multimodal interaction component for digital television. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, SAC '11, pages 1253-1258, New York, NY, USA, 2011. ACM. doi: 10.1145/1982185.1982459. URL http://doi.acm.org/10.1145/1982185.1982459.
- **D. Pedrosa**, J. A. C. Martins Jr., M. G. C. Pimentel, and E. L. Melo. Componente de Interação Multimodal no Ginga. In: *II Workshop de TV Digital Interativa* (WTVDI) do WebMedia '10. Belo Horizonte, MG. Proceedings of 16th Brazilian Symposium on Multimedia and the Web, v. 2, pages 197-202, 2010.

3.1 Going beyond the limitations of remote controls

In most existing iDTV applications, the user interaction takes place by pressing keys on a remote control—a classic device with arrows, OK, BACK, numbers, and colour keys. This type of device is sufficient for applications that only display additional contents and which allow users to navigate in such contents. However, as interactive applications become more popular, new input devices—such as microphones, cameras, touch screens, electronic pens and accelerometers—are demanded.

Examples of iDTV applications are electronic program guides (EPG), applications related to a TV show or major events (the FIFA World Cup and elections, for example), games and news [Morris and Smith-Chaigneau, 2005; Peng, 2002]. These applications offer few features because they must be easy to use, even if used only with the remote control as an input device. The restriction on the remote control—compared with a full keyboard and a direct manipulation device such as a mouse, that are common on personal computers—limits the usability of iDTV applications [Carmichael et al., 2006]. According to Roibás et al. [2005], all navigation interfaces that rely on a conventional remote control are far from expressing the interactive potential of an iDTV system. The development of richer applications may require complex interfaces, what would limit its popularization [de Miranda et al., 2008]. Cortez et al. [2012] say that the limitation of TV's input modality, based on "*a few number buttons and a directional pad*", "*is a burden on anyone trying to build rich cinematic Internet experiences to the TV*".

In this chapter, we present the architecture of a component that allows richer iDTV applications to be developed, offering them the possibility of receiving multimodal data from different devices. After discussing motivating scenarios, this chapter presents an architecture that offers to applications running on a set-top box the possibility of receiving multimodal data (audio, video, image, ink, accelerometer data, text, voice and customized data) from multiple devices (such as smartphones, tablet, laptops or even desktops). To validate the architecture, we implemented a corresponding multimodal interaction component which extends a digital TV middleware—we experimented with the Ginga Brazilian middleware—and we built applications that use the component.

3.2 Motivating application categories and the importance for accessibility

Works such as those reported by César et al. [2006] and Cattelan et al. [2008] aim to allow users to create or enrich multimedia contents. Their research exploit the case in which the contents from the television broadcast can be annotated by viewers and the enriched content can be presented on the television, but the authoring is performed using applications that run on a portable device such as a tablet PC. The interaction with the tablet allows elaborate annotations,

such as notes made with electronic ink, for instance. Other studies [Pimentel et al., 2010; Teixeira et al., 2012] also present tools for authoring multimedia content, but the interaction is restricted to the TV remote control. Their tools give few annotation options comparing to those that run on a tablet PC. In the work from Pimentel et al. [2010], the annotations are restricted to bookmarking a scene, selecting a time interval to be skipped when watched later, and selecting a time interval to be reviewed later in a loop. Teixeira et al. [2012] allow users to personalize their programs with previously created pieces of content that may be temporarily attached to the video. We observe that there is an opportunity to explore the strengths of these different types of content production: the use of devices that allow richer input formats and the comfortable environment brought by TV sets.

Teixeira et al. [2009] explore the established habit that users have of loudly commenting television contents and present a scenario in which users can create ink annotations on top of scenes of soccer matches. Something similar can be imagined in the context of a soap opera, films or various other TV shows. In such a context, a generic application allowing electronic ink annotation on video contents would benefit from multimodal input mechanisms.

Several projects in distance education make use of TV. In educational programs, such as the Novo Telecurso[2], widely known in Brazil, it is interesting to enable active participation of students through the use of tools like chat and electronic whiteboard. iDTV applications that explore more complex input devices such as keyboards and tablets could be used both in cases where students are in their homes and when they are in a classroom sharing the same TV set.

The use of other input devices emerges as a possible solution for applications which require text input. Examples are news applications that allow user comments (a common feature in most news and blog sites), e-commerce applications that ask users to fill out the name and delivery address, and applications that allow viewers to communicate through e-mail, SMS, or chat (as shown in Figure 3.1) . Examples of applications that allow content search by keyword are presented in the work of Wittenburg et al. [2006], Ibrahim and Johansson [2002], and Johnston et al. [2007]. They support voice utterances as input. The latter also supports pen input, including both pointing and handwriting, and a combination of these modalities. Considering applications that aim at allowing users to communicate while watching TV, Chat TV[3] indirectly uses a phone as an input device. Another example is Media Center Buddies [Regan and Todd, 2004]. The next chapter provides a comprehensive discussion about current and innovative methods for text input in iDTV applications.

Game applications are present in many iDTV environments: in this case, the use of input mechanisms that allow a more natural interaction has already been reported. Wii[4] was the first game console to provide an accelerometer control device; manufacturers of other consoles seem

[2] `http://www.telecurso2000.org.br`, accessed 27/Aug/2014.

[3] It was active in Brazil for several years since 2005 (`https://www.youtube.com/watch?v=lBZhOIJ1ma4`, accessed 27/Aug/2014) and it is still active in Malaysia (http://www.astro.com.my/itv/indices/index_233.html, accessed 29/Aug/2014).

[4] http://wii.com, accessed 15/Oct/2014

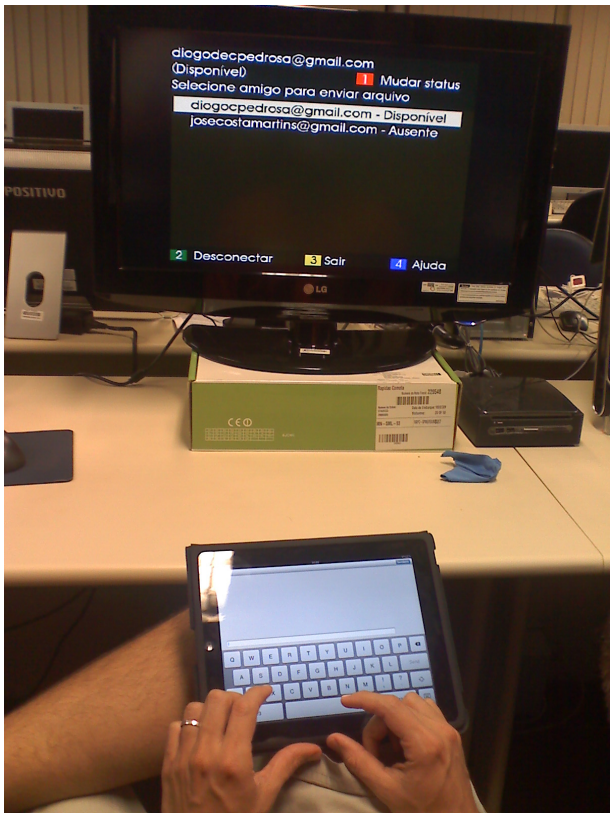


Figure 3.1: Text input provided via iPad.

to be developing similar devices. It is natural to imagine iDTV games that also explore data input from an accelerometer equipped device.

By offering easy integration with external input devices, a digital TV middleware may help to increase the accessibility of iDTV applications. Users with disabilities who are already using specific devices to overcome their limitations may also use them when interacting with iDTV. It is important that, besides allowing multimodal inputs such as text, audio, video and gesture, the infrastructure also allows different devices to provide simple key presses. For example, it should allow users to change channel using their mobile phones. Thus, applications need not worry about which type of device generated the event, it simply performs the appropriate action. Nakajima [2006] presents good arguments to allow existing interactive applications to be operated from mobile devices in ubiquitous computing environments. The attempt to offer to minority groups the same possibilities of using systems through alternative forms of interaction, in order to democratize access to information and entertainment, is being investigated in various studies, including in the field of Interactive Digital TV [Carmichael et al., 2006; Rice and Alm, 2008; Roibás et al., 2005; Springett and Griffiths, 2008].

The possibility to provide input to TV using additional devices has yet another benefit. As discussed in the previous chapter (Section 2.7.3), there are some situations and tasks in which it is beneficial to allow more than a single user to control the system. McGill et al. [2014] investigate and discuss how TV systems can be used my multiple users, without the bottleneck of a single physical remote control. However, they analysed only a task designed for control by one

user at a time. They conclude that TV systems should be augmented with mediation of control schemes to allow users to retain the familiar interfaces and mental models, while allowing new input mechanisms and modalities to be used in an effective and useful way.

3.3 Related work on support for multiple devices

In the context of Brazilian digital television system, whose middleware contains an imperative application environment (Ginga-J) and a declarative application environment (Ginga-NCL), Soares et al. [2009] present the hierarchical control model used by Ginga-NCL to support multiple devices. The authors argue that producers of video content do not wish to have their contents overlaid with additional information and suggest the display of the graphical interface of applications in a secondary device. Another argument given is that overlapping contents normally bother other viewers in the same room. They present a sample application whose graphical interface is shown on a secondary device to offer soccer boots for sale while the TV screen presents an animation about a famous soccer player.

Still in the Brazilian digital television context, a work from Brandão et al. [2010] presents the LuaTV API, which is part of the draft specification for the NCLua Extended API. It is composed of 4 modules, two of which are related to this research: *HAN*, which offers high-level access to commonly available resources on home networks, and *widget*, aimed at graphical support to applications.

Although the scenarios presented by these works use secondary devices, their main feature is to explore the additional display. It is a relevant concern, as showed by the field trial conducted by Basapur et al. [2011]. However, there is still a lack of APIs to support multimodal data input for iDTV applications. The scenarios presented by Soares et al. [2009] and Brandão et al. [2010] allow the secondary device to provide input to iDTV applications, but they neither help handling the combination of two or more input modes nor offer specific support to all data types provided by our Multimodal Interaction Component (MMIC), such as ink, voice and raw accelerometer data.

Works that involve the use of multiple devices in the imperative environment of the Brazilian middleware are presented by Silva et al. [2007; 2008]. These works present a Java API that allows applications to use device resources, such as cameras to capture video and photo and microphones for audio, telephone networks, displays, and speakers. Despite allowing input of multiple types of data, the presented API has two major differences compared with what is offered by the component described in this chapter:

1. The initiative of data transmission comes from different sides. By using a remote or MMIC, the data is sent to the middleware. In Silva et al.'s work is the middleware that fetches data from the device. As a result, while with their Java API the application has to be aware of the presence of the device before it can explicitly request any data from it, using MMIC, the application is constantly ready to receive data from the viewer's device.

2. The Java API does not support the development of multimodal applications, as it does not support the combination of natural forms of human expression, such as speech, touch, gestures and body movements. In addition to audio, video and image files supported by the work of Silva et al., MMIC supports events that combine electronic ink, accelerometer data, voice, purely textual data, and generic data to the cases not anticipated.

In the context of MHP, the middleware of the European digital television system, Lobato et al. [2009] also present an architecture for the development of iDTV applications that use inputs other than the ones provided by remote controls. The main difference is that the additional input types are restricted to utterances. Devices must provide audio I/O capabilities and must have systems for speech recognition and synthesis implemented.

In more recent study, posterior to the publication of the architecture that we present here, Cortez et al. [2012] describe a protocol and architecture that enable TV applications to use a two-way communication channel between the TV and tablets, smartphones, and laptops. Their protocol, based on message passing, emphasizes secured end-to-end communication. As a usage example, the authors cite an application that detects the TV show watched by the user and sends to a companion device some more information about the episode. The bottom line is to shift interactions from the TV to secondary devices, instead of using secondary devices to provide input to TV applications.

Their work aim to remove the disconnection between what research and standards say is needed regarding the use of secondary devices with TV systems and what can happen in commercial smart TV systems. Smart TVs are TV systems that connect to the Internet and allow users to watch streamed media and install applications via an online store [Kovach, 2010]. As Chromecast, smart TVs also allow personal devices to stream content to be displayed on the TV screen. Some examples of Smart TV plataforms are Google TV[5] and its successor Android TV[6], Apple TV[7], and Amazon Fire TV [8]. Cortez et al. provided a second screen API, built into the Yahoo! Connected TV platform[9]. Some other manufacturers, such as LG, Philips, Toshiba, and Vestel, have formed the Smart TV Alliance[10] to align on the use of a common technology based on HTML5, to enable application developers to create applications that run on all supported Smart TV Alliance platforms.

3.4 Architecture of a multimodal interaction component

To give applications the opportunity to receive multimodal data from different devices without the need to establish an explicit connection, protocols for automatic configuration and service

[5] `http://www.google.com/tv`, accessed 14/Oct/2014.

[6] `http://www.android.com/tv`, accessed 14/Oct/2014.

[7] `https://www.apple.com/br/appletv`, accessed 14/Oct/2014.

[8] `http://www.amazon.com/Streaming-Internet-Media-Player/oc/Fire-TV`, accessed 14/Oct/2014.

[9] `https://smarttv.yahoo.com`, acessed 16/Oct/2014.

[10] `http://smarttv-alliance.org/`, accessed 16/Oct/2014

discovery in IP networks such as Zero Configuration Networking (ZeroConf[11]) and Universal Plug and Play (UPnP[12]) should be used.

Figure 3.2 illustrates the three main modules of our architecture, which we implemented in a Multimodal Interaction Component (MMIC): communication modules, extended event manager, and parser. *Devices* send data in an XML format (explained in Section 3.4.1) to the component. The data is received by one of the available *Communication Modules* (Section 3.4.2) and transferred to the *Event Manager* (Section 3.4.3), which uses a *parser* to transform the XML document received into an `IMultimodalInputEvent` object. Then, the *Event Manager* notifies the applications that registered themselves as multimodal input listeners.

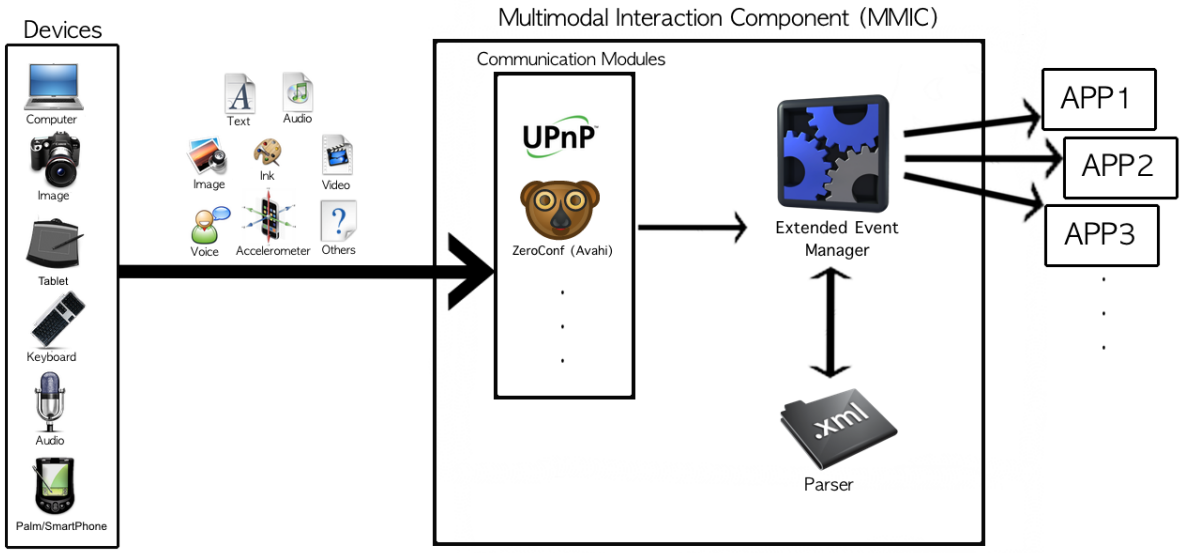


Figure 3.2: MMIC Architecture.

The conversion of the received data into a multimodal event simplifies the application development because it allows the use of the already familiar mechanism of event listeners, and especially because it provides a standardized way to access data. The location of MMIC in the Common Core of Ginga makes it usable both by the imperative environment (Ginga-J) and by the declarative environment (Ginga-NCL), that is, data can be accessed through Java methods, in the case of Xlets, or through tables with pre-specified fields, in the case of Lua applications.

This study addresses only the case in which a communication module receives input data through computer networks, but the architecture is not limited to it. The case in which USB peripherals are connected directly to the decoder, for example, may also be considered.

The Multimodal Interaction Component was developed using the virtual machine containing version 0.11.2 of the Ginga Reference Implementation[13].

[11] `http://www.zeroconf.org`, accessed 27/Aug/2014.

[12] `http://www.upnp.org`, accessed 27/Aug/2014.

[13] The source code is available at the Ginga Community of the Brazilian Public Software Portal (`http://www.softwarepublico.gov.br`, accessed 27/Aug/2014) and the virtual machine is available at http://www.gingancl.org.br/en/ferramentas (accessed 27/Aug/2014).

3.4.1 Multimodal events

Just as keystroke events are represented by the existing `IInputEvent` interface, so are multimodal events represented by the created `IMultimodalInputEvent` interface. A single multimodal event can transport at the same time all the available types of data, and it is composed of:

- `id` – A string that identifies the event. If it is one of the values specified for the key atribute of the simpleCondition element in the Ginga-NCL standard [Associção Brasileira de Normas Técnicas, 2007], a regular input event is also dispatched.
- `strings` – A vector of strings, used so that ready texts on secondary devices can be sent to applications.
- `ink` – An object of a class that stores electronic ink data as a result of the interaction between the user and a touch-sensitive device or an electronic pen, for example
- `accel` – An object of a class that stores data resulting from the interaction between the user and an accelerometer.
- `binaries` – A vector of objects that store data (name, mimetype and contents) of binary files, such as any format of audio, video and image files.
- `voice` – An object of a class that stores data representing speech, which may be the result of speech recognition systems or may serve as input to speech synthesizers systems.
- `valuesMap` – An object that maps keys to values. It provides greater flexibility because it gives the application the possibility to receive types of data that were not predicted by the API. Another possible use is to carry metadata, such as the width and height of an image present in the event.

A protocol in XML format was defined in order to allow devices to send multimodal data to Ginga. The protocol provides a header containing the device ID, the user ID, and the start / end time of the generated event. The data itself appear in the event body. Listing 3.1 shows an example of a multimodal event.

Listing 3.1: XML document excerpt exemplifying a multimodal event

```
1  <?xml version="1.0"?>
2  <multimodal ... id="testEventID">
3    <head>
4      <device id="DEADBEEF-DEAF-BABA-FEED-BABE00000006"
5              model="iPhone 3GS"/>
6      <user id="59616261-6461-6261-4E50-472050325033"/>
7      <timestamp begin="2010-05-19T09:30:10.5"
8                 end="2010-05-19T09:30:17.8"/>
9    </head>
10   <body>
11     <value id="width">400</value>
12     <value id="height">300</value>
13     <text>Beach picture</text>
```

```
14      <text>What a beautiful place!</text>
15      <inkml:ink>
16        <inkml:trace>10 0, 9 14, 8 28, 7 42, 6 56, 6 70,...
17        </inkml:trace>
18        <inkml:trace>130 155, 144 159, 158 160, 170 154,...
19        </inkml:trace>
20        ...
21      </inkml:ink>
22      <accel xValue="3" yValue="-2" zValue="1"/>
23      <voice>Here goes voicexml data</voice>
24      <binary filename="beach.jpg" mimetype="image/jpeg">
25              PD94bWwgdmVyc2l...</binary>
26    </body>
27  </multimodal>
```

We chose to use the InkML[14] W3C standard to represent electronic ink data. The developed parser uses code from the inkMLLibcpp[15] library, which was adapted to have the TinyXML[16] library replaced by the Xerces[17] library because the latter was already being used in other parts of Ginga. We also aim to use the VoiceXML[18] standard but the current implementation of the component does not support voice data.

3.4.2 Communication modules

As previously mentioned, communication between devices and decoder should be done using protocols for automatic configuration and discovery of services over IP networks, such as UPnP and ZeroConf. The current version of MMIC uses the Avahi[19] library and supports ZeroConf. A communication module that supports UPnP could be implemented so that MMIC can offer more flexibility for devices that wish to send data to iDTV applications.

To be able to send multimodal data, an application running on a secondary device must search for the ZeroConf service offered by the communication module and connect to a socket using the IP address obtained from the service. Multiple connections can be opened simultaneously.

Communication modules run on separate threads, which are initiated by the Event Manager constructor and stop running only when Ginga is finalized. This ensures that the middleware is always ready to receive multimodal events.

3.4.3 Event Manager

The Event Manager is the main module of the component. It extends and replaces the Event Manager of the Ginga Reference Implementation in a way that it is now also responsible for

[14] `http://www.w3.org/2002/mmi/ink`, accessed 27/Aug/2014.
[15] `http://sourceforge.net/apps/trac/inkmltk/wiki/InkMLLib`, accessed 27/Aug/2014
[16] `http://www.grinninglizard.com/tinyxml`, accessed 27/Aug/2014
[17] `http://xerces.apache.org/xerces-c`, accessed 27/Aug/2014
[18] `http://www.w3.org/TR/voicexml21`, accessed 27/Aug/2014
[19] `http://avahi.org`, accessed 25/Sep/2014.

receiving multimodal data from communication services, encapsulating them into events, and dispatching them to interested applications.

Due to the componentized way the Ginga Reference Implementation has been developed, the integration of the extended event manager in Ginga was an easy task to accomplish. We only needed to edit the configuration file of the Ginga Common Core Component Manager to indicate that the new Event Manager, `EnhancedInputManager`, should be used in place of the `InputManager` present in Ginga Common Core System.

One of the responsibilities of this module is to maintain a new list of listeners, specific to multimodal events, allowing interested applications to add / remove a listener to / from the list. Every time input data are generated by a device and passed to the Event Manager via a communication module, the manager creates an object that represents a multimodal input event, fills it with the received data, and iterate through the list of listeners to notify them about the occurrence of the event.

3.5 Preliminary validation

We developed two applications to initially validate MMIC. Both are composed by two parts: one that runs on a secondary device and transmits the multimodal data and one that runs in the set-top box and uses the received data.

3.5.1 Multimodal interaction: testing production and consumption

The first part of the first application (client-side) was developed in C and runs on a terminal on any device that is located on the same network as the set-top box and supports the ZeroConf protocol. The role of the client-side is to send multimodal events to Ginga. As soon as it starts, it searches the ZeroConf service provided by the communication module and starts a loop that sends the contents of an XML code located in the same directory each time the ENTER key is pressed. This pre-defined XML code is very similar to that presented in Section 3.4.1.

The second part of the first application (STB-side) was developed in C++ as a resident application that is initiated by the main function of Ginga. Its goal is to show all contents of multimodal events received as log messages in a terminal. Multimodal applications must implement the `IMultimodalInputEventListner` interface, register themselves as listeners using the method `addMultimodalInputEventListener` of the `EnhancedInputManager`, and process the events received within the method `userMultimodalEventReceived`. In addition to showing all the event data using log messages, this test application also draws the traces contained in the ink tag of the event, as shown in Figure 3.3. The traces shown have been copied from the InkML W3C standard example.

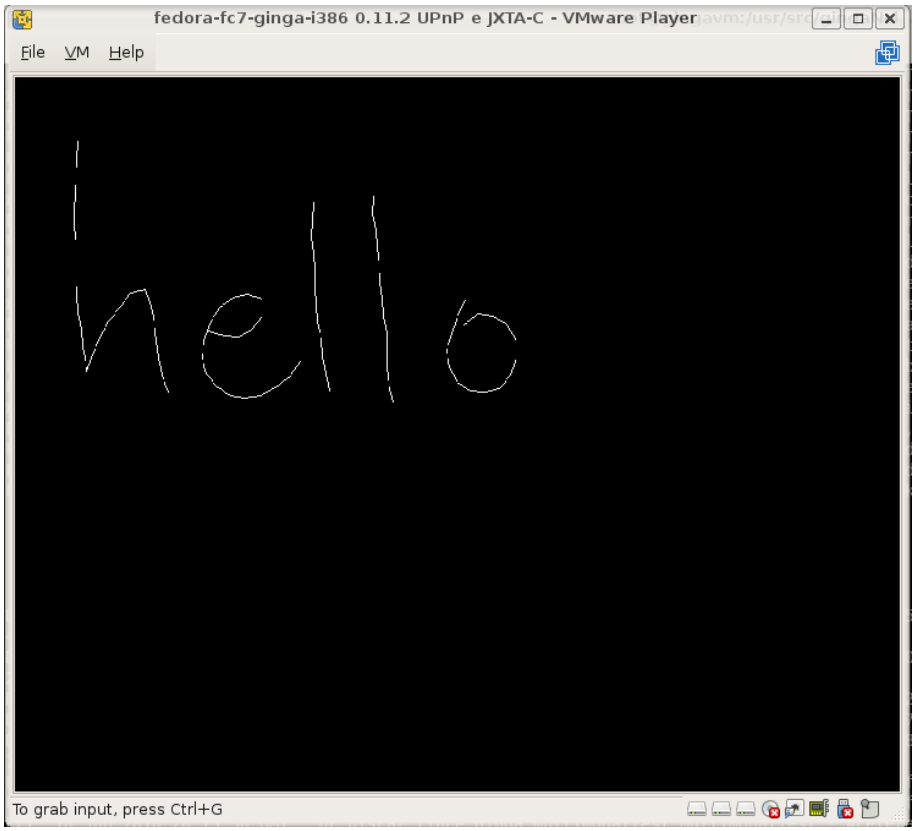


Figure 3.3: Screenshot of a MMIC test application that shows the traces of multimodal events received.

3.5.2 Multimodal interaction: a chat application

The second application is interested only in the text strings and binary files of multimodal events. However, differently from the previous application, its goal is not just to test the MMIC component: it is an application for communication which allows users to exchange text messages with each other, and to share files.

The external device (client) part of this application runs on an iPad. It has a graphic interface with only a virtual keyboard, a text field, and a send button, as shown in Figure 3.4 (left). Each time a button is pressed, it creates an XML message containing the text present in text field and sends it to the middleware.

The *chat* application itself is also a resident application developed in C++. It is composed of several screens and some other extra functionalities. In this presentation, we focus our attention on the chat screen, shown in Figure 3.4 (right), because it is the one that handles multimodal events. At this screen, when a multimodal event arrives, the application sends all the strings contained in the event as text messages to the friend with whom the user is talking to.

As far as applications are concerned, there are very few lines of code related to receiving multimodal inputs. The first thing an application must do is to register a multimodal input listener in the input manager, as shown in Listing 3.2.

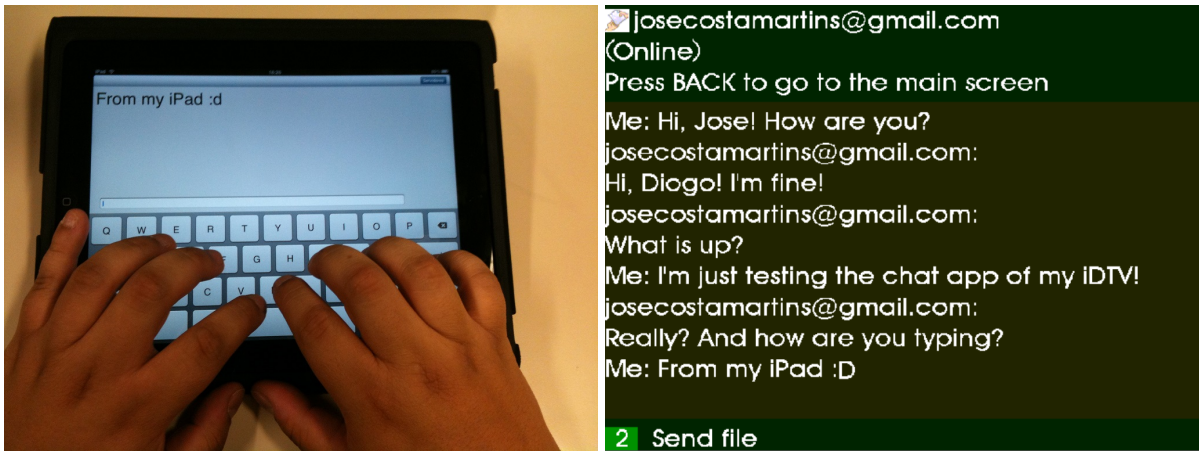


Figure 3.4: iPad interface for input messages and files to a iDTV chat application (left) and chat screen of a iDTV resident application (right).

Listing 3.2: Application registering itself as a multimodal event listener

```
static IComponentManager* cm = IComponentManager::getCMInstance();
IEnhancedInputManager* eim  = ((EnhancedInputManagerCreator*)
        (cm->getObject("InputManager")))();
eim->addMultimodalInputEventListener(this);
```

As the P2PTestApp implements the *IMultimodalInput-EventListener* interface, when a multimodal input event arrives, its method *userMultimodalEventReceived* is called. At this moment, the application has access to all data contained in the event and, thus, it can do whatever it needs, as shown in the Listing 3.3.

Listing 3.3: Accessing strings contained in a multimodal event listener

```
bool P2PTest::userMultimodalEventReceived(IMultimodalInputEvent* ev) {
    ...
    vector<string>* strs = ev->getStrings();
    for (vector<string>::iterator j = strs->begin(); j != strs->end(); j++) {
        ...
        temp << "Me: " << *j;
        ...
    }
    ...
}
```

3.6 Improvement opportunities

The current version of our component allows resident applications written in C++ to receive multimodal events. To allow a broader use of the component, APIs that enables Java and Lua applications to receive those events have to be implemented.

To further facilitate the use of the component, the protocol may extended to allow the use of different formats for the same data type. Therefore, to use another ink data representation format, for example, it would be necessary to add an extension to our MMIC component that allowed the interpretation of this new format.

After these steps, the implementation of a more robust application may be done. The interactive document editor prototype presented by Pimentel et al. [2010] could be extended in order to allow it to accept multimedia data such as those offered by the MMIC component.

Regarding the Event Manager, 3 new functionalities could be included:

1. To extract additional data of the contents received by the protocol and to include these metadata in the multimodal event created, thus providing more complete information to applications. Examples of metadata are height and width of an image.
2. To allow applications to add listeners for events containing specific data types (voice and/or ink, for example) instead of all multimodal events.
3. To capture all occurred interactions. In other words, the Event Manager is responsible for recording each event, in order to allow automatic authoring of corresponding multimedia documents or knowledge extraction regarding user actions.

3.7 Summary and future directions

The component we have presented broadens the range of options available to application developers, since it helps to break one of the main constraints of iDTV application development, which is to rely solely on inputs from the remote control. One of the application types that can benefit most from the features introduced in the component is the one that enables end users to create contents. Also, applications dependent on text input will be highly benefited.

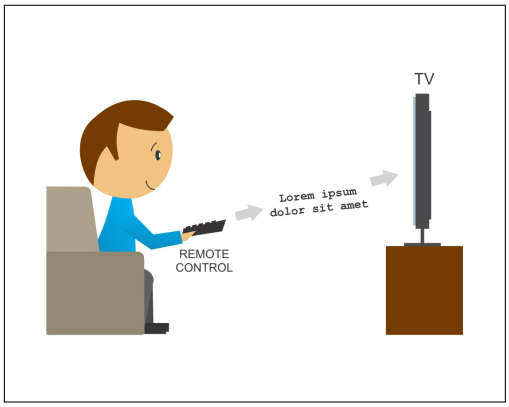
One aspect that should be reinforced is that the possibility of integrating the decoder with external input devices helps increasing the accessibility of iDTV applications. During definition of the architecture and the specification of the protocol, we tried to adopt previously established standards such as InkML, VoiceXML, ZeroConf and UPnP. The Ginga componentized architecture helped to achieve this second goal.

The lack of support for multimodal interaction in iDTV applications was the main motivation of this work. Although the Ginga standard does not foresee multimodal interaction, the good performance of the component in more robust implementations can result in a proposal to include this in future versions of the standard.

In this chapter we favored the usability of TV applications that require text input by allowing the use of external devices for entering text. However, the challenges for those who, because of convenience or lack of extra devices, want to enter text using only the remote control still persist. In the next chapter we deal with the scenario in which text input is restricted to the remote control.

Chapter 4

Text input using a remote control



With the popularization of interactive TV applications, a demand for efficient text input methods will emerge. A way to overcome this bottleneck is to provide multiple alternative mechanisms for aiding users to enter text. In this chapter[1], we describe the model of a software component that allows text entry in iDTV applications based on an interface with three input modes. We discuss our considerations with respect to the design, development and evaluation of a prototype corresponding to our model, built according to the user-centered design methodology. After conducting a research on existing text input methods in television systems, we interviewed four experts in the interactive TV domain and applied questionnaires to TV users. During the development of the prototype, we conducted usability evaluations using different techniques. The evaluations detected both a number of problems and several improvement opportunities; at the same time, they highlighted the importance of using complementary text input modes in order to satisfy the needs of different users. Overall, the evaluation results suggest that the proposed approach provides a satisfactory level of usability by overcoming the limitations of text input in the context of interactive TV applications.

[1]This chapter is based on the following papers:

- **D. Pedrosa**, D. A. Vega-Oliveiros, R. P. M. Fortes, and M. G. C. Pimentel. Text Input in Digital Television: a Component Prototype. In *Adjunct Proceedings of the Eighth European Conference on Interactive TV and Video*, EuroITV '10, pages 75-78, New York, NY, USA, 2010. ACM.
- D. A. Vega-Oliveiros, **D. Pedrosa**, M. G. C. Pimentel, and R. P. M. Fortes. An approach based on multiple text input modes for interactive digital TV applications. In *Proceedings of the 28th ACM International Conference on Design of Communication*, SIGDOC '10, pages 191-198, New York, NY, USA, 2010. ACM. doi: 10.1145/1878450.1878483. URL http://doi.acm.org/10.1145/1878450.1878483.

4.1 Multiple input modes for text entry

As discussed in the previous chapter, the development of interactive applications is hindered by the small number of user-interaction options of traditional remote controls. We have discussed the possibility of using additional devices to provide multimodal data to interactive TV applications. We now concentrate on improving the usability of text input in a scenario where only a remote control is used.

A way to foster the development of iDTV applications is to use multiple input modes to provide mechanisms for aiding user to enter text, for instance. Several classes of applications can use text input from the user, such as chat, e-mail, search on the electronic program guide (EPG), calendar and forms. Current solutions for text input are not satisfactory.

More natural forms of expression and interaction may be explored. They can be captured by some kind of technology and used by systems to allow users to interact naturally and easily. These natural forms extend the concept of interfaces based on mouse and keyboard interaction, known as WIMP (Windows, Icons, Menu and Pointer). Examples of natural forms of interaction and their corresponding devices are voice captured by microphone, electronic ink captured by tablets, touch captured by touchpads and touchscreens, and gestures captured by accelerometers and cameras.

Thus, our discussion in this chapter is not limited to traditional remote controls. However, we are also concerned with providing support for users of traditional remote controls, because it is still the main interaction device for currently available TV systems [Barrero et al., 2013].

There are several reasons to develop and use interfaces based on multiple input modes. One of them is to offer to users with disabilities alternative modes that best suit their needs, in order to democratize access and increase the number of potential users. As an example, Harada et al. [2009] are concerned with the development of a form of interaction through voice that brings benefits analogue to the possibility of direct manipulation allowed by the mouse.

In this chapter we present our model of a software component that allows users to enter text in iDTV applications by means of an interface based on multiple input modes—the component includes a virtual keyboard mode, a phone keypad mode, and a speech mode. We designed the component according with the user-centered design (UCD) methodology. We first carried out a research review with respect to text input methods used in TV systems. Next, we interviewed four experts in the iDTV domain. A third activity involved the application of questionnaires to 153 TV users, aiming at identifying a profile that corresponds to users who use text entry mechanisms.

Carrying on with the development of the prototype, we conducted usability tests using the think aloud protocol, and usability inspections using the heuristic evaluation and cognitive walkthrough techniques. The evaluations allowed the detection of a number of problems, which could be dealt with in intermediary versions. The evaluations also pointed to several opportunities of improvement on the design. In particular, they highlighted the importance of

using complementary text input modes in order to satisfy the needs of different users.

4.2 Related work on text input for TV

The use of interfaces based on multiple text input modes in digital television is the focus of this study. For better understanding the problem, we carried out a survey of methods used to input text in iDTV applications. Some of them are even older than the first applications for digital TV, as they are inherited from video game consoles with simple remote controls. In this group, we highlight two methods: virtual keyboard and sequential selection of characters. The first method consists in displaying on the TV screen a set of buttons representing characters. One of them has the focus, that can be moved using the arrow keys. The associated character can be entered using the OK / Enter key of the remote control. Two different layouts are typically used: alphabetic and QWERTY. Some studies [Go et al., 2008; Wilson and Agrawala, 2006] explore a QWERTY virtual keyboard with two foci controlled by joysticks equipped with two directional controls. The idea is to use the analogy of typing on a computer keyboard, where each hand is responsible for half of the keys.

Sequential selection of characters is a simpler method that consists in presenting to the user a character at the current cursor position: the character can be modified, in alphabetical order, using the arrows on the remote. After selecting a character, the cursor moves to next position and the user should once again go through the list of characters available for selection of the new character. This method avoids using the screen to display the virtual keyboard, but provides a less efficient text entry. Another group of text input methods that can be used in iDTV comes from the methods traditionally found in mobile phones, as traditional iDTV remote controls also have a numeric keypad. In this group, two methods stand out: multi-tap and T9 [Grover et al., 1998], a predictive text method. James and Reischel [2001] present performance metrics for each of them and show the efficiency of T9.

In a more recent study Barrero et al. [2013] investigated several input methods, classified into two categories: virtual keyboards and mobile-like methods. They extend previous work [Perrinet et al., 2011] by experimenting with more text entry methods and usage contexts. The tested layouts for vitual keyboard methods were QWERTY, alphabetic, a new layout generated with a genetic algorithm that positioned the most used letters in the center of the keyboard, and modified versions of the QWERTY and genetic layouts to enable typing special letters and symbols. Tested mobile-like methods were multi-tap, T9, 2-key—a method designed to allow typing any letter with two keystrokes, by choosing first a group of 3 or 4 letters then the desired letter [Silfverberg et al., 2000]—, and a modified version of multi-tap to facilitate typing special letters and symbols. They showed that T9 is the method with the best performance when writing simple texts, but leading to higher error rates. When writing complex texts, virtual keyboards lead to the same or better typing speeds than multi-tap (T9 was not tested), with significantly lower error rates. One of the recommendations the authors give to iTVD application developers

is to provide multiple methods for the users to choose from. Sporka et al. [2012] conducted a similar study comparing only two other mobile-like methods based on two keystrokes: The Numpad Typer (TNT) [Ingmarsson et al., 2004], which has groups of 9 characters, and TwiceTap, which has groups of 9 characters or n-grams and is similar to the 2-key method. The two methods achived equivalent typing rates, but TwiceTap led to less keystrokes per character and less errors.

Methods that use more recent technologies are also being investigated. de Miranda et al. [2008] conducted a detailed study on the challenges and guidelines that should be considered during the design and integration of new physical artifacts to the Brazilian context. According to the authors, the remote control used with analogue television, still prevalent, can act as a limiting factor for interaction with proposed and developed services for digital television. Three out of ten guidelines explicitly refer to the use of speech, recommending that interaction alternatives for people with physical disabilities, visually impaired and illiterate should be provided. However, the authors point out that not all environments can promote this form of interaction and that the system has to be trained to recognize the user speech. They also mention the problem of the collective use of TV and the noise and natural atmosphere in which TV is watched. Hence the importance of providing alternative ways to carry out a task.

In the work of Cox et al. [2008], the main objective was to explore the complementary aspects of the voice and the phone keypad to overcome the deficiencies of traditional methods of text input in circumstances of user mobility, where hands and vision are busy. Nakatoh et al. [2007] describe a speech recognition interface system for digital TV (DTV) control. An omnidirectional microphone is used due to the position variation of the speaker in relation to the microphone embedded on the remote control, which has an easy to grab format with a push-to-talk side button. The captured audio signal is sent to the TV equipment, where it is initially processed by a digital signal processor. Then, the recognized phonemes are passed to be processed by an automatic speech recognition system. The speech recognition system allows a command to mapped by several items of the phoneme dictionary. The name of a single channel, for example, can be pronounced in 6 different ways. In total, the dictionary has around 400 items—1300 items when the various ways of pronouncing the same command are considered. To improve the system usability, they developed a technique for reducing ambient noise and a technique of echo cancellation of the TV sound. They also state that the speaking style varies with the user generation and therefore they developed age-dependent acoustic models.

Currently, the most sold software for speech recognition is the Dragon NaturallSpeaking[2]. It has a large vocabulary, continuous speech recognition and reports a 99% accuracy. It requires that each user create a personal speech model by undergoing a 10-minute reading section of predefined texts. It was the 7th version of the software that was used in the experiments of the study from Cox et al. [2008].

Another study on commands by voice was performed by Wittenburg et al. [2006]. In their proposed system, users are free to pronounce any word, without vocabulary or grammar

[2] `http://www.nuance.com/dragon`, accessed 05/Sep/2014.

restrictions, and receive as output a list of possibly related programs. To help users understand the results, the authors propose as future work a variable highlight in the words of the result list. That is, words for which the system credits higher probability of been pronounced by the user gain greater prominence.

The latest commercial devices will help verify if the studies that use voice as a way to text entry in TVs, including this one, were in the right direction. Amazon Fire TV, released in April/2014 and Nexus Player[3], the first Android TV device (announced in October/2014), are equipped with a remote control that offers a push-to-talk button for program searches.

Research that use gestural input—using input devices (e.g. Wii Remote) or freehand motion tracked by a camera (e.g. Microsoft Kinect[4])—as a strategy for text input has also been conducted. Castellucci and MacKenzie [2008] present a technique for text input using the motion sensor equipped remote control of the Wii video game to capture gestures that are mapped to characters. An alphabet of gestures is proposed, in which every gesture is composed of only two primitive movements. Ren and O'Neill [2013] investigated freehand gestures captured by a Microsoft Kinect camera. Two virtual keyboard layouts and three selection techniques were evaluated. The best results were archived with the Dual-circle layout proposed by them—in which characters are arranged in two circles next to each other and accessed by the hands located in the center of the circles—and a selection technique that combines moving the hand in the 3D space with an expansion of the target.

4.3 Our design

The issues discussed here were considered during the development of the prototype of a mechanism for text input in iDTV applications. The study was conducted using a set of techniques that aim to engage the end user during all stages of development, known as User Centered Design [Dix et al., 2004]. First, we did a requirement elicitation with potential users and experts in the area. Then, the features and technical characteristics of the interface model were defined. Finally, we developed a prototype in order to conduct usability tests to validate the model.

4.3.1 Requirements Elicitation

Initially, we conducted a study to better understand the potential users of the mechanism. We interviewed four experts in the area, coming from different contexts, and we applied a questionnaire with potential users from various regions of Brazil. The main contributions are reported below.

[3] `http://www.google.com/nexus/player`, accessed 16/Oct/2014.
[4] `http://www.microsoft.com/en-us/kinectforwindows`, accessed 15/Oct/2014

Questionnaires

We applied the questionnaire on paper (32 responses), in order to reach a diverse audience, and using an on-line survey system (121 responses), which helped to obtain more responses. One of the questions asked what would be the best way to write a message to a friend using the television. The answers to this open question were very diverse and may be clustered into the following categories: 1) QWERTY keyboard, 2) T9 predictive text standard, 3) speech, 4) virtual keyboard using a simple remote control, 5) virtual keyboard using a touchscreen TV, 6) virtual keyboard using a touchscreen remote control, 7) thought, 8) pre-formulated phrases, 9) writing on a paper whose text is recognized by the television, 10) writing with a pen directly on television screen, and 11) using a mobile phone connected to the TV.

A concern regarding the need to offer more than one alternative to text input could be noticed, as in the following examples (our translation):

- "*Using speech when I'm alone and using T9 when I'm in an ambient with other people.*"
- "*Speech would be ideal, but I think we could correct some possible errors or [make some] modifications using a keyboard, for example.*"
- "*I would like to be able to choose: `i)` If I'm in the living room with my mother in law: conventional keyboard; `ii)` if I'm with other people: speech to text.*"
- "*Using speech and phonemes recognition or, to a reality closer to ours, a remote control with a LCD touchscreen...*"
- "*... but there must be other ways, that are accessible from those who can't speak or are currently without voice, for example.*"

Another recurrent concern was regarding to the interference of the environment and TV sound, in cases where the text input is performed using speech, as in the following examples (our translation):

- "*... I know there are limitations and difficulties, such as external interference from other people, background noise, etc.*"
- "*... but there may be noise problems.*"
- "*... without interfering with the audio of the program that the person is watching.*"

Interviews

In order to better understand the problem, we interviewed four experts in the area, each with different specialization. They were an university professor and researcher in the areas of multimedia, hypermedia, middleware and interactive applications for digital TV, an usability engineer of a research and software development company, and an interaction designer and a product consultant of a cable TV company. The main contributions are reported below.

The university professor thinks that the media use various auxiliary information that help in understanding the message. There are a number of ambiguities that are only broken because of

the context. He finds interesting the possibility of sending messages through TV and writing them using speech, but thinks that it could disrupt the environment. "*It should occur in a restricted environment.*" He says that "*there are several ways in which communication can be done: voice, body movement, interaction with objects, wind—blowing stronger or weaker can generate an alphabet, for instance.*" He considers necessary to define reserved voice commands in order to let the machine knows when to take certain actions instead of writing what is being spoken. He notes that young people prefer to communicate with text, even though the communication by voice are faster and more direct. "*... youths today are so good writing with the numeric keypad that perhaps other ways are not best suited to them ... Maybe it's because texts are more reserved*", he said.

The usability engineer has seen only one application that used text entry. It used a virtual keyboard. "*It was an alphabetic keyboard, not QWERTY. One of the letters had focus and the enter key was used to insert the letters focused.*" He considered feasible text entry by voice. "*It would require less effort and it is interesting because it is more natural*", he said."*The advantage would be to insert long texts. The disadvantages would be when people were watching TV together with someone, because of privacy, or when there were someone sleeping.*" He did not know any project for text entry by voice on TV.

The interaction designer and the products consultant have once "*analyzed several kinds of infrared remote controls. Some were of normal shape and others more complex-shaped, like a mouse, keyboard or joystick. Cost-benefit has led us to choose the simple remote control.*" Regarding the chosen model, *"the major difficulty is that there are no character keys in the remote control and a simulation has to be done. One way is by showing a virtual keyboard on the screen and allowing the user to navigate through the arrow keys and press OK."* They have tested QWERTY and alphabetic layouts. *"According to the performance tests, typing in the alphabetic keyboard had more success than QWERTY when the user was at are slightly larger distance from TV."* They think that *"the tendency is to use text input format similar to the phone keypad, where the letters are also printed on the keys."* They were also concerned about the user having to look at the screen and at the remote control to seek for the letters at the same time. *"Even when the remote controls have the letters printed, we will continue showing the keyboard map on screen."* When discussing the use of voice input, they mentioned that noise in the environment where the TV is located might hamper the recognition. Finally, they have also emphasized that *"the TV set has a special characteristic of being a familiar device and not individual."*

This step was crucial to help us understand the importance of offering alternative ways of entering text, in order to meet the needs of different user profiles and to be flexible enough to be used in different environments.

Requirements and usability criteria

Requirements elicitation allowed us to define the functional requirements and key criteria of usability to be considered. Nine functional requirements for the system were elicited, addressing

basic issues related to inserting and deleting characters and words, and the substitution of letters both using speech and remote control. The following usability criteria were considered the most important in the context of the study: (1) *Familiarity*: User should be able to use some of the knowledge it already has in the context of writing during his first interactions with the proposed mechanism, (2) *Substitutivity*: Complementary forms of text input must be provided for the user, and (3) *Responsiveness*: A iDTV decoder usually has little processing power compared to a personal computer. The system must be fast enough to let the user notice changes in its state.

4.3.2 Text entry model based on multiple input modes on iDTV

Our proposal aims to explore and to develop a mechanism for text entry in iDTV. We realized that no single mechanism is able to meet the needs and characteristics of all system users. The approach based on multiple input modes serves a larger number of users in a satisfactory manner. We designed the model considering three main methods of entering text:

1. Speech recognition with a microphone located on the user's remote control, which is activated via a push-to-talk key;
2. Phone keypad mode using the remote control, with the mapping between buttons and letters shown on the GUI. That method can be used with text prediction (T9) or without a dictionary aid (multitap);
3. Virtual keyboard mode in alphabetical order, in which the user navigates through the letters using the arrow keys and inserts the selected letter in the text using the OK key on the remote.

Our model allows the user input text using the mode that she chooses when she wants, as illustrated in Figure 4.1. We choose the virtual keyboard mode as the default state of the component to the detriment of the phone keypad mode because this mode can be more intuitive and comprehensive. Users not used to write text on cell phones face difficulties if they tried to write using this mechanism. As we noticed during the analysis of questionnaires, they were answered mostly by people with high levels of education (80% at the beginning of undergraduate college) and young adults (80% of people were between 18-44 years old). 30% of people said that seldom or almost never send messages via cell phone.

The user can use the speech mode concurrently with other text entry modes in a natural way by pressing the push-to-talk key on remote. Moreover, to switch between the "Virtual keyboard" and "Phone keypad" modes the user activates a button on the interface.

4.3.3 The prototype in use

Based on the requirements and principles defined on the model, two interfaces were independently designed, and the strengths of each were combined to create a third interface. Next, a functional prototype was implemented to allow the evaluation of the designed interface. It was created in

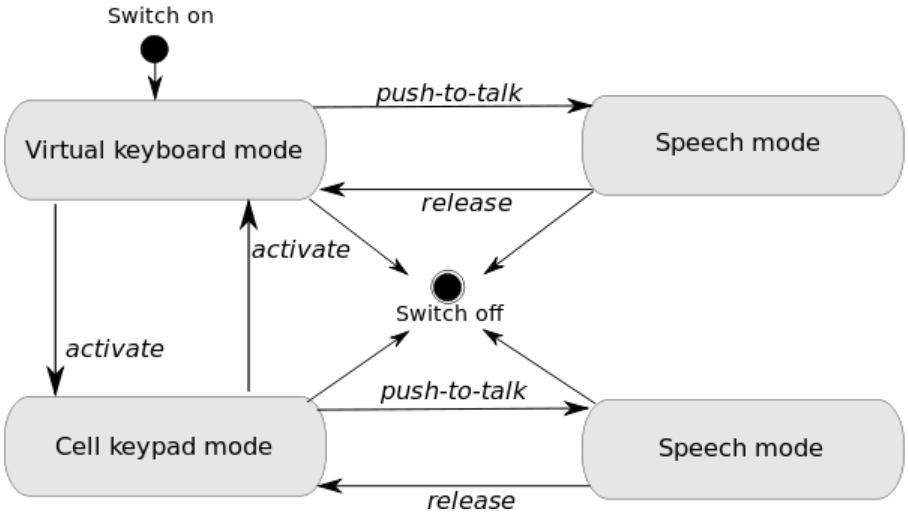


Figure 4.1: State diagram of the model.

Java to run on a PC, aiming a short development time. Figure 4.2 shows the four major states of the text input component prototype.

The speech mode would allow the user to write in the focused text box dictating the words she wants, but no speech recognition engine was implemented. This functionality was tested using the Wizard of Oz technique, which allows designers to test a limited functionality prototype by providing a missing functionality through human intervention [Dix et al., 2004]. In addition to the dictate state, shown in Figure 4.2 (top-left), the speech mode of the component has also a state where only editing commands are recognized by the speech recognition engine—although it is not available in the current version. The phone keypad mode (Figure 4.2 (top-right)) allows the user to write in the focused text box using the numeric keys of the remote, just as it is traditionally done in telephones and cell phones. It should allow both multi-tap and predictive text—although only the first mode is currently available. Finally, the virtual keyboard mode (Figure 4.2 (bottom-left)) allows writing text by selecting the desired characters using the arrows and OK key. By selecting the button on the bottom right of the keyboard, the function of the arrow keys on the remote control changes to let the user to move the cursor in the text box and, therefore, the buttons on the virtual keyboard become disabled, as shown in Figure 4.2 (bottom-right).

To go from the virtual keyboard mode to the phone keypad mode, users should press the directional keys to give focus to the "*Celular*" button, and press OK on the remote control (Figure 4.2 (bottom-left)). To switch in the opposite direction it is necessary only to press the OK key (Figure4.2 (top-right)). In case the user wants to activate the speech recognition system, she only needs to press and hold the push-to-talk key on the remote control. The interface of the component changes to show the commands that can be triggered while the user dictates the desired text. As soon as the push-to-talk key is released, the interface shows a mode according to the previous state.

Below we list some important features of the prototype:

Figure 4.2: Four main states of the text input component prototype (in Portuguese): Speech mode in dictate state (top-left), phone keypad mode (top-right), virtual keyboard mode (bottom-left) and virtual keyboard mode in cursor movement state (bottom-right).

1. The prototype was developed to be simple and to allow the tester to focus its attention in the text entry component only. In the screen, only a text box and the designed text entry component are shown.
2. The font sizes used in the prototype led to a component visible area of approximately 400 x 330 pixels, which fits even on a standard-definition television without causing visualization difficulties.
3. The names chosen to the three text entry modes needed to be small because of the low screen resolution adopted. The icons were also chosen in order to be easily identified and associated with the mode.
4. No help screen was developed because the text entry component should be described in the help screen of the application that uses it.

Finally, interaction with the real system should be performed with a special remote control containing a built-in microphone and a push-to-talk key that must be pressed to activate the speech recognition engine. In our user tests, data input was performed using an adapted computer keyboard where some keys were relabeled and unused keys were covered with adhesive paper to

Figure 4.3: Adapted keyboard to interact with the prototype.

not confuse the user, as shown in Figure 4.3.

4.4 Usability evaluation

We used two usability inspection methods for the evaluation of the prototype: heuristic evaluation and cognitive walkthrough. They have a strong acceptance and recognition [Hollingsed and Novick, 2007]. We also conducted user testing through the thinking aloud with Wizard of Oz technique. The think aloud tests were performed by 4 pairs of users, following one of the recommendations of Flores et al. (2008), who note that testing pairs allows individuals to express more naturally their actions and opinions. Figure 4.4 shows some frames taken from the recorded videos during the tests. The heuristic and cognitive walkthrough evaluations were applied each one to three experts and the heuristic evaluations were performed using the general heuristics proposed by Nielsen and Mölich [5].

Information from evaluations and tests were summarized in order to make a general compendium of problems in the proposed mechanism. Each of these problems was assigned a level of severity. Seven, due to greater severity, were used to create a list of change recommendations. The main problems were:

1. The speech mode has a different activation mechanism from the others, but this is not indicated in the interface. There was not a consensus among the evaluators if this mode should or not be activated only by clicking on the push-to-talk button, as it is now. However, if this happens, that difference must be clearly indicated in the interface.

[5]The original list of 9 heuristics from Nielsen and Mölich (1990) has been refined by Nielsen (`http://www.nngroup.com/articles/ten-usability-heuristics`, accessed 25/Sep/2014).

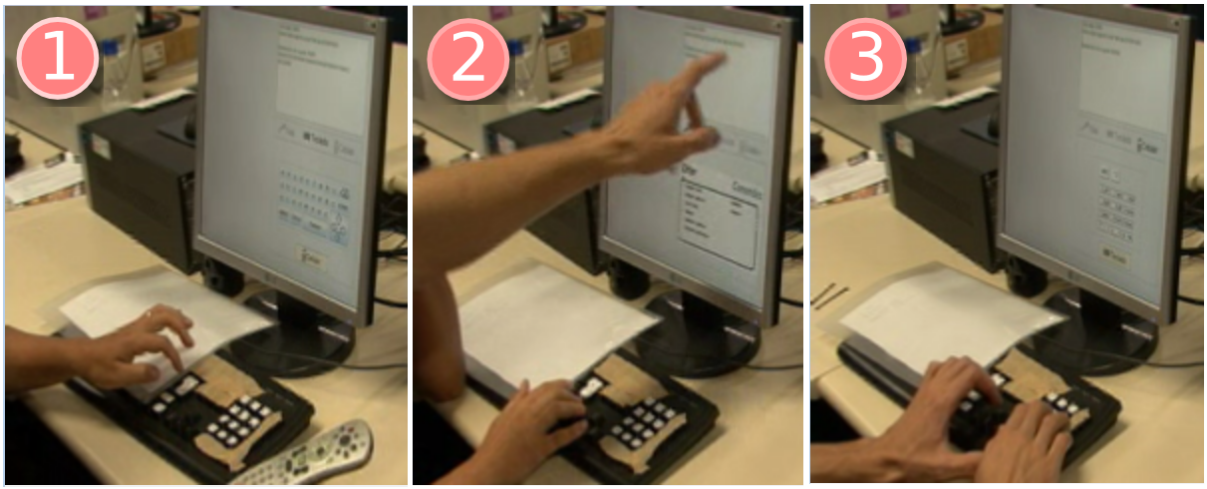


Figure 4.4: Frames taken from the recorded videos during the tests, in which the 3 different modes were used: (1) Virtual keyboard mode, (2) speech mode and (3) phone keypad mode.

2. The activation mechanism of the phone keypad mode and the virtual keyboard mode should be better crafted. The buttons that allow the activation are distant from the tabs that indicate which mode is active. It seems more intuitive to use the tabs themselves as the activation mechanism.
3. The phone keypad mode does not give a clear indication that the buttons are shown only to serve as an on-screen guide for the users. This causes several error situations, in which participants try to use the arrow keys to focus one of the illustrative buttons.
4. The term “Celular” (cell phone), used to indicate the text input mode often used in mobile devices, may not be easily understood by users, who may think it refers to their own mobile phone or to a phone call.
5. The BACK button caused a lot of frustration because it could almost never be used. This may have occurred because the component has been evaluated outside the context of an application, where it would certainly have clearer functionalities. Another problem caused by the key was the association made between the key and the backspace key of a traditional computer keyboard.
6. The key used to enter and to exit the cursor movement state, accessible from the keyboard mode, was not the same. The key to enter was the OK key, but the key to exit was the BACK key. This was extremely counterintuitive. The possibility of inserting new lines using the OK key, offered when this state is activated, is not worth the amount of errors generated.
7. The interface of the speech mode is not clear enough and causes a lot of problems. The use of two columns of commands in the dictate state makes the user associate the right column with the command state. The interface is not clear neither on how to access the command state, nor indicate clearly the difference between the dictate state and command state. Not even the list of available commands in the dictate state is satisfactory.

In addition to the problems identified in the design of the text input component, some serious problems on the specific prototype implementation ended up gaining more prominence in the evaluations than it was expected:

1. The cursor that indicates where the next character will be inserted is not shown in any of the input text modes.
2. The adjustment made on the numeric keypad resulted in the "Num Lock" key being used to insert symbols. This made the rest of the numeric keypad stop working whenever the symbols key was pressed an odd number of times.
3. The predictive text functionality (T9), despite not being implemented, is indicated by a label on the interface.
4. It would be hard to insert text into the text area of the prototype during tests of the speech mode using the Wizard of Oz technique. A parallel screen was used.

Observing the tests with users, we noticed that the think aloud test was not adequate when evaluating the speech mode, since users were asked to say all they were thinking, which does not fit well in the case of voice interfaces.

4.5 Possible improvements

In order to solve the main problems of the designed component, the following changes should be made:

1. To allow the three text input modes to be activated by selecting the corresponding tab. Thus, the "*Celular*" ("Cell phone") and "Teclado" ("Keyboard") buttons are removed from the interface. When the virtual keyboard mode is activated, the focus can move freely between the character buttons and the two other tabs. When the speech or phone keypad mode are activated, the focus can move only between the two other tabs. The button push-to-talk should be preserved and the user should be able to press it regardless of the active mode, which makes it a shortcut to the speech mode. However, if it is pressed while the speech mode is not active, when released, the previous mode should be activated again. In addition, to increase the clarity and to consider users who do not speak English, it should be relabeled to "*Segure para falar*" ("Hold to talk"). These changes aim to solve problems 1 and 2, and also have some impact on problem 3, as it does not let the focus fixed on a single button while the phone keypad mode is active.
2. The graphical interface of the phone keypad mode should be improved so that no doubt remains that the buttons shown are merely illustrative. This change also aims to solve problem 3.
3. The tabs that identify the text input modes should be bigger to allow each mode be identified by more than one word. The phone keypad mode would be called "*Estilo celular*" ("Cell phone style"), the virtual keyboard mode would be called "*Teclado virtual*" ("Virtual

keyboard"). The name of the speech mode could remain the same. This aims to eliminate possible confusion explained by problem 4.

4. The BACK key should be relabeled to "*VOLTAR*" ("back") to consider users who do not speak English and also solve problem 5. If time is available, the context of the application should be used in the prototype, to make it clear what is the function assigned to the BACK key.
5. The OK button should also be used to exit the cursor movement state. This would solve problem 6.
6. The organization of the items shown in the speech mode should be improved, to make it clear to the user that "*Ditar*" ("Dictate") is just one of the possible states of this mode. The command "*Comandos*" ("Commands") should be listed along with other possible commands of the dictate state. Also, the command "*Ditar*" ("Dictate") should be listed along with other possible commands of the command state. The "*Apagar linha*" ("delete line") command should be added to the list of commands of the dictate state. The commands that require some extra word, as in the case of "*símbolo*" ("Symbol") and "*número*" ("number"), should indicate that in the interface. All these changes address the problems related to the speech mode, and are grouped in item 7.

4.6 Summary and future directions

Given the differences between the iDTV and PC platforms with respect to user tasks associated with viewing, navigating, and interacting, in this chapter we propose an interface model based on multiple input modes to deal with the problem of text entry in iDTV applications, and present a prototype that implements the model.

We conducted usability tests using the think aloud protocol, and usability inspections using the heuristic evaluation and cognitive walkthrough techniques. The evaluations allowed the detection of both a number of problems and several improvement opportunities; at the same time they highlighted the importance of using complementary modes of text input in order to satisfy different user needs. Overall, the evaluation results suggest that the proposed approach provides a satisfactory level of usability by overcoming the limitations of text input in the context of iDTV applications.

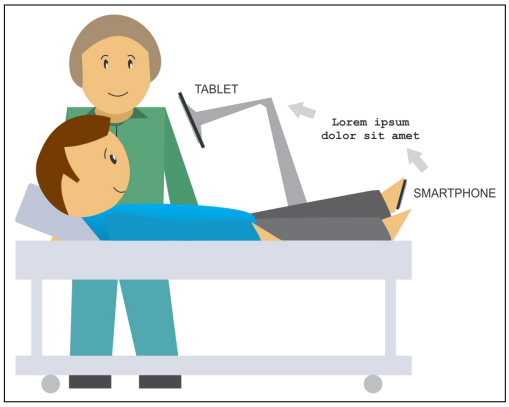
With respect to future work, the problems and new requirements identified by the specialists in usability should be tackled. A new prototype, including the suggested changes, should be implemented to run on a set-top box, so that new evaluations may be carried out. This is important since it allows to take into account factors which have been ignored in the current evaluations, in particular, the user performance when a real remote control is used.

The importance of text entry in our lives is undeniable. The research effort presented in this chapter is also justified by the rise of the interactive television market. However, some populations still demand improvements in text entry methods for regular computers. That is the

case of individuals with severe motor disabilities, who depend on text input for communicating. We turn our attention and effort to minimize the challenges they face in the next two chapters.

Chapter 5

DuoGrapher and Swinging Foot: Text entry using a foot



Some individuals affected by a motor neuron disease lose the ability to speak and the voluntary movement of the torso and arms, but keep for some time at least a partial movement of a leg and a foot. This chapter[1] presents DuoGrapher, a text entry method, and SwingingFoot, an interaction technique, which aim to improve their communication skills by leveraging these motor capabilities. The idea is to attach in one of the user's feet an accelerometer-equipped device, which detect and transmit the movements to a second device located in front of the users' eyes. DuoGrapher interpreted the movements as characters according to the codification being used. Our design is informed by the motor capabilities of a male, in his 60s, with a motor neuron disease. We developed a prototype and tested it with 15 volunteers without disabilities in order to test the usability of the system before doing further studies with motor-impaired individuals. Our goal is understand the challenges of a foot-based system interaction, which may be explored in subsequent work for many other purposes, including controlling an interactive TV system. It became evident that a semi-automatic calibration mechanism and a simpler mapping from movements to character are needed in order to reduce errors.

[1]This chapter is based on the following paper:

- **D. Pedrosa** and M. G. C. Pimentel. Text entry using a foot for severely motor-impaired individuals. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, SAC '14, pages 957-963, New York, NY, USA, 2014. ACM. doi: 10.1145/2554850.2554948. URL http://doi.acm.org/10.1145/2554850.2554948.

5.1 The importance of text entry to individuals with severe motor disabilities

Writing is an important skill in our lives. Some people can not spend a day without writing anything, even if only a couple of keywords in a search engine. People who have been affected by a motor neuron disease or any disorder that affects voluntary muscle activity may be deprived of this routine means of communication. The motivation of the several research projects that have been conducted, including this, is to provide text entry interaction techniques for those with severe motor disabilities.

Depending on the severity of the disability, different techniques may be used. In one extreme, a motor-enable individual usually writes using a regular QWERTY keyboard. Someone using only one hand may adopt a left-hand or right-hand Dvorak keyboard layouts[2] or a chorded keyboard, like Twiddler[3]. In the other extreme, a severe disability may lead to the use of a single switch text entry technique.

This chapter presents SwingingFoot and DuoGrapher, an interaction technique and a text entry method designed for people with a severe motor disability which prevents the movement of the torso and arms but, for some time, keep partial movement of a leg and a foot. Our contribution fills an important gap, as we believe there is no text entry technique that leverage the capabilities of our target population.

The DuoGrapher method is based on Morse Code [International Telecommunication Union, 2009]: two different symbols are combined to build codes that represent letters, numbers, punctuation and special characters. The basic operation of the SwingingFoot technique consists of attaching an accelerometer-equipped device in one of the user's feet and using the sensor to detect internal and external heel rotation, which are interpreted by DuoGrapher as dots and dashes, respectively. The symbols are transmitted to a tablet or any equivalent device located in front of the users' eyes. The sequence of symbols are transformed into characters according to the code being used.

Scott et al. [2010] studied the human capability associated with performing foot-based interactions which involve lifting and rotation of the foot. Participants identified heel rotation as the most comfortable gesture, with external rotation preferred over internal rotation. Although focusing on a standing position, their results also support our gesture choice.

Our design is informed by the motor capabilities of a male, in his 60s, with a motor neuron disease. Our first contact with him and his family was used to understand his needs and to specify the requirements for a prototype. In a second meeting, we were able to test a working prototype with him and understand the adjustments that had to be done. A second version of the prototype were developed and tested with 15 able-bodied users in order to establish a baseline that allows comparisons with other methods and to easily detect further potential improvements.

[2] `http://www.microsoft.com/enable/products/altkeyboard.aspx`, accessed 25/Sep/2014.
[3] `http://twiddler.tekgear.com`, accessed 25/Sep/2014.

5.2 Related work on assistive technologies for text input

While analyzing text entry methods, there are important concepts that should be observed, such as the amount and the precision of movement that it requires and its dependency on a dictionary.

5.2.1 Required movements

A regular QWERTY keyboard assumes a precise movement of all fingers of both hands. Text entry methods for people with motor disabilities have to support a limited range of movements. In the EdgeWrite method [Wobbrock and Myers, 2006], the user has to be able to use a trackball to move the mouse cursor to eight different directions, in order to produce letters using a Roman-like unistroke alphabet.

Some other methods impose less demand to the user. MDITIM [Isokoski and Raisamo, 2000], LURD-Writer [Felzer and Nordmann, 2006], and H4-Writer [Castellucci and MacKenzie, 2013; MacKenzie et al., 2011] are methods that require four different "symbols", which may be entered using different devices, such as a keyboard, a joystick, a mouse, a trackball, or a touchpad. A detailed comparison among these methods is presented by MacKenzie et al. [2011].

The most minimalist techniques assume the use of a single switch, which can be activated by a button [Belatar and Poirier, 2008; Leung et al., 2010], a headband sensor [Mackenzie and Felzer, 2010], or any intentional muscle contraction. These techniques are based on the principle of a scanning keyboard, in which letters or groups of letters are highlighted sequentially and the entry is made using a series of selections to narrow in on and select the desired letter [Mackenzie and Felzer, 2010]. The scanning pattern can be linear or grouped, as in the common row-column configuration, modeled by Simpson [2011].

In order to leverage the feet movements of our target user, as described in the next section, we opted for a less restrictive technique, in which two switches are required[4]. The internal and external rotation of the foot and leg, using the heel as the pivot, are sufficient to generate dots and dashes, the symbols that compose the characters in Morse Code. Other works have also been based on Morse Code, but use different input techniques, for instance, recognition of the open/close status of lips [Chen et al., 2008] and electro-oculogram (EOG) recognition [Wu et al., 2013].

5.2.2 Ambiguity / predictability

Methods that have an unequivocal mapping from user action to character, such as a regular QWERTY keyboard or the multi-tap in the numeric phone pad, are known to be non-ambiguous. In an ambiguous method, on the other hand, a series of key presses may result in different characters or words, as in the T9 method or in the one-line keyboard [Li et al., 2011]. The disadvantage of ambiguous methods is that its operation depends on a dictionary, and writing

[4] In fact, we use a third type of movement to provide backspace, but it can be eliminated, as discussed later on.

words out of the dictionary is usually harder. It also imposes higher cognitive load, as the user needs feedback to determine if the correct word was entered. The presence of a dictionary, however, opens the possibility to offer word completion or prediction, which usually allows for higher text entry rates. The text entry technique presented in the next chapter uses a dictionary to suggest candidate words based on the sequence of letters gazed by the user.

Examples of ambiguous and predictive methods are HandiGlyph [Belatar and Poirier, 2008] and SAK [Mackenzie and Felzer, 2010]. The method presented in this chapter is a non-ambiguous example. Other examples are EdgeWrite [Wobbrock and Myers, 2006], MDITIM [Isokoski and Raisamo, 2000], H4-Writer [Castellucci and MacKenzie, 2013; MacKenzie et al., 2011], and Character Stroke Disambiguation [Leung et al., 2010]. LURD-Writer [Felzer and Nordmann, 2006] is a non-ambiguous method that offers word completion.

5.3 Design process

Our target user, who we call by the anonymized name John in this chapter, was diagnosed with a motor neuron disease in the first semester of 2011. Motor neuron diseases (MNDs) are a group of progressive neurological disorders which affect the cells that control voluntary muscle activity such as speaking, walking, breathing, and swallowing [National Institute of Neurological Disorders and Stroke, 2012]. The most common of the motor neuron diseases is amyotrophic lateral sclerosis (ALS) [Medical News Today, 2009].

Some individuals with a motor neuron disease exhibit the Man-in-the-Barrel Syndrome—observable symptoms are paresis (semi-paralysis) and bilateral symmetrical atrophy of the muscles of the upper limbs [Orsini et al., 2009], and relatively normal mobility of face and legs [Fissi, 2009]. This was the case for John.

One year after the onset, he could not move his arms and head and lost the balance of the torso and the strength of the feet. He underwent a tracheostomy, which left him in bed since then. In our first contact with him, he could not talk or lift his legs by himself anymore and started complaining about fatigue while moving the lips. The combination of loss of upper limb movements with loss of speech, dramatically compromises these patients' ability to communicate. At that point, the two means of communication used by him were to move his lips as he was speaking so that people could read it, and to point with the big toe, with the help of someone holding his leg, to a 21" laminated QWERTY poster. John spends most of his day sitting or lying in a hospital bed that was installed in the living room of his apartment. His cognitive function were not affected.

In our first contact with John, we noticed his desire to demonstrate the movements he could still perform with the feet. We decided to base our technique on them, bearing in mind that we should be conservative due to the progressive nature of his disease.

In our second meeting, 40 days later, we brought him a preliminary version of a prototype. It was a great opportunity to verify if the prototype was appropriate for his physical capability

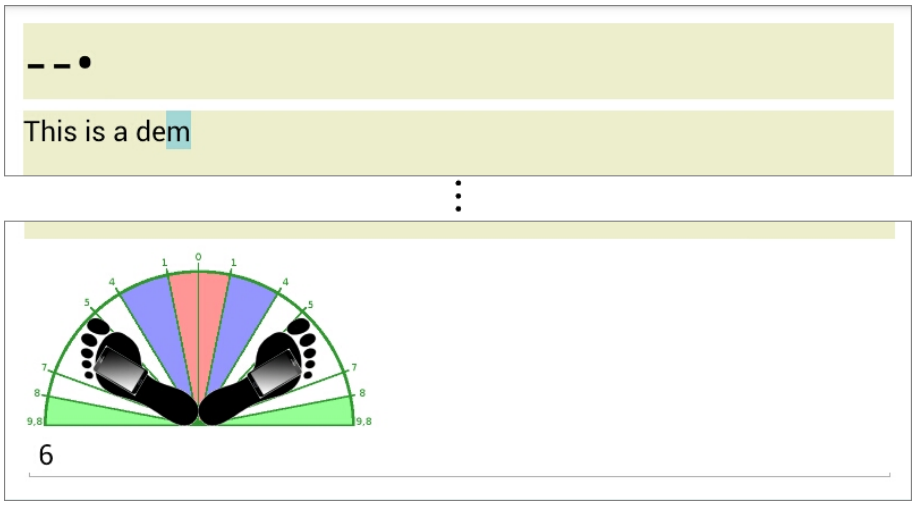


Figure 5.1: Prototype graphical interface. The bottom left part of the interface shows a reference image for heel rotation intensity needed to enter a dash (green), a dot (blue), or to erase the last symbol or character entered (pink). Movements of a left-footed users and a right-footed user are symmetric.

and to test important practical issues, such as different ways of attaching the smartphone to his feet. We were able to get feedback and suggestions from him, such as the need to offer a more flexible configuration related to the triggering inclinations, a broad range of alarms, and a more appropriate way for other people to follow what he was writing. Some of his requests could be achieved within the 2 days that we had before meeting him again. John was able to try the different input modes offered. These two days of testings and participatory design were very encouraging. John's wife said that this was the most valuable work she has ever received.

5.4 SwingingFoot and DuoGrapher

This section presents SwingingFoot and DuoGrapher along with the implemented system prototype[5]. A smartphone is attached to a foot using one or more rubber bands. When in the relaxed position the feet rest a bit rotated, as indicated in the reference image for heel rotation shown in the bottom left part of Figure 5.1.

An internal heel rotation, as shown in Figure 5.2 (left), followed by the return of the foot to the relaxed position triggers the input of a dot. An external heel rotation, as shown in Figure 5.2 (right), followed by the return of the foot to the relaxed position triggers the input of a dash. The blue and green areas of the reference image represent an internal and external rotation, respectively. A wider internal rotation, bringing the feet to the pink area of the reference image, followed by the return of the foot to the relaxed position deletes the last inserted symbol (if there is one) or character. Requiring the user to return the foot to the relaxed position before triggering an insertion or deletion is analogue to the use of a key release while handling a key event. This is

[5]The prototype is available at `https://play.google.com/store/apps/details?id=br.usp.icmc.bluemorsetext`

Figure 5.2: Two required types of movement: internal heel rotation (left) and external heel rotation (right).

necessary in order to differentiate blue and pink movements.

DuoGrapher currently support three different codifications. The first uses a Huffman coding created by the authors taking into account the frequency of occurrence of letters in Portuguese [Tkotz, 2005]. We sought to decrease the amount of movements required for Portuguese speakers, as John. However, being prefix codes, they are too long. We then opted for creating a codification based on the Morse ITU standard [International Telecommunication Union, 2009], including some adjustments mainly to allow pauses between characters of the same word. A code was assigned to the space character and an inactivity timeout is used to trigger the conversion of the current sequence of symbols into a character or command. The default value used for timeout is two seconds. The last codification uses the same principles of the Morse-based one, but replaces the set of codes by one that takes into account the same frequencies of letters used in the Huffman codification. This is the only codification detailed in the paper, as it was the one used by John and in the experiment explained in the next section. Figure 5.3 shows the code of all supported characters and commands.

An important metric used to analyze text entry methods is keystrokes per character (KSPC) [MacKenzie et al., 2011]. In our case, heel rotations per character can be seen as an equivalent metric. Considering the same frequencies of letters in Portuguese used to create the codification and that the average length of words in Portuguese is 4.53 letters [Tkotz, 2005], our method has KSPC = 2.18. MacKenzie et al. [2011] argue that methods that depend on a timeout should consider it as an extra keystroke. If we do this, our KSPC would go to 3.18. Other metrics analyzed in this chapter that depend on KSPC does not change significantly, because the keystrokes added in the formulas compensate each other. In the remainder of this chapter we use the 2.18 value.

As the smartphone screen is out of the user's eyes reach, the graphical interface of the prototype (see Figure 5.1) is shown in a tablet placed in front of the user. All movements detected by the smartphone are transmitted to the tablet over Bluetooth.

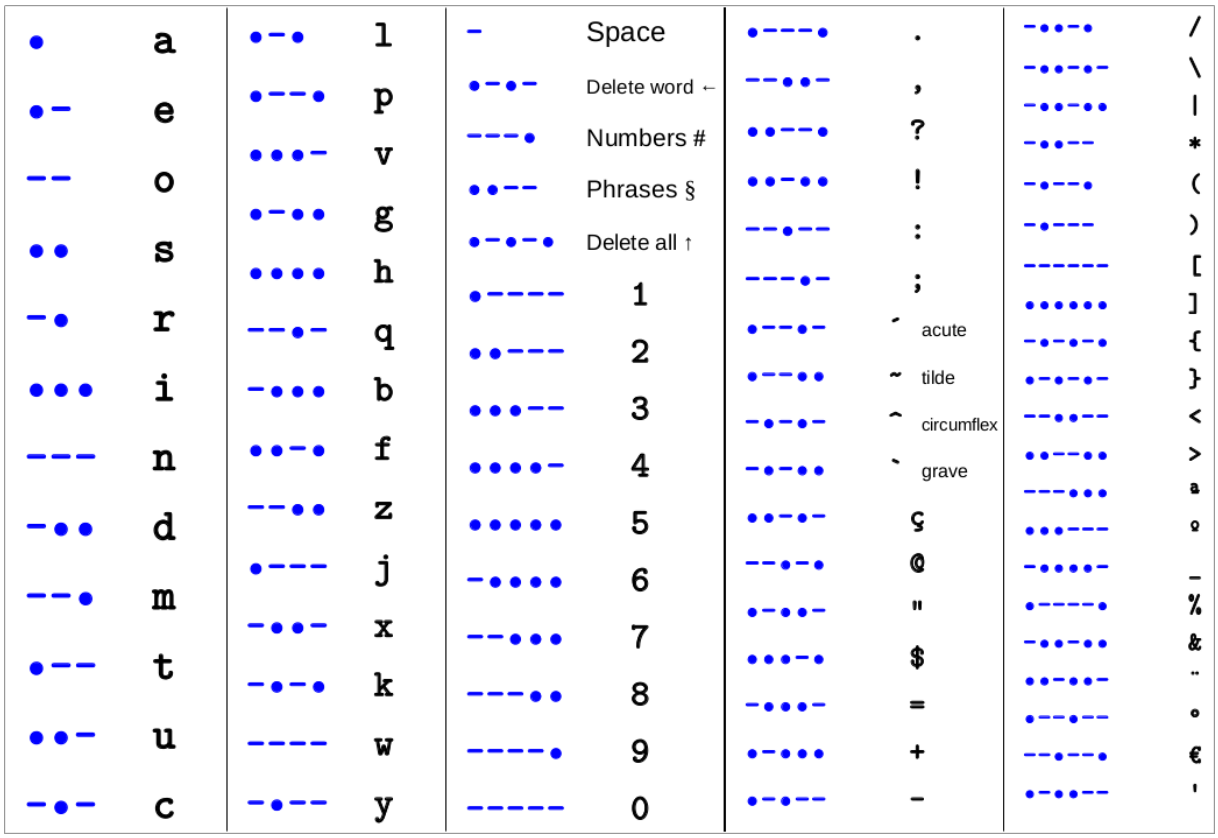


Figure 5.3: Poster with letters sorted by the frequency of use in Portuguese (translated).

Both visual and tactile feedback are given while the user enters text. Whenever a rotation movement is detected, its corresponding color is used as background (see Figure 5.4). Whenever a symbol is inserted or deleted a tactile feedback is given by vibrating the smartphone for 50 milliseconds. Whenever a character or command is entered tactile feedback is given by vibrating the smartphone for 100 ms.

In addition to the general purpose input mode explained, DuoGrapher also offers a Numbers Mode and a Phrases Mode. The Numbers Mode, accessed via code - - -•, may be used in cases where several digits need to be entered, such as when writing a telephone number. In this mode, each digit can be written with one up to three symbols. The Phrases Mode, accessed by code ••- -, offers an easy way to enter commonly used phrases. The current version offers John's 44 common phrases, such as "Increase the volume of the TV", for instance. In this mode, each phrase can be written with one up to five symbols.

5.5 Experiment procedures

We conducted an experiment with 15 volunteers without disabilities in order to understand the usability of the system. The experiment was very important to reveal potential improvements, which should be implemented before doing further studies with motor-impaired individuals. John exhibited a full range of heel rotation, which encourage us to conduct this study. We also took the

Figure 5.4: Visual feedbacks: just before entering a dash (left), just before entering a dot (center), and just before erasing a symbol (right).

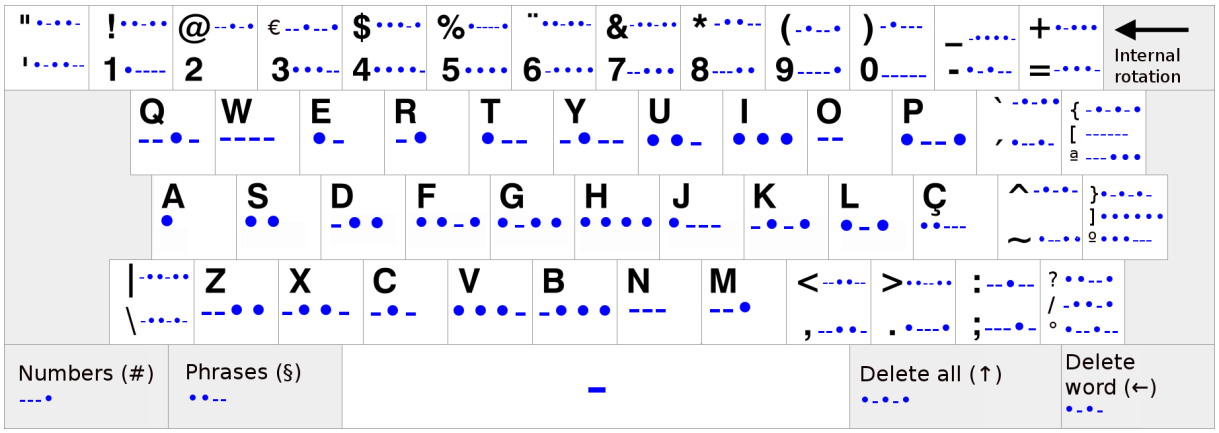


Figure 5.5: Poster with letters and special characters in a QWERTY layout (translated).

opportunity to compare 3 different kinds of code mapping posters to be used with the prototype: letters sorted by the frequency of use in Portuguese (Figure 5.3), letters sorted alphabetically (which is very similar to the previous one), and letters and special characters in a QWERTY layout (Figure 5.5).

We used a between-subjects design, randomly assigning a poster layout for each participant. Participants were 28.2 years old in average ($std = 3.26, min = 3, max = 33$) and all have extensive experience with QWERTY keyboards. All participants were right-footed and used only the right foot during the experiment. We used a Motorola Defy smartphone (118 g, 3.7" display size) on their foot and a Motorola XOOM tablet (730 g, 10.1" display size) as the graphical interface.

Each section lasted about 2 hours and were moderated by the author. With participant's consent, the prototype registered all interactions in a log file and sessions were recorded by 2 cameras (as shown in Figure 5.6) for an eventual posterior analysis.

Figure 5.6: Images captured by the cameras during a experiment.

The experiment was divided in 3 blocks of 5 phrases in each, with 10 minutes interval between blocks. Participants were told that the first block would be for training. No participant had any previous experience with the method, which were introduced to them before the first block. They were not given the opportunity to practice.

A post-test questionnaire was applied. During the two intervals participants were invited to walk. In the middle of the intervals and after the post-test questionnaire, participants were checked, without previous notice, regarding the memorization of the codes of 14 letters, ten of which among the eleven most frequent.

Our dataset is based on mem5, one of the 5 sets created and made available by Vertanen and Kristensson containing memorable text obtained from emails written by Enron employees on their BlackBerry mobile devices [Vertanen and Kristensson, 2011]. We first translated to Portuguese the 40 phrases of the set. Then, we selected the 15 phrases that sounded most natural in Portuguese and contained no number or special character (only letters and a dot or question mark). We adapted some of the phrases in minor ways –by changing from male to female, for instance– to obtain a higher correlation between the frequency of the letters in the set and in the Portuguese language. By doing this, we have increased the correlation from 0.962 to 0.979. Phrases have an average of 27.5 characters ($std = 12.12, min = 13, max = 52$). The order of the phrases was randomly selected during the pilot test and kept constant for all participants.

Participants were instructed to enter the phrases as quickly and correctly as possible and to ignore case (to use lowercase only). They had to press a button on the tablet in order to start and finish each phrase. Phrases were shown only after pressing the start button and remained visible until the finish button was pressed.

5.6 Results

We analyze the performance of participants in terms of text entry rate and rotation efficiency. We also analyze answers to subjective questions of the post-test questionnaire.

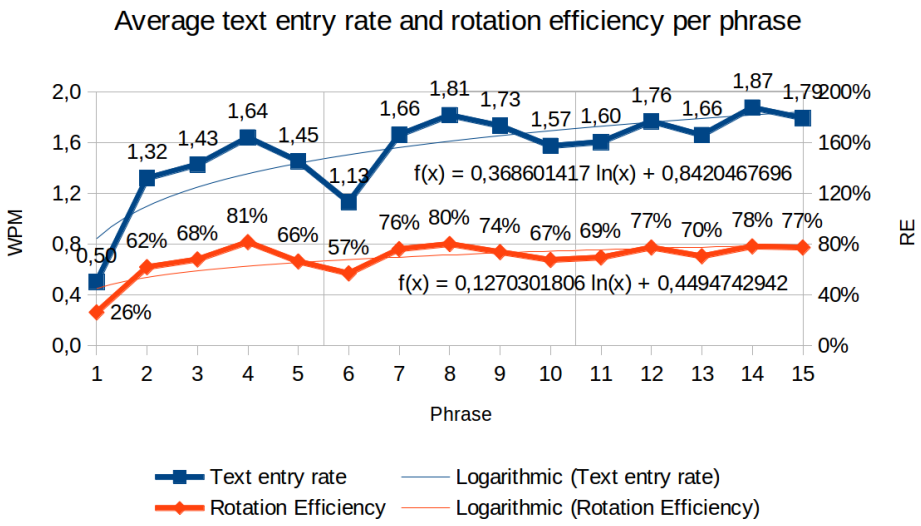


Figure 5.7: Average text entry rate and rotation efficiency per phrase, and logarithmic regression model equations that represent them.

5.6.1 Text entry rate

To obtain a text entry rate in words per minute (WPM), if we simply divide the number of words[6] in the phrase by the time in minutes that the participant took to enter the phrase, we might be misinterpreting his actual performance. The reason is that in DuoGrapher different characters may require a different number of heel rotations. Thus, a phrase formed by less frequent characters, may require more rotations than a lengthier phrase with very frequent characters. To give an example, the fourth phrase of the experiment is 35 characters long and require 70 heel rotations, a rate of 2 rotations per character. The eleventh phrase is 15 characters long and require 38 heel rotations, a rate of 2.53 rotations per character (26.5% more!).

To overcome this problem, to calculate the rate in WPM of a given user for a given phrase, we first divide the number of rotations the phrase requires by the time in minutes the participant took to write it. Then, we divide it by the KSPC of the method (2.18 rotations per character) to get a rate in characters per minute. Finally, we divide it by 5 to get the rate in WPM. The blue squares of Figure 5.7 represent the average text entry rate per phrase.

The text entry rates of the first and sixth phrases are clearly worse than of the others phrases in the same block. An independent-samples one-tailed t-test was conducted to compare text entry rate for the first phrase of each block and for the remaining phrases. There was a highly significant difference in the entry rate for the first phrases ($M = 1.08, SD = .55, n = 3$) and for the remaining phrases ($M = 1.64, SD = .17, n = 12$); $t(13) = 3.28, p = .003$. It suggests that the first phrase of each block should be considered as training. The average text entry rate of the four remaining phrases of each block are 1.37, 1.67 and 1.70 respectively. Figure 5.8 shows in blue the average text entry rate of the third block (last four phrases) per participant.

[6]Traditionally, when analyzing text entry rate in words per minute, a word is defined as any five characters [Mackenzie and Felzer, 2010].

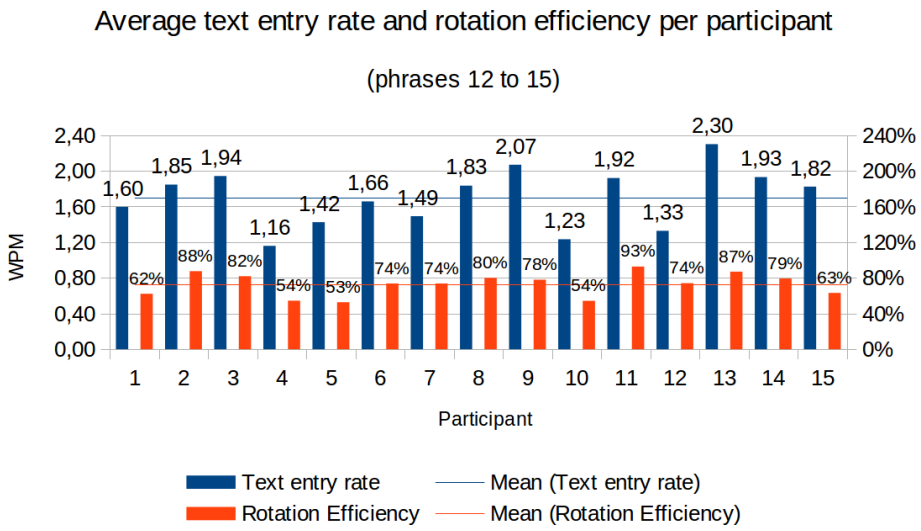


Figure 5.8: Average text entry rate and rotation efficiency per participant with mean value lines.

5.6.2 Rotation efficiency

Instead of analyzing the number of uncorrected errors, which is low[7] and does not reflect the total amount of errors committed by the user, we adapted the scanning efficiency (SE) metric introduced by Mackenzie and Felzer [2010]. In their case, it captures user performance while using scanning keyboards. In our case, the rotation efficiency (RE), as we call it, gives an indication of the user efficiency having as reference the minimum number of heel rotations required to write a given phrase:

$$RE = \frac{rot_{MIN}}{rot_{USER}} \times 100\% \tag{5.1}$$

As less errors are committed, less rotations are needed, performance improves and RE increases toward 100%.

The red diamonds of Figure 5.7 represent the average rotation efficiency of each phrase. The average RE of each block (four phrases each) are 64.39%, 72.82% and 72.92% respectively. Figure 5.8 shows in red the average RE of the third block (last four phrases) per participant.

5.6.3 Subjective data, memorization of codes and poster layout

The post-test questionnaire contained a five-point Likert scale with sentences about how easy and fast was to input text, to correct errors and to memorize codes. Figure 5.9 shows the level of agreement and disagreement specified by participants regarding the five investigated items.

The questionnaire also asked about which type of rotation was more difficult to perform. Eleven participants marked the external heel rotation as being more difficult, while only two indicated the internal rotation. The remaining two participants considered both as having the same difficulty.

[7] MSD error rate = 0.4% for last block, as defined by Soukoreff and MacKenzie [2003]

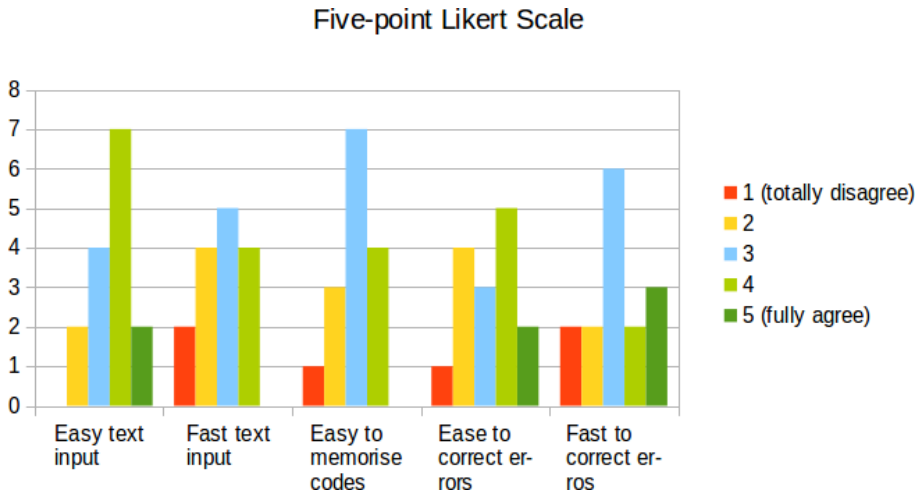


Figure 5.9: Level of agreement to the items of a five-point Likert scale about how easy and fast was to input text, to correct errors and to memorize codes.

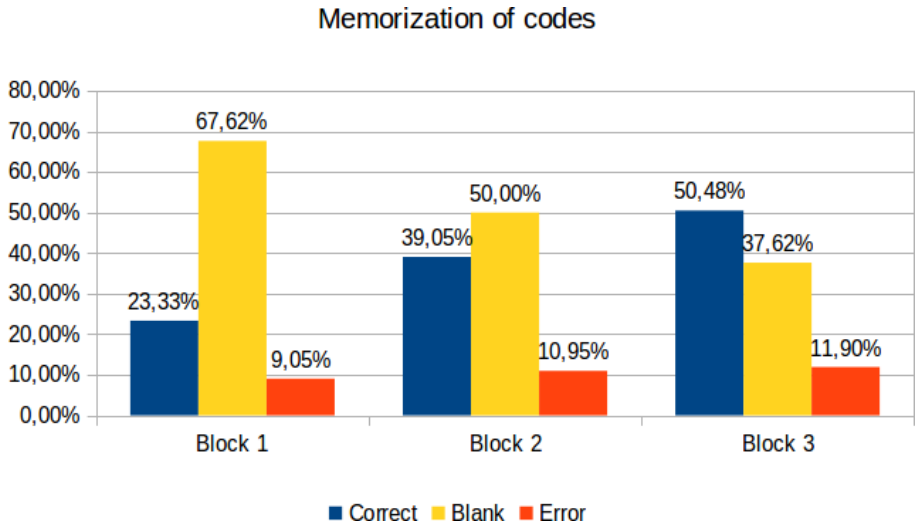


Figure 5.10: Percentage of letters whose code was correctly/not/erroneously indicated.

The memorization check clearly indicate an improvement over time. After the third block, users have memorized the code of about half of the letters that were checked. Figure 5.10 shows, for each block, the percentage of letters whose code was correctly indicated, whose code was not indicated, and whose code was erroneously indicated.

The layout of the codification poster had no statistically significant impact on the text entry rate (as determined by one-way ANOVA ($F_{2,42} = .46, p = .63$)) or the rotation efficiency ($F_{2,42} = .27, p = .76$). This may be partially due to the fact that the experiment was not specifically designed to compare the different layouts. In the post-test questionnaire, the participants were shown the other two posters and were asked to indicate which of the three they think allows a better performance. The alphabetic layout was chosen by 7 out of the 15 participants, while the frequency layout and the QWERTY layout were chosen from 4 participants each. Table 1 shows a preference matrix regarding poster layout.

Table 5.1: Preference matrix regarding poster layout.

Used \ Preferred	QWERTY	Alphabetic	Frequency
QWERTY	3	1	1
Alphabetic	0	3	2
Frequency	1	3	1

5.7 Limitations, analysis of the results, and possible improvements

Running an experiment with several individuals without disabilities was important for understanding that different people have different leg rest postures. Although the prototype has the option to configure the angle used to trigger the symbols and the backspace action, the same configuration was imposed to all participants. It was set according the author's leg rest posture, which turned out to be more opened/laid than average. That is the reason why the great majority of the participants expressed difficulty inserting dashes. A semi-automatic calibration based on a simple button press after attaching the phone to the foot would help a lot and should be implemented.

The rotation preference (11 x 2 x 2) become even more relevant knowing that some of the participants have complained about the difficulty of differentiating the blue rotation from the pink rotation. It was easy to notice during the experiment that many of the errors committed were related to using the backspace instead of inserting a dot. Nevertheless, this problem was not considered more important than the physical discomfort caused by the wide opening of the leg required by the green rotation.

The high correlation ($r(13) = 0,969, p < .001$) between the average text entry rate and the average rotation efficiency is a strong indication that the amount of errors has an important influence on the low speed obtained. To improve the rotation efficiency, besides implementing the semi-automatic calibration, we should unify the two types of internal heel rotation. As the backspace is probably more frequently used than the 'a', it should be performed with a single internal heel rotation. The code of letter 'a' would have to have two symbols. Consequently, letters 'r' and 'l' would also gain one symbol each. The mean code length, or KSPC, would increase from 2.18 to 2.38, but it would make DuoGrapher much more flexible as it would require in fact only two movements instead of three. It would be easier to apply it with different interaction techniques—using the thong, eyes or brainwaves, for instance—and, consequently, would support a wider variety of physical disabilities.

Another source of error was the lack of intuitiveness in the mappings from dots ands dashes to internal and external rotations or blue and green areas. Users were overwhelmed by the amount of mappings they should perform while writing. Sometimes, even knowing the symbol they wanted to insert, they did not guess quickly enough the kind of rotation to perform. We plan to eliminate the use of dots and dashes and start using blue and green arrows to represent the

codes. It may also help participants to memorize the codes, which is important, as the sentence "*The codification of the letters is easy to memorize*" was the second worst evaluated item of the Likert scale, beating only the sentence "*The prototype allows you to enter text quickly*". The number of participants that have memorized the code of a letter and the mean time participants took to enter it are strongly correlated, $r(12) = -.878, p < .001$. Thus, making the code of the letters easier to memorize may also increase the entry rate.

The participants were between 23 and 33 years old. We acknowledge that the problems they faced might be potentialized if we recruited individulas between 40 and 70 years old, the mean age for a ALS onset, since younger people can more easily adapt to new techniques.

If we suppose an ideal scenario in which users commit no errors, having a rotation efficiency of 100%, a simple rule of three gives us a text entry rate of 2.33 WPM. We can use the collected data to calculate an estimate for the upper bound for the text entry rate while using the prototype with a timeout of 2 seconds. If each character were always entered in the smallest time the participants took to enter it (for instance, 2020 ms for the 'a' and 2108 ms for the 'e'), and if we consider the same frequencies of letters already used, the text entry rate would be 4.23 WPM. This is the "upper bound" for the entry rate using the 2 seconds timeout. It should be emphasized that if we simplify the mapping of movements into symbols and enable users to commit fewer erros, we will be able to use a smaller timeout, leading to higher entry rates.

MacKenzie et al. [2011] obtained 20.4 WPM in a experiment with H4-Writer , but their participants used a thumb to press buttons in a joystick. Testing the same H4-Writer method, which uses 4 symbols instead of two as we do here, Castellucci and MacKenzie [2013] obtained a much lower rates while exploring touch and motion-sensing gestures: 6.6 WPM with a touchpad and 5.3 WPM with a Nintendo *Wii Remote*. Using predictive capabilities, Mackenzie and Felzer [2010] obtained an impressive 5.11 WPM with their single-switch SAK method. There is a tendency to improve the performance while offering word suggestions, which is something that DuoGrapher still does not have.

As no poster layout stood out, it should be left to the user to decide which one to use. If the device located in front of the user has a large enough display (which is the case of tablets), the codification table should be included in the graphical interface to avoid users having to look at two different locations. Related to this concern, the current symbol sequence should be shown near to the text cursor. Those changes mainly impact novices, as experts tend to memorize the codes and require less visual attention. Another important functionality for novices is a dynamic definition of the character timeout, because the high value required for avoiding errors in the beginning should smoothly fall to allow speed improvement as users gain experience.

5.8 Summary and future directions

Aiming to increase the interaction possibilities of severely motor-impaired individuals, we have presented DuoGrapher, a text entry method based on two input symbols, and Swinging-

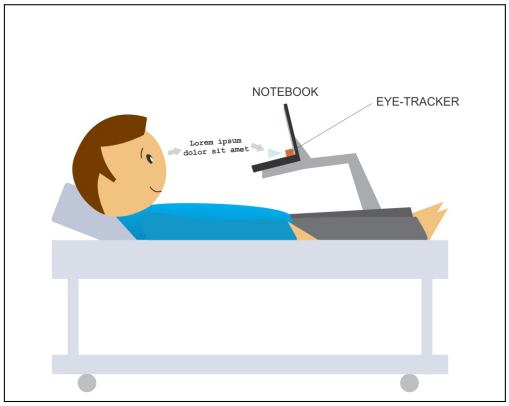
Foot, an interaction technique based on detecting internal and external heel rotations using an accelerometer.

Our design is informed by a man with a motor neuron disease and the Man-in-the-Barrel Syndrome. Unfortunately, he could not be introduced to the current version of the prototype, as the progress of the disease almost completely paralyzed his legs and feet. We have tested our prototype with 15 volunteers without disabilities, which helped us understand the challenge of a foot-based system interaction. The speed was limited by a significant amount of errors, which we claim may be greatly reduced by the changes we have indicated, such as the semi-automatic calibration and the more intuitive mapping of movements into characters. The main contribution of this study is what we believe to be the first approach that leverages the feet movements that is longer maintained by some motor-impaired individuals. The lack of experiments with motor-disabled individuals and the conduction of only one session per participant are the main limitations of this study.

As an improvement direction we suggest to incorporate predictive capabilities and to explore ambiguous methods, aiming to simplify the code of the characters and improve the entry rate. In another study direction, SwingingFoot could also be explored with different kinds of input that are required while interacting with computers and mobile devices, and for controlling home appliances, such as a TV set. In the next chapter, we present our efforts to improve text entry techniques based on eye-tracking.

Chapter 6

Filteryedping: Design challenges and user performance of dwell-free eye-typing



Another possible approach for supporting individuals with severe motor disabilities in the task of writing is to explore the movements of the eyes. In eye-typing, a virtual keyboard is shown on the screen and the user serially gazes at the intended keys. In dwell-based eye-typing, the selection of letters is made by gazing at them for a specific amount of time. However, it has two possible drawbacks: unwanted activations or slow typing rates. In this chapter[1], we propose a dwell-free eye-typing technique that is based on filtering out letters from the sequence of letters looked at by the user. It ranks possible words based on their length and frequency and suggests them to the user. We evaluate Filteryedping with a series of experiments. First, we recruit participants without disabilities to compare it with another potential dwell-free technique and with a dwell-based eye-typing interface. It proved to be a fast technique allowing an average of 15.95 words per minute after 100 minutes of typing. Then, we conduct an iterative design and evaluation with individuals with severe motor disabilities. This phase helps us to improve the technique by creating parameters that allows it to be adapted to different users.

[1]This chapter is based on the following papers:

- **D. Pedrosa**, M. G. C. Pimentel, A. Wright, and K. N. Truong. Filteryedping: Design challenges and user performance of dwell-free eye-typing. Submitted to *ACM Transactions on Accessible Computing*, 2014.
- **D. Pedrosa**, M. G. C. Pimentel, and K. N. Truong. Filteryedping: A dwell-free eye-typing technique. Submitted to *Interactivity category of the Conference on Human Factors in Computing Systems (CHI '2015)*.

6.1 Introduction

People affected by motor neuron diseases and disorders that cause muscle degeneration face communication struggles as their condition progresses over time and their ability to type and speak declines. Eye-trackers—devices that determine where on the screen the user looks—are often used to help people with these conditions communicate. While interacting with a computer using an eye tracker, eye movements may be used to control the position of the pointer. Selections can then be performed by blinking an eye (e.g., [Ashtiani and MacKenzie, 2010; Tangsuksant et al., 2012]), pushing a physical button (e.g., [MacKenzie and Zhang, 2008]) or moving specific muscles which can still be controlled and where sensors can be attached to detect such activity (e.g., [Zhao et al., 2012]). However, these solutions for performing selection do not suit all users and may cease to be viable options for many users, because of their declining capabilities. These solutions can also be considered inconvenient for tasks that demand frequent activations, which is true in the case of typing. With eye-typing, a virtual keyboard is shown on the screen and the user gazes at the intended keys that they want to type in sequence.

In dwell-based eye-typing, the user selects a letter by gazing at it for a specific amount of time, which can be less than 400 ms [Majaranta et al., 2009; Räihä and Ovaska, 2012]. Although it is currently the most common method, dwell-based eye-typing has two main drawbacks: when a short dwell-time is used, it may result in unwanted activations, which is known as the Midas' touch problem; alternatively, it can be a relatively slow text input method when a long dwell-time is used. The dwell duration has to be carefully adjusted in order to find an optimal configuration that minimizes these two problems at the same time.

A dwell-free text entry technique does not require a dwell time to detect the user's intention for inputting a letter. Some of the possible approaches include the use of eye-gestures for writing individual letters [Bee and André, 2008; Chakraborty et al., 2014; Isokoski and Raisamo, 2000; Sarcar et al., 2013; Wobbrock et al., 2008], context switching eye-typing [Morimoto and Amir, 2010], and visually navigating nested boxes of letters [Rough et al., 2014; Ward et al., 2000]. Although these works demonstrate the possibility of dwell-free eye-typing, the user must learn a new way to write a letter rather than simply looking at where the intended keys would be found on a QWERTY keyboard layout. Kristensson and Vertanen [2012] demonstrated that dwell-free eye-typing with a QWERTY-based keyboard layout can theoretically be much faster than existing eye-base text entry techniques. In their study, their software knows the text that the user wants to type and only accepts input of a character when the user looks over a key that corresponds to the next letter in the word; words can thereby be written even when the user looks at extra keys. In practice, such errors in eye-typing must be handled by an actual dwell-free technique.

In this study, we first implemented two QWERTY-based dwell-free eye-typing techniques—one of them proposed by us—and measured users' performance with each. This helped to identify a suitable candidate for dwell-free eye-typing which we were then able to evaluate alongside AltTyping [Majaranta et al., 2009; Räihä and Ovaska, 2012], currently one of the

fastest dwell-based eye-typing tools[2]. Our aim with this work was to provide an understanding of how dwell-free eye-typing compares to dwell-based approaches in actual practice.

In the preliminary study, we developed and evaluated two approaches for supporting dwell-free eye-typing. The first is a shape-based approach which implements the algorithm described by Kristensson and Zhai [2004] to recognize the intended word by comparing the shape of the path covered by the eye gaze with shapes stored in a word list. The second technique is a key filtering–based approach which recognizes the intended word by applying a weighting to the length and frequency of all possible words formed by filtering extra letters from the sequence of letters gazed by the user. For short, we refer to it as filtering. Results from this preliminary study suggested that a filtering-based approach which incorporates visual feedback of where the user is looking on the screen is a satisfactory candidate for dwell-free eye-typing that we can use to compare against a dwell-based eye-typing method—AltTyping.

Our main evaluation is divided into two phases. First, we compared the filtering using visual feedback dwell-free eye-typing method with AltTyping in an experiment involving participants without motor disabilities. Then, we recruited participants with Amyotrophic Lateral Sclerosis (ALS) and Duchenne Muscular Dystrophy (DMD) and conducted an iterative design and evaluation of the filtering method to enhance its design for those with motor disabilities. ALS is a disease that causes the degeneration of the upper and lower motor neurons, which in advanced stages causes the loss of the ability to initiate and control all voluntary movement [Medical News Today, 2009; National Institute of Neurological Disorders and Stroke, 2012]. DMD is a form of muscular dystrophy caused by a defective gene, which usually affects only boys. Individuals with this condition have progressive loss of muscle function and weakness, which begins in the lower limbs. The ability to walk may be lost by age 12, and breathing difficulties and heart disease usually start by age 20 [National Human Genome Research Institute, 2013; Patient.co.uk, 2013; U.S. National Library of Medicine, 2012].

Our evaluation shows that participants without motor disabilities were able to reach an average of 15.95 words per minute (wpm) with the proposed key filtering–based approach and 11.71 wpm with AltTyping after 100 minutes of typing with each (in the 6^{th} session). Participants affected by ALS or DMD were able to reach an average of 7.60 wpm with the filtering method and 6.36 wpm with AltTyping after using each for 60 minutes, even though many participants currently use or had previous experience with dwell-based eye-typing systems. By the end of their participation in the study, 11 out of 12 participants preferred dwell-free eye-typing over dwell-based eye-typing. Subjective workload assessment scores reveal that the workload for typing with a filtering-based technique is lower than or equivalent to the workload for typing with a dwell-based technique. In addition to these results, we learned through the course of our evaluation that preference for alternate keyboard layouts, variability in the precision and accuracy of eye-tracking, and saccades with longer duration and slower velocity are key challenges that participants with ALS and DMD had with our dwell-free eye-typing. We introduced and

[2]Downloadable at `http://www.sis.uta.fi/~csolsp/downloads.php`, accessed 24/January/2014.

evaluated two key features—a short focus dwell time and a slow movement threshold—to address these issues. These features helped overcome the problem of selecting wrong words from the candidate list and allow the system to differentiate slow eye movements from when the user's eye gaze has reached a target.

6.2 Related work

Research regarding eye-typing techniques has increased in the last 7 years, probably due to popularization of eye-trackers. One of the first works aimed to support dwell-free text input was proposed by Isokoski and Raisamo [2000]. It uses off-screen targets as a way to avoid unwanted activations caused by unintended dwells when the user gazes at a target and tries to recognize what she is looking at—the Midas' touch problem. This approach also helps to conserve the display area required by the keyboard. Isokoski discussed adaptations of different schemes—Morse code, MDITIM, Quikwriting, and Cirrin—for decoding target hit sequences into text, but has not validated the technique with controlled experiments. Quikwriting, which was originally created for pen-based computers, was also the base scheme used in Bee and André's work [2008]. In a controlled experiment with 3 participants, Quikwriting led to a typing rate of 5.0 wpm, which was slower than the 7.8 wpm rate achieved with their implementation of a dwell-based keyboard that used a dwell duration of 750 ms.

Some other gesture-based techniques for typing letters have also been proposed. In EyeK, the dwell time is replaced by moving the eye pointer from inside the key area for a letter to outside the key area for that letter and back to the key area again; with this approach, users reached rates between 5.6 wpm and 8.8 wpm [Chakraborty et al., 2014; Sarcar et al., 2013]. With EyeWrite [Wobbrock et al., 2008], which is based on EdgeWrite's letter-like unistroke alphabet [Wobbrock et al., 2003], users type by moving the gaze point over the four corners of the gesture area. In a longitudinal experiment, participants typed at an average rate of 4.87 wpm. pEYEdit, Iwrite and StarWrite are three techniques proposed by Urbina and Huckauf [2007] that rely on following the selection of a letter with a look to a specific area of the screen. These three techniques had equivalent rates, ranging from 5.9 to 7.6 wpm with novice users and from 8.4 to 11.4 wpm with advanced users. The idea of looking at a specific region to trigger the input of a letter was also explored by Morimoto and Amir [2010]. Their work introduces "context switching" through the duplication of the keyboard. In this way, the region used to trigger the input of a letter from one keyboard is the other keyboard, which is then the first step in the selection of the following letter. Trading screen space for speed, this approach led to an input rate of about 12 wpm.

One of the best performing techniques for gaze-based text entry is Dasher. It is a predictive text entry technique in which text is written by navigating nested boxes containing a letter or another symbol. The size of each box is proportional to the probability of the corresponding symbol under a language model [Rough et al., 2014; Ward et al., 2000]. In an experiment comparing it with a baseline eye-typing method, Dasher resulted in significantly faster entry rates

(14.2 wpm versus 7.0 wpm) [Rough et al., 2014]. Rough et al. noted that "different experimental setups sample different participants, use different apparatus and stimuli, and use slightly different procedures", however, the rates obtained with the baseline eye-typing method were comparable with several works that they cited. A disadvantage is that Dasher uses an unfamiliar interface that requires a lot of concentration to learn and use.

The fastest eye-typing tool reported to date is AltTyping [Majaranta et al., 2009], with reported entry rates reaching 20–24 wpm [Räihä and Ovaska, 2012]. It enables the user to adjust the dwell time directly on the keyboard interface. AltTyping's relatively high input rate might be due to the method that was used to analyze the data, which focused on expert and error-free performance. In particular, they "decided to analyze the character-level text entry rate only for characters entered correctly after another correctly entered character. [...] Furthermore, if the participant glanced at either the model line or the result line in between two key presses, the latter key press was again omitted from analysis" [Räihä and Ovaska, 2012].

Our Filteryedping prototype is an implementation of a dwell-free eye-typing technique for a QWERTY keyboard layout. Kristensson and Vertanen [2012] previously showed that such a method could be "potentially much faster" than current eye-typing implementations. In their experiment, users reached a mean entry rate of 46 wpm using a system which simulates a perfect recognizer for a dwell-free eye-typing. That is, their study software knew in advance what the user wanted to type. Each time the user looked at the next letter in the sequence, that letter was inserted. There was no way to commit an insertion error even though the user may have looked at additional letters while typing.

6.3 Dwell-free eye-typing

Before performing a comparison between dwell-free and dwell-based eye-typing techniques, we first explored two possible approaches for dwell-free eye-typing. We compare a shape-based approach against a key filtering–based approach. In a position paper, Hoppe et al. [2013] previously suggested that perhaps a dwell-free technique could significantly increase the entry rate of eye-typing systems. They proposed an approach, called Eype, which turns eye gaze data over a QWERTY keyboard into a trace that joins characters looked at by the user. Eype removes repetitions within the trace and then compares the trace against optimal traces of all words in a corpus to identify the closest match. No performance evaluation of Eype has been reported. However, this approach is similar in concept to SHARK^2 [Kristensson and Zhai, 2004], an established touch-based word-level gesture keyboard technique which has inspired many subsequent systems, such as Word Flow[3], and has been well evaluated. Thus, for our shape-based approach, we implement an adaptation of SHARK^2, to work with eye-trackers. As an alternative approach, we developed Filteryedping, a key filtering technique which supports the idea that

[3] `http://research.microsoft.com/apps/video/default.aspx?id=211650`, accessed 10/July/2014.

some of the letters that the user looks at may not be part of the desired word.

In this section, we first describe our implementation of the two dwell-free techniques in detail. Then, we describe an experiment comparing them. The obtained results indicate that Filteryedping is a satisfactory candidate for dwell-free eye input.

6.3.1 The Filteryedping technique

The Filteryedping technique recognizes the intended word by looking in a word frequency list for all the words that can be formed when discarding none or some of the letters which the user has looked at. The possible words are sorted based on the length and frequency and presented to the user as a list of options. The name of the technique is an example of a stream of letters that can generate, among others, the words "filtered", "eye", and "typing". Thus, it succinctly describes and demonstrates the idea of the technique: filtered eye-typing.

Interface

Figure 6.1 shows the interface of the prototype that implements Filteryedping. The user writes a word using this technique by looking at each letter of the word, the same way she would while using a dwell input technique, except that she does not have to dwell over a letter to select it. Visual feedback is provided to show the user where the system recognizes her current gaze position to be on the screen. Filteryedping displays the key looked at by the user in a different color (see letter "a" in Figure 6.1). The time it takes for the system to recognize the location and highlight a key (< 33 ms) is almost imperceptible.

After typing the last letter of a word, the user must look at the bottom part of the interface. An empty target helps the user to look at the position where the top-ranked suggested word will appear. Then, the user can traverse the candidate list to look for the intended word. Arrow buttons in the extremities support paginating for more candidates. Again, no dwell time is involved. If the user does not find the word she wanted to type, she can try to type that word again by looking back to the keyboard while one of the two arrow buttons is selected. By doing this, no word will be written. To accept the highlighted candidate word, the user must simply look to the target / typed text area or back to the keyboard to type the next word. In this way, the user can confirm that the correct word has been entered or continue to type if she does not need to check. A storyboard illustrating this process is shown in Figure 6.2.

We intentionally designed the interface to show a small visible area for each key to help direct the user's gaze to the center of the detection area. However, the detection area for a key is not the same as its visible area. Figure 6.3 shows in green the actual detection area for each key. Note that the keys overlap horizontally. Not considering the overlap area, the aspect ratio of a key matches the one used in physical QWERTY keyboards, which is 0.867. However, the small width of a key may cause some horizontal recognition errors. To overcome this issue, we made the detection area wider, introducing an overlap area. If the system recognizes the gaze

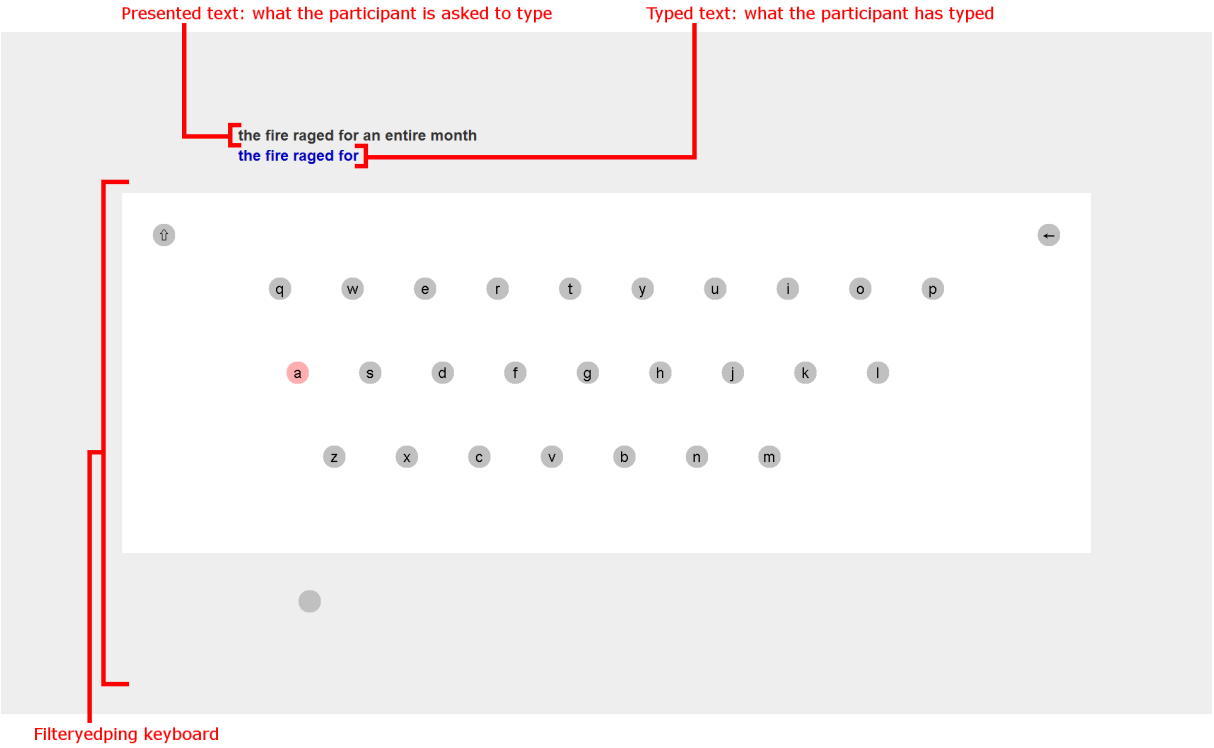


Figure 6.1: Study software with the Filteryedping interface while user looks at the a key.

inside an overlap area, such as between ‘e’ and ‘r’, it includes in the character stream not only one letter, but the concatenation of the first key with the second key and then the first one again. In the above example, it would concatenate “ere” in the stream. That way, the stream is useful not only for words that include one of the letters, but also for words that include “er” or “re”. The user will see both letters highlighted in the interface at the same time and can safely proceed to type the next letter in the word.

If the user looks at the top right key ("←"), a menu with four options is shown. The first three options are Delete word, Backspace, and Enter. The option menu works exactly like the candidate list: no dwell is required and the functionality corresponding to the selected button is activated only after the user looks back to the keyboard or to the text area. The fourth button works as a dismiss button, for cases in which the menu was opened by mistake (Figure 6.4 (right)). Similarly, the top left key (“⇑”) offer options for Shift, Caps Lock, and switching to a numbers and punctuation layout (Figure 6.4 (left)).

In order to also provide auditory feedback, we used the FreeTTS[4] speech synthesizer library. The prototype speaks each written word or command name (“deleted”, “backspaced”, “shift”, “caps lock”, and “lower case”) immediately after confirmation / activation.

[4] `http://freetts.sourceforge.net`, accessed 07/July/2014.

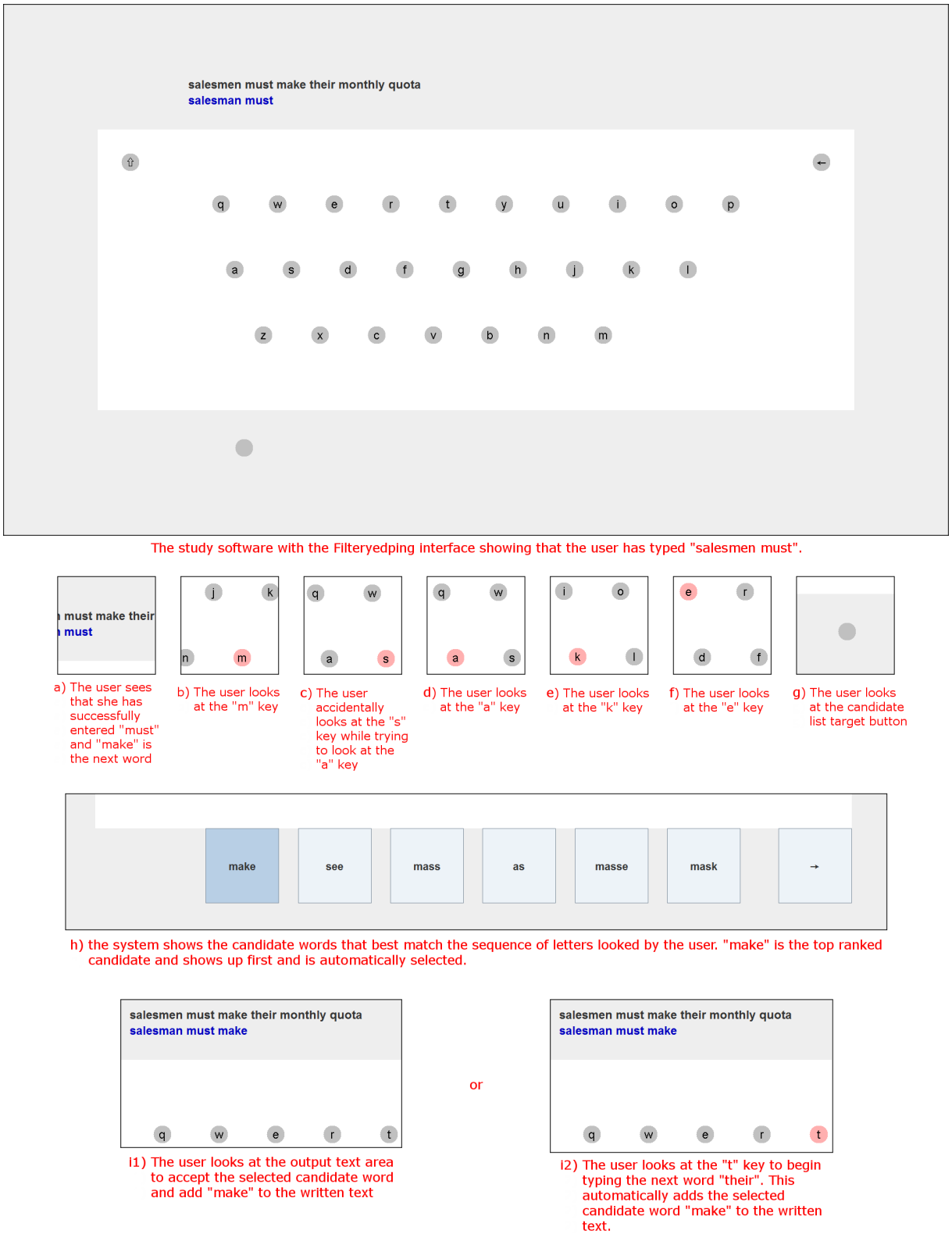


Figure 6.2: A storyboard illustrating a user typing make.

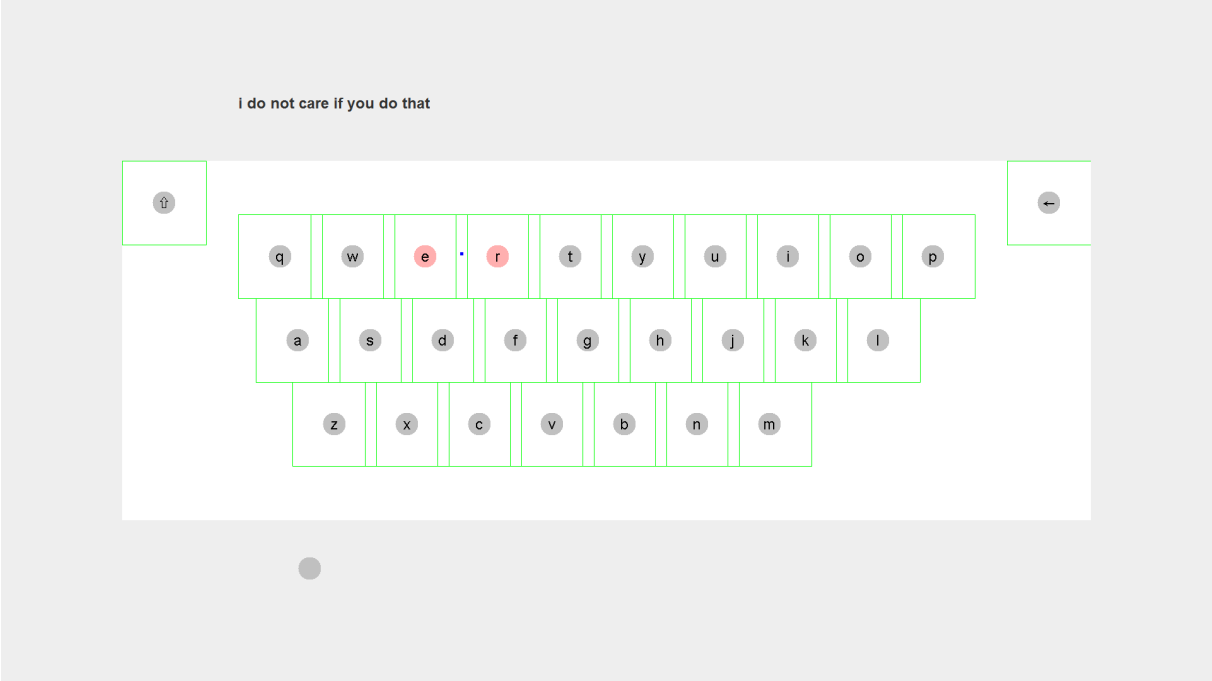


Figure 6.3: Filteryedping interface: Actual detection area for each key (not seen by the user).

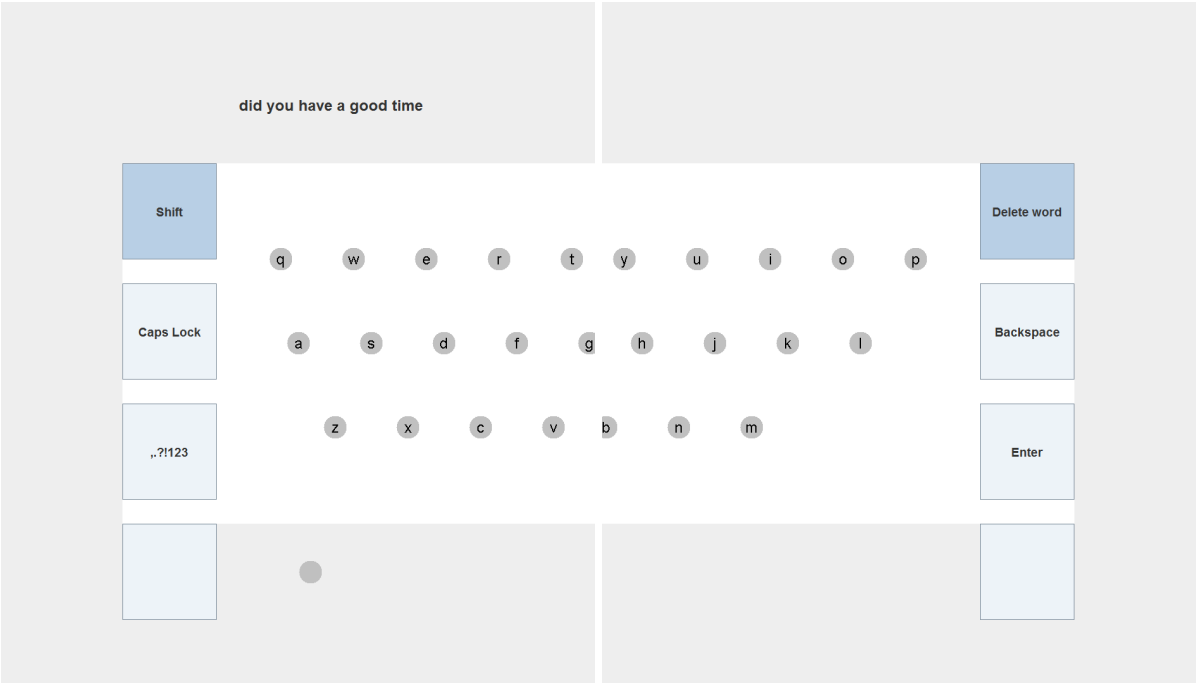


Figure 6.4: Filteryedping interface: left view while user looks at “Shift” button (left) and right view while user looks at “Delete word” button (right).

Word frequency list

Our word frequency list was created by starting with words from the Corpus of Contemporary American English (COCA) [Davies, 2008], and reducing it to omit words that contained non-alphabetical characters and words that are not included in dictionary.com. Then, we added the British spelling for 2 words that were in the MacKenzie and Soukoreff [2003] phrase set used in the experiments (see Section 6.3.3). The result was a list of 133,223 words with their associated frequencies of occurrence. Additionally, we included several common misspellings (see Section 6.3.1 – Spell correction).

Support for entering out-of-list words

The technique explained so far allows for input of words contained in the word frequency list. To allow users to type words out of the list, such as passwords or less common first and last names, the user can dwell about 1 second (though this value is adjustable) over each desired keys and then look at the left-most button in the candidate list after typing all keys. It will show the word composed by the dwelled keys. That place is where the "previous page" button is shown after paginating at least once. If the user has not dwelled over any letter, the first page of the candidate list does not show anything in there. If the user has dwelled by mistake over one or more letters, all the user has to do is to ignore the word suggested in the left most button. In this way, we support dwell-based eye-typing without requiring an explicit mode change.

The dwell time is a configurable parameter. After half of this duration, the visual feedback starts to change to indicate to the user that the letter is about to be included as a dwelled letter. The key color smoothly makes a transition from pink to red. Having completed the dwell time, an abrupt transition (or blink) from red to pink is used to indicate that the letter was appended to the dwelled sequence of characters to be shown in the first candidate word button.

Ranking algorithm

As commented before, words in the candidate list are sorted based on the length and frequency. We weight those two factors using the following formula:

$$score(word) = log_{10}(freq(word)) + w \times length(word) \tag{6.1}$$

where $word$ is in the frequency list, $freq(word)$ is the number of times that word appears in the corpus, $length(word)$ is the number of characters in the word, and w is a weight. The higher the score of a word, the closest to the beginning of the candidate list it should be. To define the value of w, we compare the average position that all the words in the word frequency list would have in the candidate list in case of a perfect gaze input from the user, for different values of w. We weigh the average by considering the frequency of the word. Figure 6.5 shows the average position for different values of w. The jumps in the graph happen when two words that have

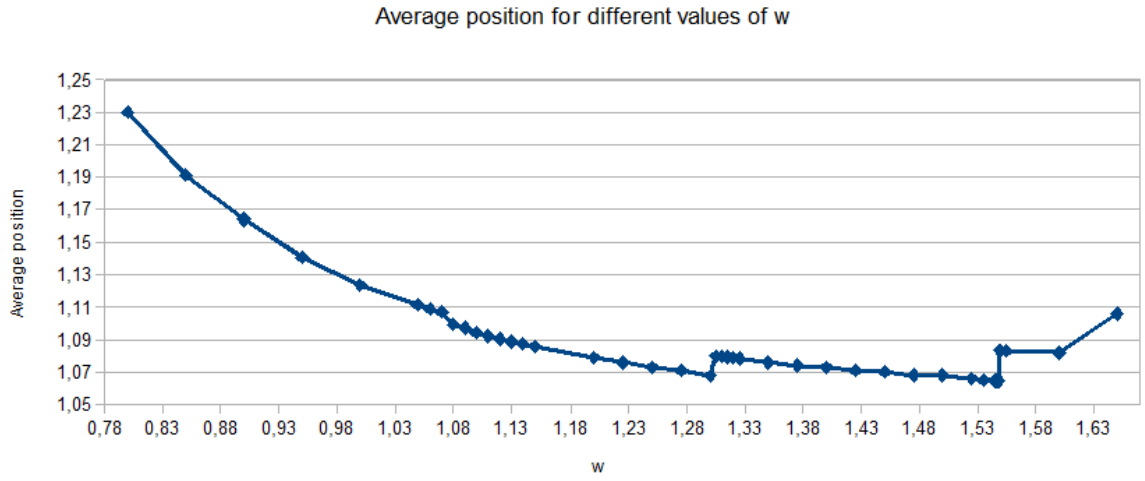


Figure 6.5: Average position of words in the candidate list for different values of w (weight of the length of word factor).

the same input ("to" and "too", for example) change positions. By choosing the weight that leads to the minimum average position, we would be trusting too much in the user's ability to perform perfect input. Instead, we decided to use the value of 1.08, which is small enough to not overweigh the length of the word in detriment to the frequency, but still leads to a small average position of 1.0996.

The basic algorithm for creating the candidate list is shown in Algorithm 6.1 and the pseudo-code for the method that tests if an input stream contains a word is shown in Listing 6.1.

Algorithm 6.1: Creating the list of suggestions.

1: **for all** words in the frequency list **do**
2: **if** input stream contains it (match) **then**
3: Add to the suggestions list
4: **end if**
5: **end for**
6: Sort the list by the rank

Listing 6.1: Testing if an input stream contains a word.

```
1  boolean match(String input, String word) {
2      valid = false;
3      i = 0;
4
5      for (w = 0; w < word.length(); w++) {
6          for (; i < input.length(); i++) {
```

```
            if (word.charAt(w) == input.charAt(i)) {
                valid = true;
                break;
            }
        }
        if (!valid) {
            break;
        }
    }
    return valid;
}
```

Processing eye-tracker data

Because of the way human eyes work and the limited accuracy and precision of eye trackers, the data provided by eye-trackers via the APIs are noisy. To address this issue, we implemented the saccade detection and fixation smoothing algorithm proposed by Kumar [2007]. Our module for processing the data generated by the eye-tracker is completely independent from the prototype. It first acquires the point on the screen corresponding to the eye gaze, then it applies the smoothing algorithm, and finally it issues an operating system command to move the mouse pointer to that position.

In each eye-tracker cycle, the application is notified if neither, one, or both eyes were detected. When both eyes are detected, the application has access to the separate position for each eye. We calculate and use the average position. If only one eye is detected, we simply use the provided position. We do not do anything in the cycles when no eye is detected by the eye-tracker.

The first step of the saccade detection and fixation smoothing algorithm is to determine whether the most recent data point is the beginning of a saccade or a continuation of the current fixation. We use a saccade threshold of 90 pixels, which is a bit less than the 115 pixels horizontal separation between the keys of our keyboard. The basic idea of the algorithm is to handle the current fixation as a weighted mean of the last points belonging to the fixation (less than the saccade threshold apart) that occurred within the dwell time. We use a list of 15 points, which for an eye-tracker that works at 30 Hz is about 500 ms.

Spell correction

Misspelled words can be easily highlighted or automatically corrected in many text editors, which handle this immediately after the user has finished typing the word. In Filteryedping, misspellings caused by letter insertions are not a problem, because of the filtering approach used. However, if the user skips a letter or switches the order of letters, neither the misspelled word nor the desired word would be shown in the candidate list. To help users in these cases, we have created a list of misspelled words by merging two public online available lists[5] and combined

[5]"Wikipedia:Lists of common misspellings/For machines" (`http://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings/For_machines`, accessed 06/Mar/2014) and "Common misspellings" (`http://www.oxforddictionaries.com/words/common-misspellings`, ac-

this with our word frequency list. We create the rank for these words using the length of the misspelled version, but the frequency of the corrected one. Thus, when the user misspells a word, the algorithm is able to detect the user's intention to write a word. Instead of adding the misspelling to the candidate list, the system adds the corresponding correctly spelled word.

6.3.2 Shape-based eye-typing

The shape-based eye-typing technique uses an algorithm which recognizes the intended word by comparing the shape of the path covered by the gaze with those from a dictionary that stores the shape for each word in a corpus [Kristensson and Zhai, 2004]. We followed the algorithm described by the authors as close as possible. We used 20 points as the total number of sampling points in the Proportional Shape Matching algorithm. We used 1000 pixels as the bounding box length L while normalizing the shapes in scale and location. For the integration of the location and shape channel, we used 44 and 100 for σ in the shape Gaussian probability density function for location and shape, respectively.

As the authors suggested, we also prune all word candidates that have shape or location distance bigger than 2σ. Then, as a second pruning step, we further process only the first 48 words (6 pages of suggestions) from the list. Finally, we sort the candidates by distance, showing first the ones that better match the input.

When eye-typing, there is no explicit delimiter for a stroke—which is the case in touch-based typing, where the finger contact with the screen is the delimiter. As a result, the beginning and end of a stroke must be trimmed to remove the additional points that the user's eye gaze crosses through as it enters and leaves the keyboard. Our implementation discards points in the beginning of the stroke until the first point is within 60 pixels of the following two points. Similarly, it discards points at the end of the stroke until the last point is within 60 pixels of the previous two points. Other strategies could be tested and employed, such as using a dwell time to unequivocally identify the first letter of the word [Hoppe et al., 2013].

Except for the visual feedback, the interface for our implementation of shape-based eye-typing is the same as the interface for Filteryedping. Besides highlighting the key currently looked at by the user in pink, the system also draws a fading line connecting the last 60 gazed points (Figure 6.6). The eye-tracker data processing method and spell correction described for Filteryedping were also used with the shape-based eye-typing technique.

6.3.3 Comparison of approaches

To identify a suitable dwell-free eye typing candidate, we conducted an experiment in which half of the participants were asked to perform typing tasks with and without visual feedback using the shape-based technique and the other half did the same using the Filteryedping technique.

cessed 06/Mar/2014)

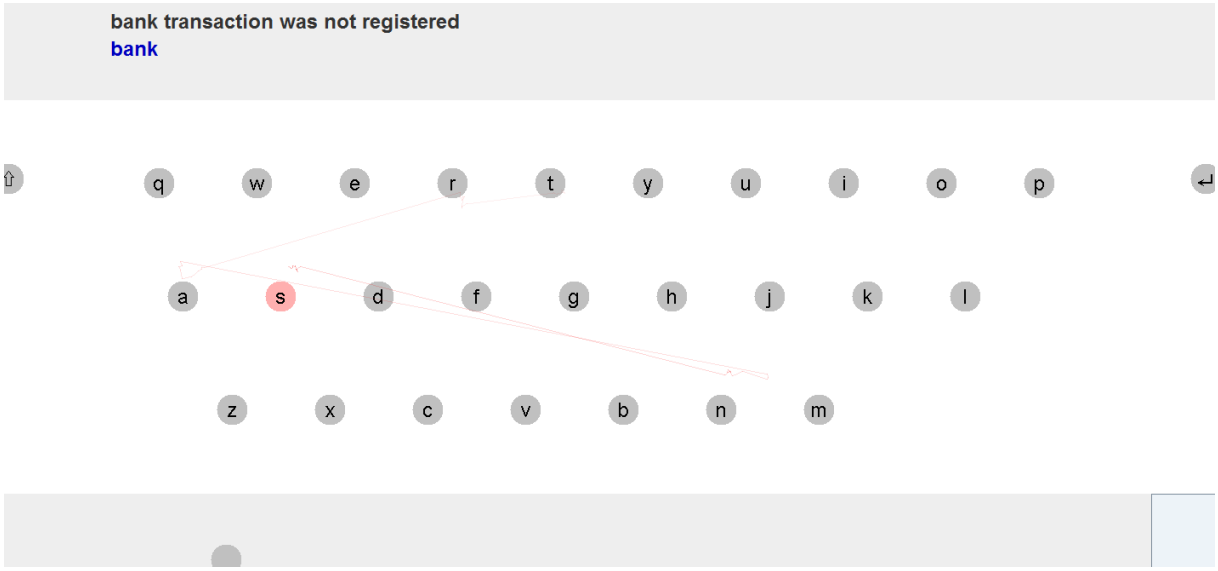


Figure 6.6: Visual feedback for the shape-based eye-typing technique.

In the typing without feedback conditions, the gaze position recognized by the system is not indicated to the user. The technique and the order of feedback used were randomly selected. The last participant(s) were assigned a technique and order of feedback type so that we had the same amount of users for each condition. Participants were asked to type as fast and accurate as possible; additionally they were instructed not to use the dwell functionality.

The version of Filteryedping used in this preliminary study was the same as described in Section 6.3.1, except for four differences: 1) confirmed words were not spoken by the prototype, 2) the candidate list bar had 8 suggestions instead of 6, and there was no space between the suggestions, 3) there was no space between options in the vertical menus and 4) Enter was the first option in the right vertical menu.

We recruited 12 participants (5 females) via a university mailing list containing the e-mails of students and staff, and via word of mouth with personal contacts in our social circle who are or have access to potential participants. Participants were 25.2 years old in average ($std = 7.6, min = 19, max = 44$). All have extensive experience with QWERTY keyboards and were fluent in English. None of the participants have used an eye-tracker before, except one, who had used it for less than one hour. Two participants use glasses, two use contact lenses, and 8 performed the tasks without any corrective lenses.

The experiment consisted of 3 sessions. No more than 72 hours elapsed between sessions, and no more than two sessions occurred on the same day. If two sessions were performed on the same day, at least two hours elapsed between them. Each session was divided into 4 blocks. In each block, participants were asked to type, as fast and accurate as possible, randomly selected phrases from the set of 500 phrases created by MacKenzie and Soukoreff [2003], for 5 minutes (time between phrases were not considered). The number of phrases typed by a participant in each block depends entirely on how fast she was. Participants were not interrupted in the middle of a phrase when time was completed. After pressing Enter to indicate the end of a phrase, the

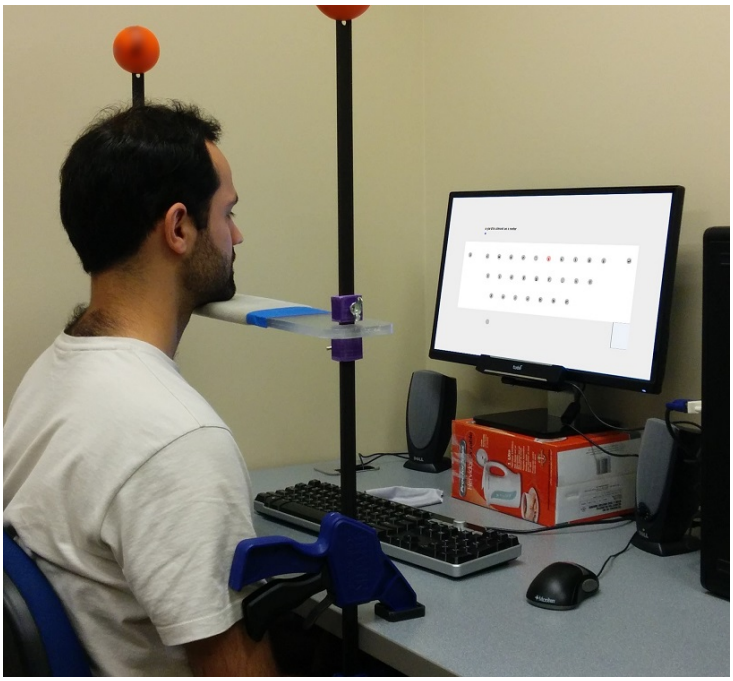


Figure 6.7: Study setup for the trials: A chin rest, speakers, a monitor showing a virtual keyboard, and an eye-tracker attached to its bottom part.

interface showed to the participant the adjusted typing rate (typing rate adjusted to take into consideration the uncorrected error rate) near the typed phrase. This was done to motivate the participants to try to beat their own marks.

Half-way through each session, a 5-minute break was taken and the feedback type was changed. To allow the participants to become familiar with the technique and to warm up, they started each half-session by practicing for five minutes. In all sessions, participants were asked to sit still in front of the monitor, resting their chin over a chin rest (Figure 6.7), to calibrate the eye-tracker and perform the typing tasks. We used a Tobii REX eye-tracker (which works at a sampling rate of 30 Hz) attached to a 21.5" Dell monitor (1920 x 1024 pixels). Nine-point calibration was used in the study. Although the chin rest was not necessary, it was used to help participants understand that they only needed to move their eyes. Before each block the participants were allowed a chance to recalibrate until they were comfortable with the accuracy of the eye-tracker.

The first session began with an informed consent agreement and a demographic survey. Also in the first session, the participants were briefed on the procedures of the experiment and introduced to the dwell-free eye-typing technique that they were about to use. Additionally, after finishing the second block of the first session only, the participants were given a page to read containing instructions on how to rate scales of a subjective workload assessments with the NASA Task Load Index (NASA TLX)[6] and a page containing the instructions on how to specify the sources-of-workload. In all sessions, after blocks 2 and 4, the participants were asked

[6]NASA TLX – Paper/Pencil Version. `http://humansystems.arc.nasa.gov/groups/tlx/paperpencil.html`. Accessed 17/Jul/2014.

to fill a NASA TLX rating sheet and the sources-of-workload evaluation. At the end of each session, the participants were asked to fill a form regarding their experience, thanked for their time, and compensated US$ 10. At the end of the final session, each participant also received a US$ 25 bonus for completing all sessions. An additional bonus of US$ 10 was given to the two participants (one per technique) who obtained the fastest adjusted typing rate in each session. This compensation schedule was chosen to encourage continued participation in the experiment and effort to type fast and accurate.

6.3.4 Results

To analyze the study data, we used StreamAnalyzer for computing the typing rate and the rate of errors that were left in the transcribed text. It is a publicly available tool that analyzes the text input stream logs and produces text file output containing several statistics [Wobbrock and Brad A. Myers, 2006]. Results from each block were averaged to form a single measure per block. Then, results from blocks 1 and 2 and results from blocks 3 and 4 were averaged to form a single measure per participant per session per condition. All statistical significance tests used a significance level of α = 0.05.

Metrics

Text entry rate (WPM) is defined in words per minute, where a word is 5 characters, including spaces. It is obtained by dividing the number of typed words by the number of minutes measured from the moment the user's gaze enters the keyboard for the first time after reading the target phrase to the moment the last word is written. The time taken to hit the Enter key was not included.

As a metric for the amount of errors left in the transcribed text, we use:

$$\text{MSD error rate} = \frac{MSD(P,T)}{\overline{S_A}} \times 100\% \tag{6.2}$$

where $MSD(P,T)$ is the minimum string distance between the present and transcribed strings, and $\overline{S_A}$ is the mean length of the alignment strings [MacKenzie and Soukoreff, 2002].

To better understand the quality of the ranking algorithm, we are also interested in the average position of the selected words in the candidate list (avgPos). An average of 1 would mean that all the written words were found in the first position. Recall from Section 6.3.1 (Ranking algorithm) that with perfect user input, our Filteryedping algorithm results in an avgPos of 1.0996 for all words in the dictionary.

To evaluate the task workload of each condition, we used the NASA TLX assessment tool. The overall workload score is the average of the ratings of the six subscales, weighted by the contribution of each factor.

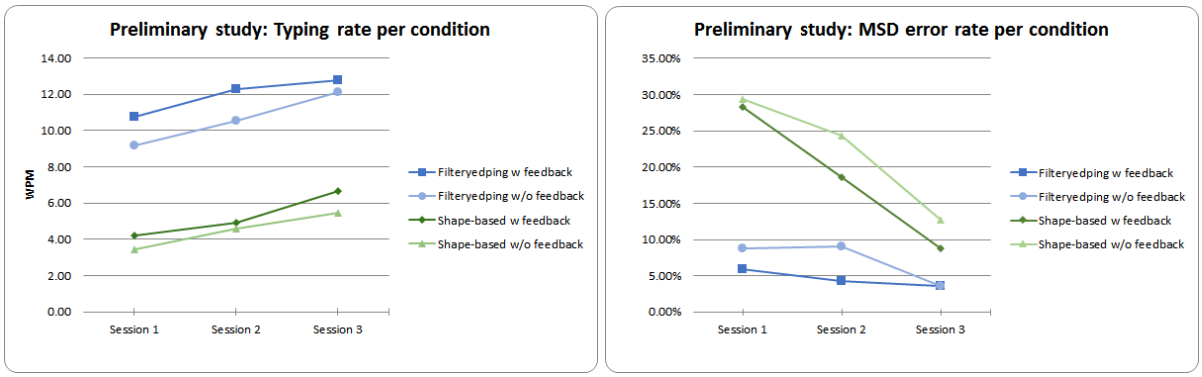


Figure 6.8: Results of the preliminary study comparing two dwell-free eye-typing techniques: Text entry rate (left) and MSD error rate (right).

Text entry rate (WPM)

Figure 6.8 (left) shows the average entry rate obtained per condition per session. The entry rate with Filteryedping was on average 2.4 times faster than with a shape-based approach.

We conducted two repeated measures analysis of variance, one for each technique, to analyze the impact of Feedback and Session on the typing rate. For Filteryedping, there was no significant main effect for Feedback ($F_{1,30} = 1.17, \eta^2 = 0.034, p = 0.29$); no significant main effect for Session ($F_{2,30} = 1.36, \eta^2 = 0.080, p = 0.27$); and no significant interaction between Feedback and Session ($F_{2,30} = 0.07, \eta^2 = 0.004, p = 0.93$). For the shape-based method, there was no significant main effect for Feedback ($F_{1,30} = 0.50, \eta^2 = 0.015, p = 0.49$); no significant main effect for Session ($F_{2,30} = 1.48, \eta^2 = 0.088, p = 0.24$); and no significant interaction between Feedback and Session ($F_{2,30} = 0.05, \eta^2 = 0.003, p = 0.95$). Although we expected that participants would improve with practice and that the use of feedback would help them to achieve better performance, we found no evidence to support this.

Next, we aggregated participants' typing rate in each session by averaging the values obtained with and without feedback. A new repeated measures analysis of variance was conducted to analyze the impact of Technique and Session on the typing rate. There was a significant main effect for Technique ($F_{1,30} = 34.79, \eta^2 = 0.512, p < 0.001$); but no significant main effect for Session ($F_{2,30} = 1.59, \eta^2 = 0.047, p = 0.22$); and no significant interaction between Technique and Session ($F_{2,30} = 0.02, \eta^2 = 0.001, p = 0.98$).

MSD error rate

Figure 6.8 (right) shows the average MSD error rate obtained per condition per session. The shape-based approach resulted in an average of 3.5 times more errors than Filteryedping.

To analyze the impact of different factors on the MSD error rate, we conducted the same sequence of analysis as we had done for typing rate. For Filteryedping, there was no significant main effect for Feedback ($F_{1,30} = 2.75, \eta^2 = 0.071, p = 0.11$); no significant main effect for Session ($F_{2,30} = 2.11, \eta^2 = 0.110, p = 0.14$); and no significant interaction between

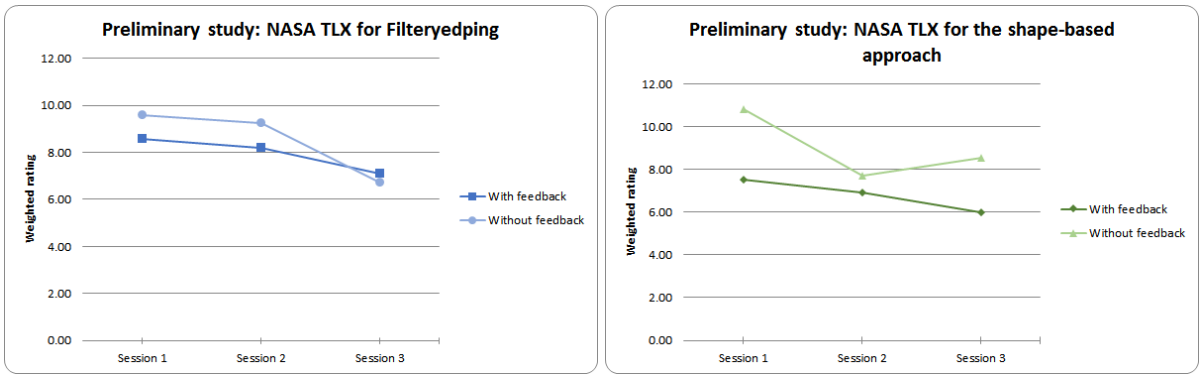


Figure 6.9: NASA TLX weighted ratings of the preliminary study comparing two dwell-free eye-typing techniques: Filteryedping (left) and shape-based approach (right).

Feedback and Session ($F_{2,30} = 0.82, \eta^2 = 0.042, p = 0.45$). For the shape-based approach, there was no significant main effect for Feedback ($F_{1,30} = 0.45, \eta^2 = 0.012, p = 0.51$); a significant main effect for Session ($F_{2,30} = 3.83, \eta^2 = 0.200, p = 0.03$), which means participants committed less errors with practice; and no significant interaction between Feedback and Session ($F_{2,30} = 0.06, \eta^2 = 0.003, p = 0.94$).

After averaging values obtained with and without feedback, there was a significant main effect for Technique ($F_{1,30} = 21.51, \eta^2 = 0.340, p < 0.001$); and a significant main effect for Session ($F_{2,30} = 4.14, \eta^2 = 0.131, p = 0.03$); but no significant interaction between Technique and Session ($F_{2,30} = 1.77, \eta^2 = 0.056, p = 0.19$).

NASA TLX

For the task workload, we consider only the responses from 8 participants (4 per technique), because we do not have separated responses for each feedback condition from the first four participants. Figure 6.9 shows the average NASA TLX rating obtained per condition per session. Because it is a subjective evaluation, it is hard to compare ratings given to Filteryedping with ratings given to the shape-based approach, because users tend to give ratings that differentiate between the two conditions that they have used, which were with and without feedback. For both approaches, the results suggest that the workload when using visual feedback is lower than or equivalent to the workload when typing without feedback.

avgPos

Figure 6.10 (left) shows the average position of the selected words in the candidate list (avgPos) per condition per session.

We aggregated avgPos in each session by averaging the values obtained with and without feedback. A repeated measures analysis of variance was conducted to analyze the impact of Technique and Session on the avgPos. There was a significant main effect for Technique ($F_{1,30} = 56.27, \eta^2 = 0.639, p < 0.001$); but no significant main effect for Session ($F_{2,30} =$

$0.68, \eta^2 = 0.016, p = 0.51$); and no significant interaction between Technique and Session ($F_{2,30} = 0.22, \eta^2 = 0.005, p = 0.80$). This suggests that the key filtering–based approach identified the words typed by users in dwell-free mode better than a shape-based approach.

Feedback conditions

In the post test questionnaire, we asked the participants: "Which of the two feedback conditions (with feedback or without feedback) allows easier input?" Figure 6.10 (right) shows the participants' answers separated by technique. Except for the first session with Filteryedping, participants believed that the use of feedback facilitates input.

6.3.5 Discussion

The objective metrics (WPM, MSD error rate, and avgPos) and subjective metric (NASA TLX) indicate that the key filtering–based approach is a suitable dwell-free eye-typing method. Participants also indicated that the use of visual feedback was a positive functionality to include. Thus, we decided to make some improvements and continue our investigation using the Filteryedping with visual feedback prototype only.

One of the main problems that we learned about the prototypes was related to users selecting words in the candidate list and options in the vertical menus accidentally. This was a common problem because the position reported by the eye-tracker is subject to tiny, rapid, and unstable movements associated with visual fixations, and lack of accuracy and precision in the eye tracking mechanism itself. Because the selection of a candidate word or a menu option requires only the detection of one single point inside that button followed by the detection of a single point in the keyboard or in the text area, unstable fixation or noise in the eye tracking data can easily cause this error.

Related to this, four participants mentioned in the post-test questionnaire that they would prefer to have the "Delete word" and "Enter" buttons further apart from each other. Because

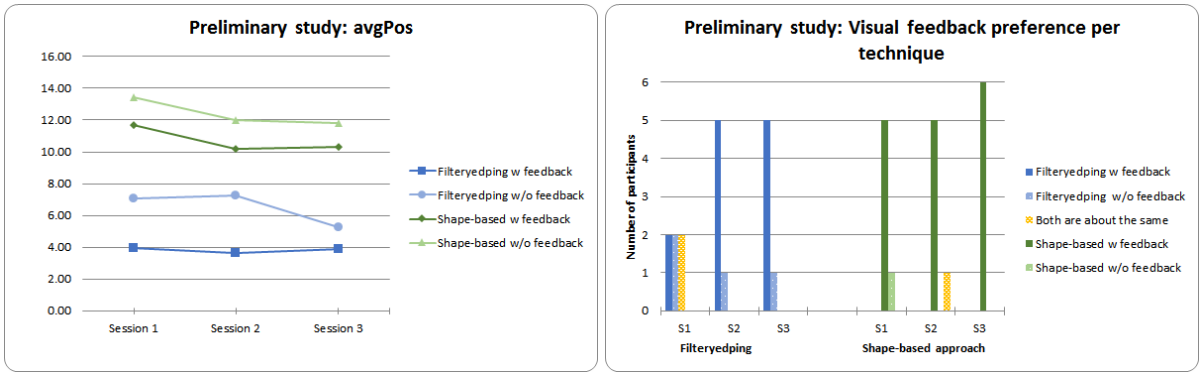


Figure 6.10: Results of the preliminary study comparing two dwell-free eye-typing techniques: average position of the selected words in the candidate list (left) and answers of the participants to the question: "Which of the two methods allows easier input?" after each session (right).

both options were in the same vertical menu, users sometimes select one instead of the other by error. Additionally, we also observed that participants wanted to use "Delete word" more frequently than "Enter," thus, it should be easier to access.

To overcome these problems, we introduced two changes to the prototype: 1) we reduced the number of options in the candidate list from 8 to 6 to create some space between them, 2) we changed the order of the options in the right vertical menu from "Enter", "Delete word", and "Backspace" to "Delete word", "Backspace", and "Enter".

6.4 Dwell-free versus dwell-based evaluation

Having identified a satisfactory candidate for dwell-free eye-typing, the next step was to perform a study comparing it with dwell-based eye-typing. For this purpose, we used AltTyping [Räihä and Ovaska, 2012], the fastest dwell-based eye-typing tool reported in the literature. We divided this study in two phases:

1. Phase 1: a performance evaluation with participants without physical disabilities;
2. Phase 2: an iterative design and evaluation with participants with ALS and DMD.

In this section, we first describe how AltTyping was used. Then, we describe the experiment conducted in Phase 1 and its results. We finish by presenting the methodology and results for Phase 2.

6.4.1 AltTyping

AltTyping allows the user to type a letter by fixating her gaze over that key for a certain amount of time. Räihä and Ovaska's evaluation of AltTyping [2012] was divided into two phases: 1) a Learning Phase, during which participants used AltTyping for ten sessions of about 15 minutes each and could adjust dwell time as they wished, and 2) an Advanced Phase, during which the same participants used AltTyping for five 15-minutes sessions. In the Advanced Phase, the dwell time started at 410 ms in first session and was decreased by 40 ms each session.

In our study, we used the first session to allow participants to become familiar with the interface, similar to the Learning Phase from Räihä and Ovaska's study. The dwell time started at 450 ms and could be adjusted by the participant at any time. From the second to the sixth session, we reproduced the dwell times used in the five sessions from the Advanced Phase in Räihä and Ovaska's study.

In Räihä and Ovaska's study [2012], a 17-inch monitor with 1280×1024 resolution was used. AltTyping was displayed as a full screen interface. To mimic their conditions as much as possible, we set AltTyping window to be 17 inches diagonally (using specifically a 1350×1080 window size to keep the same aspect ratio as their study), centered it, and used a software frame around it to hide all other elements of the interface (Figure 6.11).

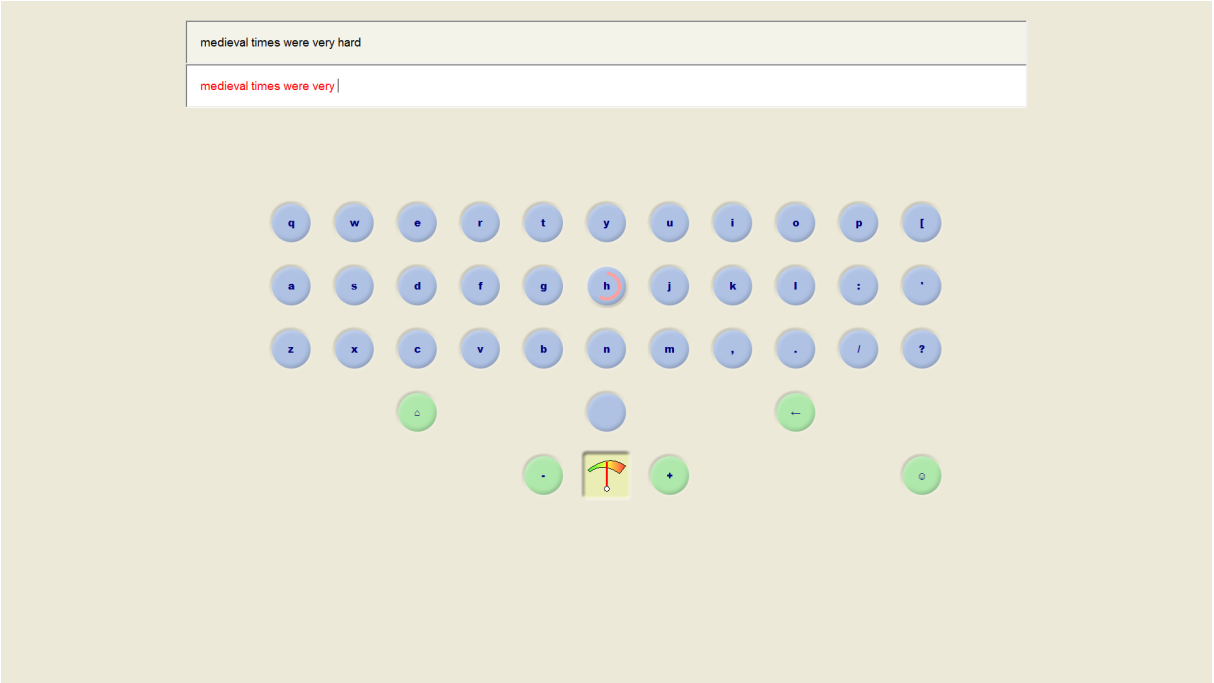


Figure 6.11: The AltTyping interface during first session of the experiment. The following 5 sessions did not show the "-" and "+" buttons nor the dwell time indicator.

Because Tobii REX is not one of the eye-trackers supported by AltTyping, we configured the software to work with the mouse mode and used the same module for processing eye-tracker data used with Filteryedping. We also configured the operating system to hide the mouse cursor, to minimize visual distraction for the participant.

6.4.2 Phase 1

In Phase 1, we recruited participants without physical disabilities. The experiment followed a within-subject design, in which all the participants performed typing tasks using Filteryedping with visual feedback and using AltTyping. The order of use of the techniques was randomly selected at the beginning of the first session for each user, and was kept constant for each participant's remaining sessions. The last participant(s) were assigned the order of use of the techniques so that we had the same amount of users starting with each condition.

The version of Filteryedping used in Phase 1 was the same as described in Section 6.3.1, except for three differences: 1) confirmed words were not spoken by the prototype, 2) there was no space between the options in the vertical menus, and 3) the stream of letters was cleared every time the user looked outside the keyboard (for example, to check the typed or presented text). Differing to what we had in the preliminary study, this version did not show the adjusted typing rate after each phrase. We removed that feedback in order to ensure that consistent information was offered to the participants while using AltTyping and Filteryedping. Participants were again instructed not to use the dwell functionality.

We recruited 6 participants (2 females) using the same recruiting methods as described for

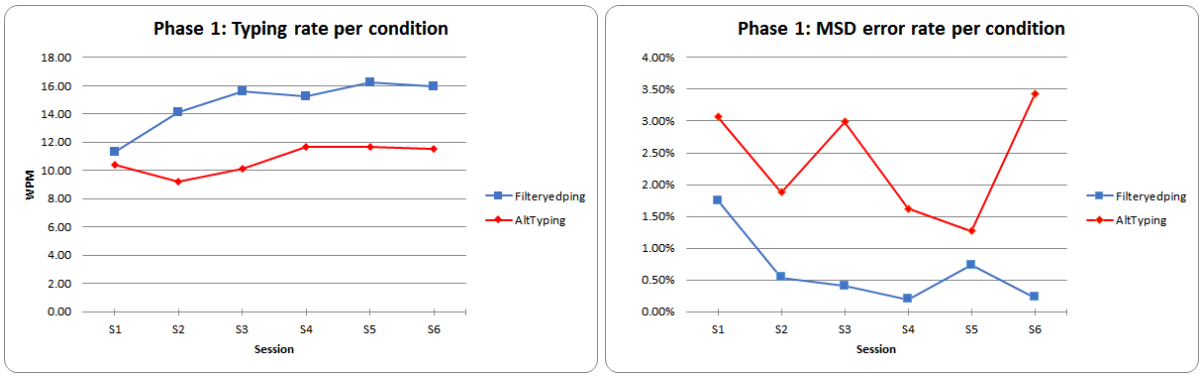


Figure 6.12: Results of the comparison between a dwell-free and a dwell-based eye-typing technique: Text entry rate (left) and MSD error rate (right).

the preliminary study comparing the dwell-free approaches. Participants were 22.8 years old in average ($std = 1.3, min = 21, max = 24$). All have extensive experience with QWERTY keyboards and were fluent in English. None of them have used an eye-tracker before. Three participants used glasses and three performed the tasks without any corrective lenses.

This phase consisted of 6 typing sessions. Each session was divided into 2 blocks of 20 minutes each. Before each block, participants were given 2 minutes to practice and become familiar with the typing technique they would use next. Again, we used NASA TLX to evaluate subjective workload. However, we adjusted the procedure so that each participant specified the sources of workload only once per technique, after using it in the last session. The phrase set, the rules to session schedule, payment schedule and values, and equipment were the same as the scheme used in the preliminary study of the dwell-free methods (see Section 6.3.3).

6.4.3 Results of Phase 1

Text entry rate (WPM)

Figure 6.12 (left) shows the average entry rate obtained per condition per session. Filteryedping was an average 37% faster than AltTyping across the 6 sessions.

A repeated measures analysis of variance was conducted to analyze the impact of Technique and Session on the typing rate. There was a significant main effect for Technique ($F_{1,60} = 39.49, \eta^2 = 0.335, p < 0.001$); and a significant main effect for Session ($F_{5,60} = 2.57, \eta^2 = 0.109, p = 0.04$); but no significant interaction between Technique and Session ($F_{5,60} = 1.10, \eta^2 = 0.047, p = 0.37$). These results suggest that Filteryedping is faster than AltTyping, and that users become faster with practice.

Note that the AltTyping text entry rates measured in this study are lower than the ones reported by [Räihä and Ovaska, 2012]. We believe the difference lies in the way they analyzed the data, where their analysis focused on provided an understanding of potential expert and error-free performance.

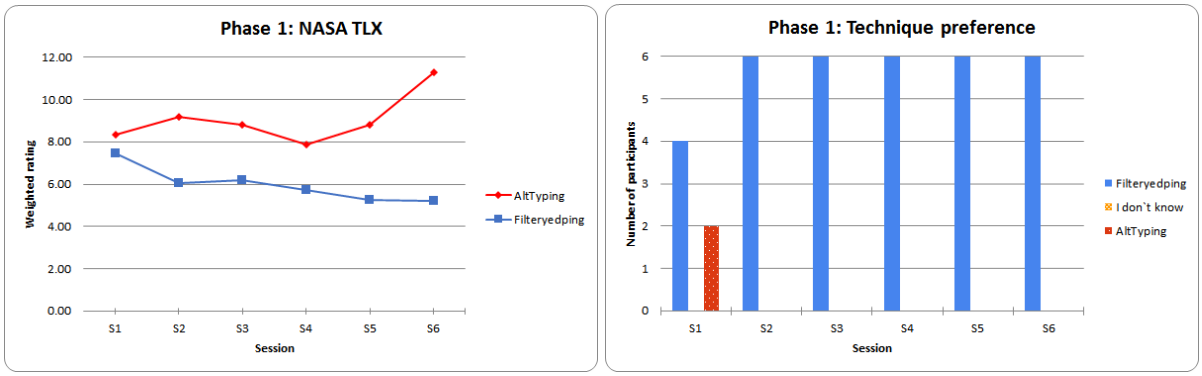


Figure 6.13: Subjective results collected about AltTyping and Filteryedping: NASA TLX weighted ratings (left) and responses to the question: "Which of the two techniques would you prefer to use?" after each session (right).

MSD error rate

Figure 6.12 (right) shows the average MSD error rate obtained per condition per session. AltTyping led to an average of 5 times more errors than Filteryedping across the 6 sessions.

A repeated measures analysis of variance was conducted to analyze the impact of Technique and Session on the MSD error rate. There was a significant main effect for Technique ($F_{1,60} = 8.95, \eta^2 = 0.120, p = 0.004$); but no significant main effect for Session ($F_{5,60} = 0.65, \eta^2 = 0.044, p = 0.66$); and no significant interaction between Technique and Session ($F_{5,60} = 0.47, \eta^2 = 0.031, p = 0.80$).

NASA TLX

Figure 6.13 (left) shows the average NASA TLX rating obtained per condition per session. For Session 1, the average for Filteryedping and AltTyping were 7.57 and 10.70, respectively. For Session 6, the averages were 4.87 and 11.80. The results suggest that the workload of typing with a filtering-based technique is lower than the workload of typing with a dwell-based technique.

Technique preference

In the post-test questionnaire, we asked the participants: "Which of the two techniques would you prefer to use?" Figure 6.13 (right) shows the participants' answer per session. Except in the first session, preference for Filteryedping was unanimous.

6.4.4 Phase 2

It has been previously demonstrated that studies of gaze interaction techniques intended for people with physical disabilities involving only participants without disabilities have a compromised ecological validity [Istance et al., 2012]. Thus, in Phase 2, we performed an iterative design and evaluation of the Filteryedping prototype with participants with ALS and DMD.

Procedure

In Phase 2, we aimed to understand how well the input performance of our dwell-free eye-typing approach generalizes when it is used by participants with motor impairments in natural settings. We conducted this phase of the study in each participant's home. Additionally, we wanted to learn about challenges that participants face when using a dwell-free eye-typing approach and to explore ways of addressing those issues. Thus, we divided Phase 2 into two sub-phases and two case studies.

In the two sub-phases, we asked these participants to type phrases using both Filteryedping and AltTyping. However, we expect that most, if not all, participants would have some experience with eye-trackers and dwell-based text entry methods. Because it would not be possible to collect data about how Phase 2 participants would learn to use and improve with both dwell-based and dwell-free eye-typing methods overtime, we included AltTyping only as a reference technique to help us understand how fast participants can currently type with a dwell-based method. As a result, these sub-phases did not consist of a fixed number of sessions as we had asked of Phase 1 participants. Still, this approach allowed us to understand how participants performed with and felt about a dwell-free eye-typing method, and provided us with an opportunity to identify major challenges that they had when using Filteryedping. We then conducted two case studies, in which we explored how to address these challenges and directly tested those solutions with the participants.

Thus, Phase 2 consisted of the following sub-phases and case studies:

1. *Sub-phase A*: Phase 2 began with three participants (P1–P3) using the same version of Filteryedping software that was used in Phase 1.
2. *Sub-phase B*: Based on results learned in the *Sub-phase A*, we improved the system by adding two key features: 1) a *focus dwell* time which allows the system to determine when the user has changed focus between the keyboard and the candidate list, and 2) a *slow movement threshold* time which allows the system to determine if the user is performing a saccade with slow eye movements or not. We also improved the prototype by adding auditory feedback and separation between the options in the vertical menus, and by not clearing anymore the stream of letters when the user looked outside the keyboard. We evaluated these changes to the system with participants P3–P6 in this sub-phase.
3. *Focus Dwell Case Study*: In this case study, we explored the effect of different values for the focus dwell parameter. We wanted to gain a better understanding of the relationship between this value and the different quality of calibration that users might have with an eye-tracker. The *Focus Dwell Case Study* was conducted with P5 and P6.
4. *Slow Movement Threshold Case Study*: In this case study, we examined the effect of different values for the *slow movement threshold* parameter. We wanted to gain a better understanding of how a well-adjusted threshold affects the input performance of users with slow eye movements. The *Slow Movement Threshold Case Study* was conducted with P3.

AltTyping was used in the two sub-phases the same way as in Phase 1. However, we used the dwell time of the end of the first session as reference. In the *Sub-phase A*, a reduction of 40 ms after each session was applied. In the *Sub-phase B*, we used a reduction rate of 10% per session. This change was employed when we realized that a reduction by 40 ms was too small when applied to the large dwell values used by motor-disabled participants.

Participants

In total, we recruited 6 participants (2 females), 4 of them with ALS and 2 with MDM, in different stages of the disease. All were fluent in English. Below, we provide a brief description of each participant based on the observation of the researchers during trials. For each participant, we also present in parenthesis their rating under the "Speech" sub-scale of the revised ALS functional rating scale (ALSFRS-R) [J. M. Cedarbaum et al., 1999]. Possible values for the sub-scale are: 4 – Normal speech processes; 3 – Detectable speech disturbance; 2 – Intelligible with repeating; 1 – Speech combined with non-vocal communication; and 0 – Loss of useful speech.

P1 is a woman in her sixties, living with ALS for decades. She can still communicate with her daughter by muttering (Speech rating: 0). She also uses an AAC device controlled by a button in one of her feet. She has had about 1–3 hours experience with eye-trackers before this study. She uses glasses and had some difficulties calibrating the eye-tracker. The layout of her keyboard was organized by frequency of use. As a result, she complained about having to look for the letters in a QWERTY layout. Because of her low familiarity with the QUERTY layout, she took part only in one session in *Sub-phase A*.

P2 is a 62-year-old female whose ALS onset occurred in 2003. She can still communicate verbally with difficulty (Speech rating: 2), but could not move anything below the neck. She was very familiar with eye-trackers. She did not use glasses and was able to successfully calibrate the eye-tracker easily. She took part in 4 complete sessions in *Sub-phase A*.

P3 is a 45-year-old male whose ALS onset occurred in 2005. Among the first three participants, he was the participant who showed the most physical debility in general. He currently uses an AAC (Augmentative and Alternative Communication) device controlled by an eye-tracker. During *Sub-phase A*, he communicated with us most of the time by giving yes or no answers with his eyebrows (Speech rating: 0). He uses glasses and had some difficulties calibrating the eye-tracker. He completed six sessions in *Sub-phase A*, six sessions in *Sub-phase B*, and eight sessions in the *Slow Movement Threshold Case Study*. Forty-eight days elapsed between his last session in *Sub-phase A* and his first session in *Sub-phase B*. It could be noticed that the disease had progressed a bit during this period. An indication of the progression was that he had stopped being able to use his eyebrows to answer yes or no questions and started using discrete cheek movements.

P4 is a 76-year-old male whose ALS onset occurred in 2010. He could still produce sounds, but lost the ability to produce useful speech (Speech rating: 0). He still maintained good

Table 6.1: Summary of the description of participants of Phase 2.

Participant	Age	Gender	Disease	Speech rating	Previous experience with eye-trackers	Quality of calibration with eye-trackers	Corrective vision	Participation
P1	60s	F	ALS	0	1–3 hours	Some difficulties	Glasses	1 session in *Sub-phase A*
P2	62	F	ALS	2	Very familiar	Good results	None	4 sessions in the *Sub-phase A*
P3	45	M	ALS	0	Very familiar	Some difficulties	Glasses	6 sessions in *Sub-phase A* 6 sessions in the *Sub-phase B* 8 sessions in the *Slow Movement Threshold Case Study*
P4	76	M	ALS	0	1–3 hours	Great difficulty	Glasses	1 session in the *Sub-phase B*
P5	33	M	DMD	2	Never used	Very good results	None	8 sessions in *Sub-phase B* 2 sessions in the *Focus Dwell Case Study*
P6	37	M	DMD	2	Never used	Some difficulties	Glasses	6 sessions in the *Sub-phase B* 1 session in the *Focus Dwell Case Study*

movements of his hands, which allowed him to control his wheelchair and use a tablet. He uses an AAC software system on his tablet to communicate. He has had about 1–3 hours experience with eye-trackers before this study. He uses bifocal glasses and had a great difficulty calibrating the eye-tracker. In fact, he successfully completed only one session in *Sub-phase B* because of this problem. Two other session attempts did not materialize because of the eye-tracker was unable to successfully calibrate with and track his gaze.

P5 is a 33-year-old male, with DMD. He can talk, however his voice is weaker than that of most people and fails sometimes (Speech rating: 2). He does not use corrective lenses and was able to achieve very good calibration results with the eye-tracker. He had never used an eye-tracker before the study. He uses a head mouse every day, which allows for general computer use, such as sending e-mails, navigating the web, and playing games. He completed eight sessions in *Sub-phase B* and two sessions in the *Focus Dwell Case Study*.

P6 is a 37-year-old male, with DMD. He is still able to talk, but must often repeat what he wants to say (Speech rating: 2). He uses glasses and had some difficulties calibrating the eye-tracker. He can still issue some commands to his wheel chair using his right hand, such as leaning it forward and backward without any help. He had never used an eye-tracker before the study. He uses a head mouse every day, primarily to play computer games. He completed six sessions in *Sub-phase B* and one session in the *Focus Dwell Case Study*. A summary of the descriptions is presented in Table 6.1.

Findings from *Sub-phase A*

P1 achieved an entry rate of 3.26 wpm with AltTyping and 0.95 wpm with Filteryedping, and a MSD error rate of 1.67% with AltTyping and 7.90% with Filteryedping. Normally, P1 uses a keyboard layout based on the frequency of use of the letters. As a result, she found difficulty with using a QWERTY based layout. Unfortunately, we were not prepared to quickly adjust the keyboard layout of our prototype and so she only participated in one session. Despite her performance results, among these two techniques, she felt that she preferred Filteryedping over AltTyping.

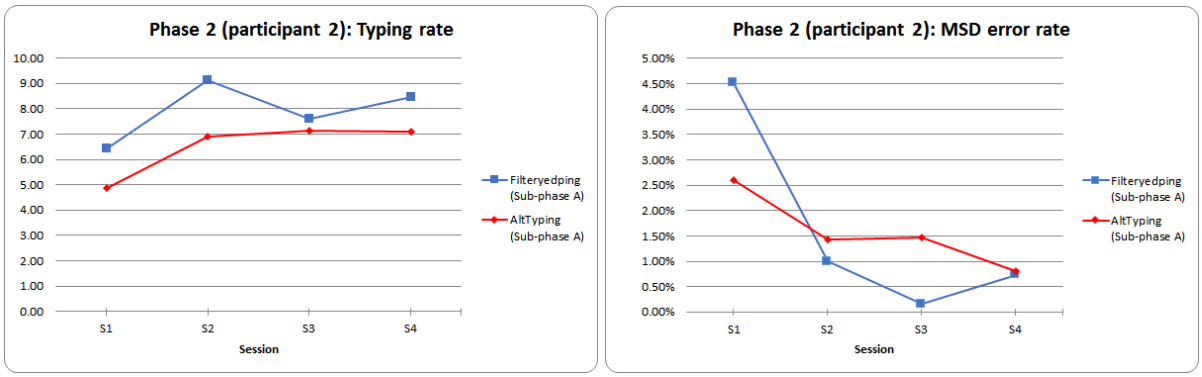


Figure 6.14: Results of participant 2 of Phase 2: text entry rate (left) and MSD error rate (right).

In the four sessions that P2 took part, she achieved an average entry rate of 6.50 wpm with AltTyping and 7.90 wpm with Filteryedping. Her average MSD error rate was 1.57% with AltTyping and 1.60% with Filteryedping. Figure 6.14 shows her WPM and MSD error rate. In the post-test questionnaire, she showed a preference for using Filteryedping in the first two sessions and AltTyping in the last two sessions.

The advanced stage of P3's disease impacts the movements of his eyes. He has saccades with longer duration and slower velocity, which is a reported symptom for some ALS patients [Leveille et al., 1982]. With a dwell-based eye-typing technique, this problem can be overcome by using a higher dwell time. As a result, he finished his first session with AltTyping configured to use 1120 ms dwell time, but was able to type with the system. The impact of this condition on Filteryedping proved to be a bigger challenge. While typing the word "appointment", for instance, during the saccade from the 'a' to the 'p', the letters 's', 'd', 'f', 't', 'y', 'h', 'j', 'u', 'i', and 'o' were also included in the stream. A user without disabilities is capable of "jumping" from the 'a' to the 'p' without selecting all of those letters. Naturally, because of the longer stream to be filtered, more candidate words can be suggested and the quality of the results is negatively affected. The average avgPos for his Filteryedping sessions was 6.66, almost twice the average of avgPos for users from Phase 1. In all six sessions, he had an average of 2.47 wpm using Filteryedping and 4.05 wpm using AltTyping, and an average MSD error rate of 9.12% using Filteryedping and 3.11% using AltTyping. The NASA TLX weighted rate was 8.82 for Filteryedping and 6.77 for AltTyping, indicating that he considered the workload of the task of typing with Filteryedping higher. However, contradicting these metrics, his preference showed a trend toward Filteryedping. He preferred AltTyping in the first session, had no preference in the second and third sessions, and preferred Filteryedping in the last three sessions. Figure 6.15 shows his WPM and MSD error rate with the initial prototype.

From this sub-phase, we learned about three aspects of the prototype that needed improvements:

1. Some users may be familiar with or have a preference for alternate keyboard layouts. Thus, it is important for the prototype to allow the user to customize the layout. Because only a small number of people use alternate layouts, this change to the application was not a

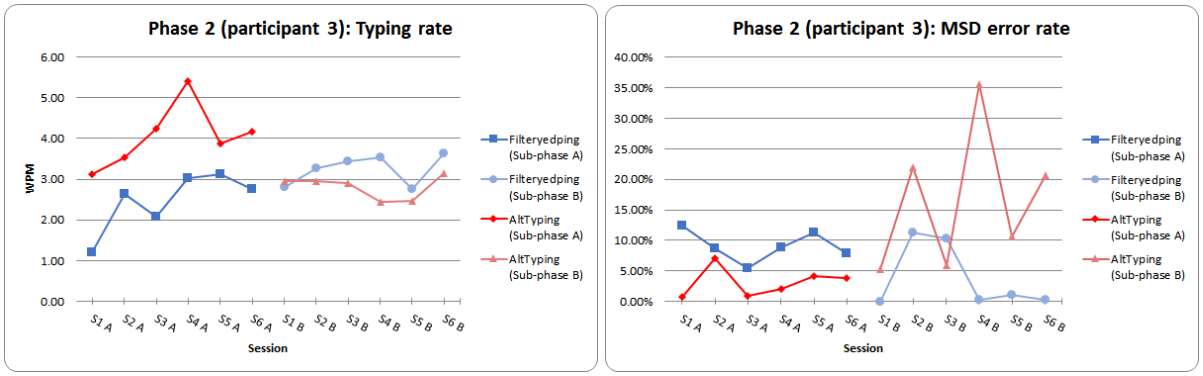


Figure 6.15: Results of participant 3 of Phase 2: Text entry rate (left) and MSD error rate (right).

high priority. After the conclusion of Phase 2, we evolved the prototype to include the possibility of using the same keyboard layout that P1 uses. We did this as a proof of concept, to assure that different keyboard layouts could also be used with Filteryedping technique. In about 100 minutes of typing, in blocks of 5 minutes, the first author reached 10.04 wpm with the new layout, which was completely new to him.

2. A clear issue with eye-trackers that we learned about is that the calibration quality varies a lot from user to user (see Table 6.1). Low calibration quality results in excessive noise in the tracker data for some users and is an important source of errors in the prototype. While using Filteryedping, an error often occurs while participants type a letter located in the third line of the keyboard. The tracker sometimes mistakenly detects a single gaze point in the area of the candidate list, causing the system to mistake that the user has selected a candidate word. Another problem often occurs while participants select a word in the candidate list. If the tracker detects a single gaze point in the keyboard area, the currently selected word is written and the candidate list disappears. Because it is not the desired word, the participant must then delete the word and type the desired word again. Many of these errors can be detected in the log files. If the participant's gaze stays in the candidate list for only a very short period of time, it means that it was an involuntary activation. We use a time threshold of 200 ms because the average reading speed is 200–240 milliseconds/word [Just and Carpenter, 1987]. Thus, it should represent the shortest amount of time possible where the user invokes the candidate list, sees that the word selected is what she wants, and continues to type. The candidate list was open for less than 200 ms 38.0% of the time. Noisy eye-tracker data inside the keyboard did not cause a problem, because extra letters included in the stream are simply filtered away by our algorithm later. Similarly, noisy eye-tracker data inside the candidate list is not a strong problem either because of the spaces included between the suggestions. As a result, we introduce a short *focus dwell* time to help the system determine a switch in focus from the keyboard to the candidate list and vice versa.
3. The saccades with longer duration and slower velocity which we observed from P3 motivated us to introduce a *slow movement threshold*. In analyzing our logs, we learned

that while P3's gaze moved from one desired letter to the next, his slow eye movements caused the eye-tracker to regularly detect gaze points between them. This could be verified by comparing, for the three participants, the average number of letters added to the stream of letters that is filtered for every typed word. P3 gazed on average at 44.4 letters per word, while P1—who is not familiar with the QWERTY layout—and P2 gazed 34.6 and 16.6 letters per word respectively. The logs also showed that number of gaze points at the desired keys is higher than the gaze points at keys that are crossed while the eye moves toward the target. This means that a small threshold value can be used to distinguish a letter that should be added to the stream of letters that is filtered from those gazed at because of slow eye movements. This slow movement threshold differs from a regular dwell in several ways: 1) the slow movement threshold is an implicit mechanism to differentiate gaze points detected because of slow eye movements between fixation points, while a regular dwell is an explicit user action to indicate a selection at a fixation point, 2) it can be set to be much shorter than a regular dwell (and imperceptibly fast) because the penalty for falsely adding letters to the stream of letters to be filtered is mitigated by our Filteryedping algorithm.

Findings from *Sub-phase B*

Based on findings from *Sub-phase A*, we modified the prototype to improve it. We used the version described in Section 6.3.1, adding the focus dwell and slow movement threshold features described above. To calculate how long the slow movement threshold should be for each user, we measure the amount of time the user takes to alternate her gaze between two points located about 25 cm apart from each other for 3 times (to travel about 75 cm). We performed this measurement before the first practice block in the first session for each user. It resulted in a slow movement threshold of 167 ms (5 points reported by the eye-tracker) for P3, 67 ms (2 points) for P4, and no slow movement threshold time (a single point adds the letter to the stream) for P5 and P6. The focus dwell time that we used for all participants was 100 ms, which is the time it takes for our 30 Hz eye-tracker to report 3 points.

It is hard to directly compare the results from *Sub-phase B* with results from *Sub-phase A* because of the small number of participants involved and the big variability among them. However, we believe, based on what we observed, that the inclusion of the focus dwell, the slow movement threshold, and other changes in the prototype improved the user experience.

One indication of the improvement of the Filteryedping prototype comes from the results of P3. He was the only participant that took part in both *Sub-phases A and B* (6 sessions each). In *Sub-phase B*, the entry rate achieved by him using AltTyping was 2.81 wpm, about 70% of what he obtained with AltTyping in the *Sub-phase A*. His MSD error rate was 16.69%, more than 5 times his error rate in *Sub-phase A*. The decline in the results was caused by the disease progression in the 48 days separating his last session in *Sub-phase A* and his first session in *Sub-phase B*, as he indicated to us. Even with a slower eye movement, he was able to improve

his typing rate and reduce his error rate using Filteryedping (from 2.47 wpm and 9.12% to 3.24 wpm and 3.86%, respectively). He finished his first session with AltTyping using a regular dwell of 1240 ms and followed the scheduled reduction of 10% each session, leading to a dwell of 910 ms in the fourth session. That was already too fast for him and we decided for the last two sessions to repeat the dwell times with which he achieved better performance in *Sub-phase A*: 1040 ms and 1000 ms. Figure 6.15 shows a per session comparison of those two metrics. In this sub-phase, he preferred Filteryedping in all the sessions. Analysis of the responses to the NASA TLX form also indicates a lower workload when using Filteryedping (6.91) compared to AltTyping (9.87).

In the one session P4 took part, he achieved an entry rate of 0.82 wpm for AltTyping and 0.63 wpm for Filteryedping, and a MSD error rate of 4.03% for AltTyping and 36.38% for Filteryedping. Despite this, he declared that, among these two techniques, he would prefer to use Filteryedping. He finished the AltTyping session with the dwell time set at 1120 ms. An issue that he faced was the small font size used in the interfaces. He sometimes adjusted the position of his head in order to be able to read what was on the screen when using both prototypes. Bigger font sizes would have been welcomed.

For P5, he was most comfortable with his head position tilted a little to the left. Surprisingly, the difference in the height of his two eye positions did not prevent the eye-tracker from being able to track his eye gaze well. In eight sessions, he achieved an average entry rate of 10.61 wpm for AltTyping and 11.40 wpm for Filteryedping. He finished his first session with AltTyping using a dwell time of 620 ms and followed the scheduled reduction of 10% each session, leading to a dwell time of 300 ms in the eighth session. Figure 6.16 (left) shows his WPM in *Sub-phase B*. His performance in the fifth session might have been affected by low concentration caused by conversations with a personal visitor during the session. His average MSD error rate was 0.30% for AltTyping and 0.45% for Filteryedping. He preferred AltTyping in Sessions 1, 2, 5 and 6, and Filteryedping in Sessions 7 and 8. He had no preference in Session 3 and did not provide a response to this question in Session 4. Analysis of the responses to the NASA TLX form indicates an equivalent workload when using Filteryedping (4.03) and AltTyping (4.25).

In six sessions, P6 achieved an average entry rate of 3.08 wpm for AltTyping and 6.49 wpm for Filteryedping, and a MSD error rate of 0.23% for AltTyping and 1.76% for Filteryedping. He finished his first session with AltTyping using a regular dwell time of 840 ms and followed the scheduled reduction of 10% each session, leading to a dwell time of 500 ms in the sixth session. Figure 6.16 (right) shows his WPM in *Sub-phase B*. From sessions 2 to 6, he expressed a preference for using Filteryedping over AltTyping. He had no preference in the first session. Also for P6, NASA TLX results indicated an equivalent workload; both techniques received a weighted rating of 3.81. From this sub-phase, we learned the following:

1. Including the focus dwell reduced the occurrence of problems introduced by noisy eye-tracker data. In particular, the candidate list was opened for less than 200 ms only 0.3% of the time—it was 38.0% in *Sub-phase A*. Although it especially helped participants who

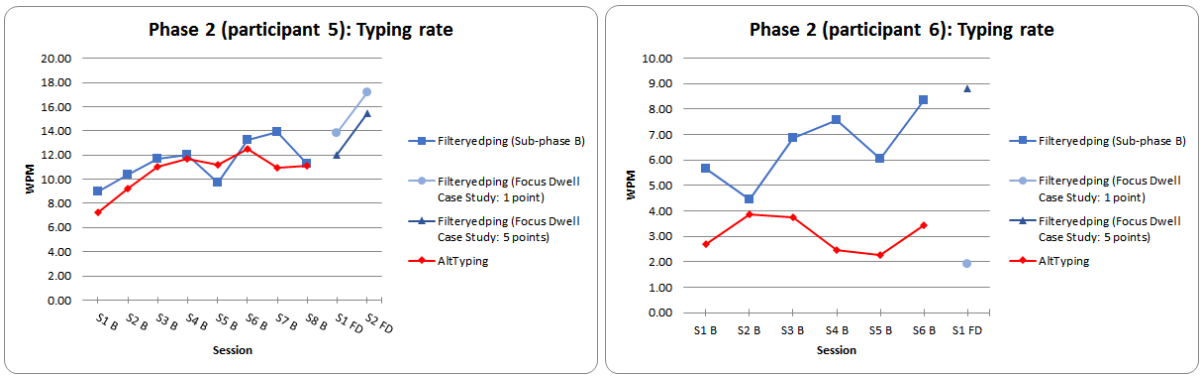


Figure 6.16: Text entry rate of participants 5 (left) and 6 (right) of Phase 2. *Sub-phase B* used a value of about 100 ms for the focus dwell parameter, which is the time it takes for the eye-tracker to report 3 points. *Focus Dwell Case Study* used a single point or 5 points (167 ms).

had trouble calibrating the eye-tracker well, it did not benefit those who already had good calibration results. This is perhaps because the dwell time required to change focus is reducing their input speed.

2. Including the slow movement threshold parameter clearly helped P3—the only participant with slow eye movements. His input performance continued to improve with Filteryedping, but declined with AltTyping as his ALS condition progressed. An analysis of our log showed that with the introduction of the slow movement threshold parameter, the average number of gazed letters per word was now 19.5 for P3. This is smaller than the 44.4 letters per word for P3 from *Sub-phase A* and the 34.6 letters per word for P1 from *Sub-phase A*—who is unfamiliar with QWERTY. However, this value is still higher than the 16.6 letters per word for P2 from *Sub-phase A* and 16.4 letters per word for the other participants from *Sub-phase B*. This suggests that the threshold value used for P3 still could be adjusted and perhaps that could improve his input performance even more.

Findings from the *Focus Dwell Case Study*

Results of the *Sub-phase B* provided good indication that the focus dwell helps to minimize the impact of poor calibration quality with the eye-tracker in the Filteryedping interface. At the same time, however, it seemed to stifle the input performance of those who obtained good accuracy and precision after successfully calibrating the eye-tracker. To further validate this, we performed a case study with P5 and P6 using Filteryedping with different focus dwell values. While P5 usually gets good calibration results with the eye-tracker, P6 experiences poor calibration, as shown in Figure 6.17. The use of different focus dwell values with participants who have different calibration quality with the eye-tracker would allow us to evaluate the relationship between the parameter and calibration quality.

In *Sub-phase B*, a focus dwell was performed when the eye-tracker reported 3 consecutive gaze points in either the keyboard area or the candidate list. In this case study, we compared the use of a single gaze point reported by the eye-tracker for the focus dwell (which is the

Figure 6.17: Cropped screenshots of the calibration check screen of participants 5 (left) and 6 (right) of Phase 2. The scattered points obtained by P6 indicate a lack of precision of the eye-tracker after his calibration. Displacement of the points regarding the target also indicates low accuracy, in this case.

configuration used by participants in *Sub-phase A*) against the use of a five gaze points. Both configurations were used in each session. We expected that P5 would have better results with the single-point configuration, because he would be able to avoid the delay required for changing focus between the keyboard and the candidate list. On the other hand, we expected that P6 would have better results with the 5-points configuration, because it would reduce the amount of errors caused by lower accuracy at which the eye-tracker detected his gaze in comparison to P5.

P5 took part in two sessions. In the first session, he started with the 1-point configuration. The opposite ordering was used in the second session. After the first session, he said he could not notice the difference between the two configurations. After the second session he said that the difference was perceptible and that he preferred using the 1 gaze point focus dwell configuration. In both sessions, he was faster with the 1-point than the 5-points configuration, as we had expected. He reached an average of 15.54 wpm using the 1-point configuration and an average of 13.77 wpm using the 5-points configuration (Figure 6.16 (left)). In *Sub-phase B*, P5 reached a max average of 13.92 wpm (session 7) using the 3-points configuration. Thus, for P5, when the focus dwell time is reduced, he is able to input text faster.

P6 took part in only one session, but the difference in the results was even more evident. He started with the 1-point configuration and reached 1.91 wpm using it and 8.82 wpm using the 5-points configuration (Figure 6.16 (right)), and indicated a clear preference for the 5-points configuration. In *Sub-phase B*, P6 reached a max average of 8.36 wpm (session 5) using the 3-points configuration. Thus, his results were also in agreement with what we had expected.

From *Sub-phase B* and this case study, we learn that dwell-free text input methods which have an input area and a candidate list must take into consideration the effect of the different levels of precision and accuracy at which eye-tracker can detect the user's gaze. When the eye-tracker detects the user's gaze with low precision and accuracy, gaze points near the border between the input area and the candidate list can falsely change the focus between these two components.

Figure 6.18: Optional animation indicating the timing of the gaze over a letter in relation to the slow movement threshold.

One way to address this issue is increase the distance between these components; however, this means the interface for the text input method would need to consume more screen space. A different approach is to introduce a focus dwell. Findings from *Sub-phase B* demonstrate that this can reduce false changes in the focus between these two components. Findings from this case study further validate this approach. However, it also points out that the mechanisms for addressing this problem should be tailored to the level of precision and accuracy at which the eye-tracker detects the user's gaze. When the eye-tracker can detect a user's gaze with high precision and accuracy, adding a large distance between these two components or introducing a long focus dwell time can lower the user's input performance. Thus, future research should explore ways to customize the interface of dwell-free text input methods based on the level of precision and accuracy at which the eye-tracker is detecting a user's gaze.

Findings from the *Slow Movement Threshold Case Study*

The introduction of a slow movement threshold seemed to allow P3 to improve his input performance during *Sub-phase B*. However, while observing him type, we noticed that the dwell time calculated by our simple method of measuring the amount of time the participant takes to alternate his gaze between two points located about 25 cm apart from each other for 3 times may not have been a large enough value. For P3, this method returned a value of 5 gaze points. At that threshold value, there were still many letters being included in the stream whenever he performs a saccade to a distant letter. Thus, we performed a case study with P3 exploring the effect of different slow movement threshold values. P3 was the only participant with a slow eye movement, and thus was the sole participant in this case study. After testing the use of large values as the slow movement threshold on ourselves, we learned that perhaps additional visual feedback would be useful. Thus, we implemented an animation, very similar to the one used in AltTyping, to indicate the timing of the gaze over a letter as it relates to the slow movement threshold (Figure 6.18). When the animation completes a 360^{o} arc, the whole key becomes pink, as before, and the letter is included in the stream. Nothing changed with the animation for the regular dwell.

In this case study, we adjusted the threshold value based on his feedback after each session. Each session consisted of a single block and the block duration was reduced from 20 to 12 minutes to avoid fatigue. A break of at least 30 minutes was taken between the sessions. The case study occurred over two consecutive days, with four sessions completed per day. Table 6.2 presents all the configurations (threshold values and presence of animated feedback or not) that

Table 6.2: Configurations (slow movement threshold values and presence of animated feedback or not) tested and typing rates obtained by P3 during the *Slow Movement Threshold Case Study*.

Session	Configuration	WPM	Letters added to filtered text stream per word	Options presented to the participant for the next session with P3's preference highlighted in bold
1	700 ms (21 gaze points) with feedback	3.23	8.6	1) 700 ms without feedback 2) A slower threshold with feedback **3) A faster threshold with feedback** And then: 1) 600 ms with feedback **2) 500 ms with feedback** 3) 400 ms with feedback
2	500 ms (15 gaze points) with feedback	3.79	12.94	1) 500 ms without feedback **2) 600 ms with feedback** 3) 400 ms with feedback
3	600 ms (18 gaze points) with feedback	2.96	12.42	1) 600 ms or slower threshold **2) 500 ms or faster threshold** And then: **1) 500 ms without feedback** 2) 400 ms with feedback 3) 400 ms without feedback
4	500 ms (15 gaze points) without feedback	3.65	10.09	**1) 400 ms without feedback** 2) 400 ms with feedback
5	(The wrong configuration was tested) 1 gaze point without feedback	1.89	31.52	
6	400 ms (12 gaze points) without feedback	3.39	12.11	1) 500 ms or slower threshold **2) 400 ms or faster threshold** And then: 1) 400 with feedback **2) 300 ms without feedback** 3) 433 ms without feedback 4) 466 ms without feedback
7	300 ms (9 gaze points) without feedback	3.72	17.09	1) Faster than 300 ms **2) Slower than 300 ms (but faster than 400 ms)** 3) Slower than 400 ms And then: **1) 333 ms without feedback** 2) 367 ms without feedback
8	333 ms (10 gaze points) without feedback	2.04	14.92	

P3 tested, the typing rates obtained and the options for the next configuration that were presented to him after each session.

Following this iterative evaluation process, we learned that for P3 a slow movement threshold of about 300–400 ms (9–12 points) without additional visual feedback would allow him to achieve a comparable input rate to what he obtained with a larger slow movement threshold value. In the last 3 sessions, the average number of gazed letters per word was 14.6. His preference was for a shorter threshold value without visual feedback. After the last session, we let him use the prototype again to give us a comment. He typed: "I really liked the software." Additional research effort is needed to develop methods to automatically characterize eye movement velocity and determine the appropriate slow movement threshold value.

6.5 Discussion

By running experiments with both people without disabilities and motor-disable individuals, we could verify that Filteryedping was a strong technique, both in terms of objective metrics and subjective metrics. Studies with participants without disabilities helped us collect an amount of data that would be difficult to obtain only with the target population. Motor-impaired participants

are hard to find and have many restrictions regarding schedule and location. At the same time, a small study with motor-impaired participants was important to include because it helped us understand how the results with participants without disabilities generalizes to the target user population. Furthermore, it helped us understand key issues that must be addressed in our prototype and explore solutions to those problems.

Along the study, we have identified several points that could be improved. An important observation was the difficulty participants faced in detecting an error in the typed text. As they use their eyes to type, constantly checking the results causes a slower typing rate. We perceived that the inclusion of an auditory feedback brought more confidence to users, letting them know that an error was committed without requiring their constant checking of the text area.

One of the biggest findings is that the calibration results vary a lot and are extremely user dependent. In general, participants who did not use corrective lenses seemed to achieve a better calibration than those with corrective lenses. In the preliminary study, two potential participants could not be included in the experiment because the eye-tracker could not be calibrated to work with those individuals well. One uses glasses and the other contact lenses. In the second phase, another participant (a glasses wearer) had similar problems. In the early experiments, we noticed that a good strategy for avoiding the selection of a wrong word accidentally would be to include some space to separate the options in the candidate list, even though this reduces the number of candidates shown per page. Later, we verified that the user was writing words accidentally even when she did not wanted to write a word, because excessive noise in the tracker data was causing the system to mistake that the user wanted to open the candidate list.

The variability in the precision and accuracy of the calibration and tracking must be taken into consideration in the development of any gaze-based interface. We included a focus dwell parameter in our prototype to help overcome the problem of selecting wrong words from the candidate list. A case study on the effect of the focus dwell parameter corroborates our belief that users with poor calibration results require longer focus dwell values. An automatic metric that indicates the quality of the calibration should be developed and be included to automatically configure the interface. Besides the focus dwell, a parameter that determines how close the keys on the keyboard should be to each other has the potential to improve users' performance. Users with good calibration results would benefit from the extra screen space enabled by a smaller virtual keyboard.

The precision and accuracy of gaze points reported by the eye-tracker was not the only source of variation among users. One of the participants with ALS demonstrated impairments in the velocity of his saccades. This type of problem may affect not only ALS patients [Leveille et al., 1982], but also DMD patients [Lui et al., 2001]. Ashtiani and MacKenzie [2010] previously mentioned this limitation while advocating the development of a text entry technique based on blinking instead of eye movements for severely motor impaired. However, the introduction of the slow movement threshold parameter seems to help with this problem. One of the advantages of the slow movement threshold solution is that it may be adjusted as the disease progresses,

requiring no abrupt changes in the interface to which the user is already familiarized. Furthermore, this parameter allows the system to differentiate slow eye movements from when the user's eye gaze has reached a target. Although from an external observer's perspective, it may seem similar to a normal dwell, from the perspective of a user with slow eye movement, this parameter is still smaller than what a normal dwell threshold value is for them. The mechanism itself does not require any explicit user action. As result, some letters along the eye movement towards the target letter might still be added to the text stream, our text-filtering based approach minimizes the effect of such errors and does not require the user to delete unintended letters. Thus, from the user's perspective, this input approach differs from a dwell-based approach. P3's preference for Filteryedping over AltTyping lies in the fact that he did not need to perform full dwells over keys that he wanted to enter. This difference also enabled him to be faster with Filteryedping.

We discussed with P5 and P6, the two participants who use a head mouse, if they think that Filteryedping could be used with a head mouse instead of with an eye-tracker. Both of them said that they think it would be possible. In a discussion with P5, we concluded that the dwell-based eye-typing could be even more integrated to Filteryedping. If a high threshold value is used for the normal dwell—two seconds, for example—we would practically eliminate the possibility of selecting a letter by mistake. Then, we could enable the real time dwell-based eye-typing; i.e., without having to search for the dwelled word in the first position of the candidate list.

Finally, the difference in performance between participants with and without disabilities needs to be examined. Perhaps one reason is the presence and progression of ALS or DMD. However, there were several other factors that could have contributed to the performance difference. One possible reason is the difference in the study conditions. Individuals without disabilities took part in a controlled laboratory study, while participants with ALS and DMD typed in their homes. This facilitated their participation, but at the same time hindered us from controlling for other factors, such as lighting, noise and other distractions. Another possible reason might be the difference in age. Motor-disabled participants were older (avg = 53.0 years old) than participants without disabilities (avg = 22.8 years old). However, despite the difference in the rate of typing, the results were consistent in showing that both participants with and without motor disabilities liked dwell-free eye-typing and performed better with a dwell-free method than with a dwell-based approach.

6.6 Summary and future directions

In an effort to increase the communication power of severely motor impaired individuals, we have introduced and evaluated a dwell-free eye-typing technique that allows the user to enter text without requiring a long fixation over a key to input that letter. With Filteryedping, the user types simply by gazing sequentially at all of the letters in a word. It overcomes the Midas' touch problem by filtering out letters that were gazed at accidentally by the user. Filteryedping then creates a ranked list of candidates based on the frequency and length of the words in a corpus.

Our first evaluation step was to compare it with another plausible way that dwell-free eye-typing could be implemented—a shape-based approach which has been shown as an effective typing method for touch-based interfaces. The shape-based technique identifies candidate words by comparing the shape of the path covered by the gaze with the optimal shape of each word in a dictionary. It has been suggested and investigated as a good candidate for eye-typing [Hoppe et al., 2013; Kristensson and Vertanen, 2012]. Objective and subjective results from a study with 12 users without disabilities testing our implementation of the two dwell-free techniques indicated that Filteryedping is a satisfactory method to support eye-typing.

We then evaluated the Filteryedping technique along with AltTyping, the fastest dwell-based eye-typing tool reported in the literature. The evaluation was divided into two phases, the first with 10 participants without disabilities and the second with 6 severely motor-disabled individuals. Results of the first phase show that Filteryedping enabled participants to reach an average of 14.75 wpm in six sessions (about 2 hours of typing per user) and an average of 15.95 wpm in the last session, and 10.77 wpm with AltTyping in six sessions and an average of 11.71 wpm in the fastest session (session 5, using a dwell time of 290 ms). The fastest participant typed at 19.28 wpm with Filteryedping in the sixth session. The fast typing rate with Filteryedping was not reached at the expense of accuracy. The average MSD error rate in six sessions was 0.64%. Even before the last improvements, Filteryedping fared better than AltTyping not only in these objective metrics, but also in terms of user's preference and workload. The second phase iteratively evolved the Filteryedping technique to address problems that we learned by evaluating the method with participants with ALS and DMD in natural settings. We implemented and conducted case studies of two important parameters (focus dwell and slow movement threshold), which allows the technique to be adapted to different user needs.

Our goal here was to examine how well dwell-free approach would work in practice against a dwell-based one. Our preliminary study was only done to identify which of our two implementation of possible dwell-free techniques could be used to test against AltTyping. The results show that dwell-free typing is an approach that could potentially be faster than dwell-based and furthermore is lower in workload and preferred by participants. Although we tested a key filtering–based approach against AltTyping, a shape-based approach can also be employed to support dwell-free eye-typing. Future work should further explore how to adapt SHARK^2 for eye-typing and evaluate it.

The positive results achieved with Filteryedping are encouraging, but the experiments reported in this chapter were only the first steps in the exploration of this new dwell-free eye-typing approach. We plan to investigate the robustness of a key filtering–based approach by:

- Adapting the word ranking algorithm used by Filteryedping to a different language (Portuguese), to check the impact of using a list of words with different characteristics in terms of length and frequency distribution;
- Implementing Filteryedping as a virtual keyboard module and integrating it in an operational system, so that we can observe its use in the context of an existing application;

- Evaluating the impact of using different eye-tracker models in the performance of Filteryedping users, so that challenges encountered when using simpler and cheaper eye-trackers can be overcome, which will allow the technique to be used by a larger population.

Chapter 7

Conclusion

In the previous chapters, we presented detailed discussions regarding specific scenarios. We start this chapter by recalling how we leveraged the capabilities of digital devices while investigating data input and content exploration in scenarios with restrictions. Then, we present for each scenario a synthesis of the contributions, limitations and future work. The dissertation ends with final remarks and the list of publications originated from the graduate study.

7.1 Leveraging the capabilities of digital devices

We started the dissertation by exploring the use of an interactive coffee table as a means of collaboratively exploring personal photos and videos in a typical leisure space. Interactive tables are a good alternative for media exploration, not only because of their large size and high resolution, but also because multi-touch interfaces fosters collaboration. Our interfaces were designed for a tabletop located in the center of the living room, with a couch set facing a television display. This allows the integration of the tabletop with the TV set.

However, a television set is useful not only as an output device for interactions occurring in a tabletop or other mobile devices. Interconnecting these devices with a TV set enable them to be used as input devices for interactive TV applications. In the second scenario, we enable iDTV applications to handle multimodal data generated from multiple devices. This possibility fosters the development of richer applications, including applications that require text input.

Considering current TV systems, however, the main interaction device is still the remote control. Furthermore, current solutions for text input are not satisfactory. Due to the importance of text input for applications, we focused in the third scenario on improving the task of text entry using a remote control.

In the last two scenarios, we continued investigating better solutions for text entry, but we shift our focus to the restricted motor capabilities of individuals with ALS or DMD. In the fourth scenario, we attached a smartphone in a foot of the participants and used the accelerometer to detect movements that are interpreted as characters according to a Morse-based codification. In

the last scenario we explored an eye-tracker and created a dwell-free eye-typing technique to improve the performance of people who write with the eyes.

7.2 Contributions, limitations and future work

7.2.1 Table Scenario

In Chapter 2, we present a discussion about the use of an interactive coffee table for exploration of personal photos and videos. Our study shed light on the scenario by showing the following:

- People liked the idea of sitting around a table and passing photos to family members and friends, however, it is necessary to complement the experience by offering some software support regarding the alignment and distribution of media items on the tabletop.
- The use of a TV helps creating an environment that supports several users and improves the experience by providing a better image quality and comfort.
- For storytelling scenarios, it seems that a single control panel is more appropriate. In a more explorative scenario, the use of a control panel for each two or three users may be the optimal configuration.
- Personal spaces would be very useful as a space that allows zooming and navigation capabilities without disturbing other users.

The main limitations of the study were the use of small media collections and a hardware offering low sensitivity. As future work, we suggest the development and evaluation of a tabletop hardware that allows it to be tilted and a prototype version that provides flexible use of control panels and personal spaces.

In the table scenario, the potential of a television set as an output device for tabletops and mobile devices could be verified. We also believe that the interconnection among devices can greatly improve the power of interactive TV applications, as explored in the Multimodal Scenario.

7.2.2 Multimodal Scenario

In Chapter 3, we propose an architecture for a component that offers to DTV applications the possibility of receiving multimodal data from multiple devices. The component facilitates the development of innovative applications, since it helps to break one of the main constraints of DTV application development, which is to rely solely on inputs from the remote control. Furthermore, it allows the integration of the STB with external input devices, which helps increasing the accessibility of Digital TV applications. An important feature is the adoption of previously established standards such as InkML, VoiceXML, ZeroConf and UPnP.

A limitation on the implementation of the component is the absence of APIs that enables Java and Lua applications to receive multimodal events. The current version supports only resident applications written in C++. New functionalities may also enhance the usability of the Event

Manager, such as automatically extracting metadata of the received content and including them in the multimodal event created, and allowing applications to listen only for events containing specific data types. To better validate the component, a more robust application that uses multimodal events should be implemented.

An implemented application to validate the component provides communication between users, allowing exchange of text messages and files. Our Remote Scenario also deals with the challenge of entering text in an interactive TV application.

7.2.3 Remote Scenario

In Chapter 4, we present an interface model based on multiple input modes to deal with limitations of text input in iDTV using a remote control. Our model is built according to the user-centered design methodology, so we give several quotes from potential users and experts in the area, having different specializations, to justify design decisions. The requirement elicitation highlights the importance of offering alternative ways of entering text, in order to meet the needs of different user profiles and to be flexible enough to be used in different environments. We implemented and presented a component prototype that offers a virtual keyboard mode, a phone keypad mode, and a speech mode.

Even though we realized that the use of the think aloud technique does not fit well in the case of voice interfaces, our comprehensive evaluation allowed us to detect various usability issues. However, the main limitation of the study was the absence of a prototype running on a set-top box, so that the user performance when a real remote control is used could be evaluated. As future work, the possible improvements and new requirements identified—such as more intuitive ways of accessing each mode and a clearer organization of the commands in the speech mode—should be tackled, and a high-fidelity prototype implemented.

In the Foot and Eye Scenarios, we continued investigating better ways to provide text entry, however, we consider a different type of restriction. We aimed to develop new techniques for severely motor-disabled individuals.

7.2.4 Foot Scenario

In Chapter 5, we propose a text entry method and an interaction technique for people with a severe motor disability, who keep at least a partial movement of a leg and a foot. We believe there was no text entry technique that leverage the capabilities of the target population. Our study concludes the following:

- Different people have different leg rest postures. Our evaluation indicates that a semi-automatic calibration should be offered to configure the angles used to trigger the symbols.
- A dynamic definition of the character timeout is important, because the high value required for avoiding errors in the beginning should smoothly fall to allow speed improvement as users gain experience.

- Although Morse code is a useful foundation for defining the codification, the lack of intuitiveness in the mappings from dots and dashes to internal and external rotations or blue and green areas overwhelms users. Codes should be defined in terms of user actions instead of dots and dashes.

The two major limitations of the study were the lack of experiments with motor-disabled individuals and the conduction of only one session per participant. As improvement directions for DuoGrapher we suggest the incorporation of predictive capabilities and the exploration of ambiguous methods. In another study direction, SwingingFoot could also be explored with different kinds of input. Although much slower than the eye-typing approach explored in the Eye Scenario, the combination of DuoGrapher and SwingingFoot might be the best option for those who cannot achieve satisfactory calibration with current eye-tracking technology.

7.2.5 Eye Scenario

In Chapter 6, we propose a dwell-free eye-typing technique for individuals with a severe motor disability. Filteryedping overcomes the Midas' touch problem by filtering out letters that the user mistakenly gazed at while sequentially looking for letters in the intended word. We implemented and evaluated two dwell-free eye-typing techniques: Filteryedping and a shape-based approach. Objective and subjective results from a study indicated that Filteryedping is a satisfactory method to support eye-typing. Future work should further explore how to adapt SHARK^2 for eye-typing and evaluate it.

We also evaluated Filteryedping along with AltTyping, the fastest dwell-based eye-typing tool reported in the literature. Even before the last improvements, Filteryedping fared better than AltTyping not only in these objective metrics, but also in terms of user's preference and workload. To allow two the technique to be adapted to different user needs, we also created two important parameters: focus dwell and slow movement threshold. The focus dwell parameter help overcome the problem of selecting wrong words from the candidate list, caused by the variability in the precision and accuracy of the eye-tracking. Calibration results vary a lot and are extremely user dependent. In general, participants who did not use corrective lenses seemed to achieve a better calibration than those with corrective lenses. A parameter that determines how close the keys on the keyboard should be to each other can also be explored in future work to overcome poor calibration results. The slow movement threshold parameter help overcome the problem of saccades with longer duration and slower velocity that affects ALS and DMD patients. One of the advantages of the slow movement threshold solution is that it may be adjusted as the disease progresses, requiring no abrupt changes in the interface to which the user is already familiarized.

To verify even further the robustness of a key filtering–based technique, researchers should also investigate the impact of its use in the context of a regular application—not only in a tool for performance measure—, with different languages, and with different models of eye-tracker.

7.3 Final remarks

The technology is constantly evolving and benefiting people in their daily activities. It extends our abilities to communicate, store and process data. Different devices help us in different tasks. Nowadays, a significant amount of the information we handle are stored in a digital format. By creating a rich digital ecosystem we are capable of exploring the advantages of each device type, what allows us to handle and create an increasing amount of data.

The more technology is created, the clearer becomes the potential of new technology on helping people to go further. This situation challenges us to create usable solutions for several new scenarios. In this dissertation, we report on our investigation of some of these challenges. More specifically, we are interested in data input or content exploration in scenarios with restrictions.

We explore each specific scenario in depth, which is important for a study to achieve relevant results and contributions. On the other hand, the wide scope of this dissertation allowed the student to learn about several technologies and techniques in the field of human-computer interaction. The accumulated experience will be important for the student's performance in future projects.

7.4 Publications

This section lists the publications originated from the graduate study. These are classified in directly related to the PhD research—cited in footnotes in the beginning of the chapters—and indirectly related, i.e., those originated from collaboration at research projects of members or partners of the research group, or class projects.

7.4.1 Directly related to the dissertation

Journal papers

1. **D. Pedrosa**, M. G. C. Pimentel, A. Wright, and K. N. Truong. Filteryedping: Design challenges and user performance of dwell-free eye-typing. Submitted to *ACM Transactions on Accessible Computing*, 2014.

Full papers in conferences

2. **D. Pedrosa** and M. G. C. Pimentel. Text entry using a foot for severely motor-impaired individuals. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, SAC '14, pages 957-963, New York, NY, USA, 2014. ACM. doi: 10.1145/2554850.2554948. URL http://doi.acm.org/10.1145/2554850.2554948.
3. **D. Pedrosa**, R. L. Guimarães, M. G. C. Pimentel, D. C. A. Bulterman, and P. Cesar. Interactive coffee table for exploration of personal photos and videos. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, pages

967-974, New York, NY, USA, 2013. ACM. doi: 10.1145/2480362.2480548. URL http://doi.acm.org/10.1145/2480362.2480548.

4. **D. Pedrosa**, J. A. C. Martins Jr., E. L. Melo, and C. A. C. Teixeira. A multimodal interaction component for digital television. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, SAC '11, pages 1253-1258, New York, NY, USA, 2011. ACM. doi: 10.1145/1982185.1982459. URL http://doi.acm.org/10.1145/1982185.1982459.
5. D. A. Vega-Oliveiros, **D. Pedrosa**, M. G. C. Pimentel, and R. P. M. Fortes. An approach based on multiple text input modes for interactive digital TV applications. In *Proceedings of the 28th ACM International Conference on Design of Communication*, SIGDOC '10, pages 191-198, New York, NY, USA, 2010. ACM. doi: 10.1145/1878450.1878483. URL http://doi.acm.org/10.1145/1878450.1878483.

Workshop papers

6. **D. Pedrosa**, J. A. C. Martins Jr., M. G. C. Pimentel, and E. L. Melo. Componente de Interação Multimodal no Ginga. In: *II Workshop de TV Digital Interativa* (WTVDI) do WebMedia '10. Belo Horizonte, MG. Proceedings of 16th Brazilian Symposium on Multimedia and the Web, v. 2, pages 197-202, 2010.

Demos in conferences

7. **D. Pedrosa**, M. G. C. Pimentel, and K. N. Truong. Filteryedping: A dwell-free eye-typing technique. Submitted to *Interactivity category of the Conference on Human Factors in Computing Systems (CHI '2015)*.
8. **D. Pedrosa**, D. A. Vega-Oliveiros, R. P. M. Fortes, and M. G. C. Pimentel. Text Input in Digital Television: a Component Prototype. In *Adjunct Proceedings of the Eighth European Conference on Interactive TV and Video*, EuroITV '10, pages 75-78, New York, NY, USA, 2010. ACM.

7.4.2 Indirectly related to the dissertation

Full papers in conferences

9. **D. Pedrosa**, R. L. Guimarães, P. Cesar, and D. C. A. Bulterman. Designing Socially-Aware Video Exploration Interfaces: A Case Study using School Concert Assets. In: *Proceedings of International Conference on Making Sense of Converging Media*, AcademicMindTrek '13, pages 110-117, New York, NY, USA, 2013. ACM. doi: 10.1145/2523429.2523454 URL http://doi.acm.org/10.1145/2523429.2523454.

Short papers in conferences

10. A. K. Gomes, **D. Pedrosa**, and M. G. C. Pimentel. Evaluating Asynchronous Sharing of Links and Annotation Sessions as Social Interactions on Internet Videos. In: *Proceedings of the 11th IEEE/IPSJ International Symposium on Applications and the Internet*, SAINT '11, pages 184-189, Munique, 2011.
11. B. C. R. Cunha, **D. Pedrosa**, R. Goularte, and M. G. C. Pimentel. Video annotation and navigation on mobile devices. In: *Proceedings of the 18th Brazilian symposium on Multimedia and the web*, WebMedia '12, pages 261-264, New York, NY, USA, 2012. ACM. doi: 10.1145/2382636.2382691. URL http://doi.acm.org/10.1145/2382636.2382691.
12. D. A. Vega-Oliveiros, **D. C. Pedrosa**, M. G. C. Pimentel, and R. Goularte. Navegação em Vídeo via Quadros Recentes. In: *Anais do XV Simpósio Brasileiro de Sistemas Multimídia e Web*, WebMedia '09, v. 2. pages 15-18, 2009.

Posters in conferences

13. **D. Pedrosa**, D. A. Vega-Oliveiros, and M. G. C. Pimentel. What do you want to watch (again)? Video Navigation Using Recency Frames. In: *Adjunct Proceedings of the Eighth European Conference on Interactive TV and Video*, EuroITV '10, pages 115-118, New York, NY, 2010. ACM.

Demos in conferences

14. **D. C. Pedrosa**, D. A. Vega-Oliveiros, M. G. C. Pimentel, and R. Goularte. Uma Aplicação NCL/Lua para Navegação em Vídeo via Quadros Recentes. In: *Workshop de Ferramentas de Aplicações* (WFA) do WebMedia '09, Fortaleza, CE. Anais do XV Simpósio Brasileiro de Sistemas Multimídia e Web, v. 2, pages 136-138, 2009.

Bibliography

T. Apted, J. Kay, and A. Quigley. Tabletop sharing of digital photographs for the elderly. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, CHI '06, page 781–790, New York, NY, USA, 2006. ACM. ISBN 1-59593-372-7. doi: 10.1145/1124772.1124887. URL http://doi.acm.org/10.1145/1124772.1124887.

B. Ashtiani and I. S. MacKenzie. BlinkWrite2: an improved text entry method using eye blinks. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ETRA '10, page 339–345, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-994-7. doi: 10.1145/1743666.1743742. URL http://doi.acm.org/10.1145/1743666.1743742.

Associção Brasileira de Normas Técnicas. *ABNT NBR 15606-2 – Digital terrestrial television – Data coding and transmission specification for digital broadcasting – Part 2: Ginga-NCL for fixed and mobile receivers – XML application language for application coding*. Nov. 2007. Corrected version 2 2009.04.17. Available at http://www.dtv.org.br/download/en-en/ABNTNBR15606_2D2_2007Ing_2008Vc2_2009.pdf.

A. Barrero, D. Melendi, X. G. Pañeda, R. García, and S. Cabrero. An empirical investigation into text input methods for interactive digital television applications. *International Journal of Human-Computer Interaction*, 30(4):321–341, Nov. 2013. ISSN 1044-7318. doi: 10.1080/10447318.2013.858461. URL http://dx.doi.org/10.1080/10447318.2013.858461.

S. Basapur, G. Harboe, H. Mandalia, A. Novak, V. Vuong, and C. Metcalf. Field trial of a dual device user experience for iTV. In *Proceddings of the 9th International Interactive Conference on Interactive Television*, EuroITV '11, pages 127–136, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0602-7. doi: 10.1145/2000119.2000145. URL http://doi.acm.org/10.1145/2000119.2000145.

N. Bee and E. André. Writing with your eye: A dwell time free writing system adapted to the nature of human eye gaze. In E. André, L. Dybkjær, W. Minker, H. Neumann, R. Pieraccini, and M. Weber, editors, *Perception in Multimodal Dialogue Systems*, number 5078 in Lecture

Notes in Computer Science, pages 111–122. Springer Berlin Heidelberg, Jan. 2008. ISBN 978-3-540-69368-0, 978-3-540-69369-7. URL http://link.springer.com/chapter/10.1007/978-3-540-69369-7_13.

M. Belatar and F. Coldefy. Sketched menus and iconic gestures, techniques designed in the context of shareable interfaces. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, page 143–146, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0399-6. doi: 10.1145/1936652.1936681. URL http://doi.acm.org/10.1145/1936652.1936681.

M. Belatar and F. Poirier. Text entry for mobile devices and users with severe motor impairments: handiglyph, a primitive shapes based onscreen keyboard. In *Conference on Computers and Accessibility*, ASSETS '08, pages 209–216. ACM, 2008. ISBN 978-1-59593-976-0. doi: 10.1145/1414471.1414510. URL http://doi.acm.org/10.1145/1414471.1414510.

R. Brandão, G. de Souza Filho, C. Batista, and L. Gomes Soares. Extended features for the Ginga-NCL environment: Introducing the LuaTV API. In *2010 Proceedings of 19th International Conference on Computer Communications and Networks (ICCCN)*, pages 1–6, 2010. doi: 10.1109/ICCCN.2010.5560066.

S. Buisine, G. Besacier, A. Aoussat, and F. Vernier. How do interactive tabletop systems influence collaboration? *Comput. Hum. Behav.*, 28(1):49–59, Jan. 2012. ISSN 0747-5632. doi: 10.1016/j.chb.2011.08.010. URL http://dx.doi.org/10.1016/j.chb.2011.08.010.

A. Carmichael, M. Rice, D. Sloan, and P. Gregor. Digital switchover or digital divide: a prognosis for usable and accessible interactive digital television in the UK. *Univers. Access Inf. Soc.*, 4(4):400–416, 2006. ISSN 1615-5289. doi: http://dx.doi.org/10.1007/s10209-005-0004-x.

S. J. Castellucci and I. S. MacKenzie. Unigest: text entry using three degrees of motion. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 3549–3554, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-012-X. doi: http://doi.acm.org/10.1145/1358628.1358889.

S. J. Castellucci and I. S. MacKenzie. Gestural text entry using huffman codes. In *Conference on Multimedia and Human-Computer Interaction*, MHCI '13, pages 1–8. International ASET, Inc., 2013. URL http://www.yorku.ca/mack/mhci2013e.html.

R. G. Cattelan, C. Teixeira, R. Goularte, and M. D. G. C. Pimentel. Watch-and-comment as a paradigm toward ubiquitous interactive video editing. *ACM TOMCCAP*, 4(4):1–24, 2008. ISSN 1551-6857. doi: http://doi.acm.org/10.1145/1412196.1412201.

T. Chakraborty, S. Sarcar, and D. Samanta. Design and evaluation of a dwell-free eye typing technique. In *Proceedings of the Extended Abstracts of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI EA '14, page 1573–1578, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2474-8. doi: 10.1145/2559206.2581265. URL http://doi.acm.org/10.1145/2559206.2581265.

S.-C. Chen, C.-Y. Chien, W.-M. Chang, and S.-W. Lin. A new assistive communication system for the serious disabled. In *Convention on Rehabilitation Eng. & Assistive Technology*, iCREATe '08, pages 59–64. START, 2008. URL http://dl.acm.org/citation.cfm?id=1983222.1983240.

Y.-X. Chen, M. Reiter, and A. Butz. PhotoMagnets: supporting flexible browsing and searching in photo collections. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*, ICMI-MLMI '10, page 25:1–25:8, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0414-6. doi: 10.1145/1891903.1891936. URL http://doi.acm.org/10.1145/1891903.1891936.

P. Chiu, J. Huang, M. Back, N. Diakopoulos, J. Doherty, W. Polak, and X. Sun. mTable: browsing photos and videos on a tabletop system. In *Proceedings of the 16th ACM international conference on Multimedia*, MM '08, page 1107–1108, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-303-7. doi: 10.1145/1459359.1459585. URL http://doi.acm.org/10.1145/1459359.1459585.

J. Cortez, D. A. Shamma, and L. Cai. Device communication: A multi-modal communication platform for internet connected televisions. In *Proceedings of the 10th European Conference on Interactive TV and Video*, EuroiTV '12, pages 19–26, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1107-6. doi: 10.1145/2325616.2325622. URL http://doi.acm.org/10.1145/2325616.2325622.

A. L. Cox, P. A. Cairns, A. Walton, and S. Lee. Tlk or txt? using voice input for SMS composition. *Personal and Ubiquitous Computing*, 12(8):567–588, November 2008. ISSN 1617-4909 (Print) 1617-4917 (Online). doi: 10.1007/s00779-007-0178-8.

P. César, D. Bulterman, and A. Jansen. An Architecture for End-User TV Content Enrichment. *Journal of Virtual Reality and Broadcasting*, 3(9), Dec. 2006. urn:nbn:de:0009-6-7594, ISSN 1860-2037.

M. Davies. The corpus of contemporary american english (COCA): 450 million words, 1990-2012, 2008. URL http://www.americancorpus.org.

L. C. de Miranda, L. S. G. Piccolo, and M. C. C. Baranauskas. Artefatos físicos de interação com a TVDI: desafios e diretrizes para o cenário brasileiro. In *Brazilian IHC 2008*, pages 60–69, 2008. ISBN 978-85-7669-203-4.

A. Dix, J. Finley, G. Abowd, and R. Beale. *Human-computer interaction (3nd ed.)*. Person Education Limited, Edinburgh Gate, Harlow, Essex CM20 2JE, England, 2004. ISBN 0130-461091.

S. Fei, A. Kerne, A. Jain, A. M. Webb, and Y. Qu. Positioning portals with peripheral NFC tags to embody trans-surface interaction. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, pages 317–320, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2271-3. doi: 10.1145/2512349.2514593. URL http://doi.acm.org/10.1145/2512349.2514593.

T. Felzer and R. Nordmann. Alternative text entry using different input methods. In *Conference on Computers and Accessibility*, Assets '06, pages 10–17. ACM, 2006. ISBN 1-59593-290-9. doi: 10.1145/1168987.1168991. URL http://doi.acm.org/10.1145/1168987.1168991.

L. Fissi. Man-in-the-barrel syndrome (MIBS). http://neuroradiologyonthenet.blogspot.com/2009/05/ man-in-barrel-syndrome-mibs.html, 2009. Accessed 17/Sep/2013.

K. Go, H. Konishi, and Y. Matsuura. Itone: a japanese text input method for a dual joystick game controller. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 3141–3146, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-012-X. doi: http://doi.acm.org/10.1145/1358628.1358821.

D. Grammenos, Y. Georgalis, N. Kazepis, G. Drossis, and N. Ftylitakis. The booTable experience: iterative design and prototyping of an alternative interactive tabletop. In *Proceedings of the 8th ACM Conference on Designing Interactive Systems*, DIS '10, page 272–281, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0103-9. doi: 10.1145/1858171.1858221. URL http://doi.acm.org/10.1145/1858171.1858221.

D. L. Grover, M. T. King, and C. A. Kuschler. Reduced keyboard disambiguating computer, 1998.

S. Harada, J. O. Wobbrock, J. Malkin, J. A. Bilmes, and J. A. Landay. Longitudinal study of people learning to use continuous voice-based cursor control. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 347–356, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. doi: http://doi.acm.org/10.1145/1518701.1518757.

O. Hilliges and D. S. Kirk. Getting sidetracked: display design and occasioning photo-talk with the photohelix. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, page 1733–1736, New York, NY, USA, 2009. ACM. ISBN

978-1-60558-246-7. doi: 10.1145/1518701.1518967. URL http://doi.acm.org/10.1145/1518701.1518967.

T. Hollingsed and D. G. Novick. Usability Inspection Methods After 15 Years of Research and Practice. In *SIGDOC '07: Proc. 25th ACM International Conf. Design of Communication*, pages 249–255, 2007.

S. Hoppe, M. Löchtefeld, and F. Daiber. Eype – using eye-traces for eye-typing. In *Workshop on Grand Challenges in Text Entry (CHI 2013)*, 2013. URL http://hci.uni-saarland.de/?cat=4.

A. Ibrahim and P. Johansson. Multimodal dialogue systems for interactive TV applications. In *in Proceedings of 4th IEEE International Conference on Multimodal Interfaces*, pages 117–222, 2002.

M. Ingmarsson, D. Dinka, and S. Zhai. TNT: A numeric keypad based text input method. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 639–646, New York, NY, USA, 2004. ACM. ISBN 1-58113-702-8. doi: 10.1145/985692.985773. URL http://doi.acm.org/10.1145/985692.985773.

International Telecommunication Union. Recommendation ITU-R m.1677-1 – International Morse code. Technical report, Oct. 2009.

P. Isokoski and R. Raisamo. Device independent text input: a rationale and an example. In *Conference on Advanced Visual Interfaces*, AVI '00, pages 76–83. ACM, 2000. ISBN 1-58113-252-2. doi: 10.1145/345513.345262. URL http://doi.acm.org/10.1145/345513.345262.

H. Istance, S. Vickers, and A. Hyrskykari. The validity of using non-representative users in gaze communication research. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, page 233–236, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1221-9. doi: 10.1145/2168556.2168603. URL http://doi.acm.org/10.1145/2168556.2168603.

J. M. Cedarbaum, N. Stambler, E. Malta, C. Fuller, D. Hilt, B. Thurmond, and A. Nakanishi. The ALSFRS-r: a revised ALS functional rating scale that incorporates assessments of respiratory function. BDNF ALS study group (phase III). *Journal of the Neurological Sciences*, 169(1-2):13–21, Oct. 1999. ISSN 0022-510X.

D. Jackson, T. Bartindale, and P. Olivier. FiberBoard: compact multi-touch display using channeled light. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, page 25–28, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-733-2. doi: 10.1145/1731903.1731908. URL http://doi.acm.org/10.1145/1731903.1731908.

C. L. James and K. M. Reischel. Text input for mobile devices: comparing model prediction to actual performance. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 365–371, New York, NY, USA, 2001. ACM. ISBN 1-58113-327-8. doi: http://doi.acm.org/10.1145/365024.365300.

M. Johnston, L. F. D'Haro, M. Levine, and B. Renger. A multimodal interface for access to content in the home. In *ACL*, pages 376–383, 2007.

M. A. Just and P. A. Carpenter. *The psychology of reading and language comprehension*. Allyn and Bacon, Boston, MA, 1987. ISBN 0205087604 (hardcover).

D. Kirk, S. Izadi, O. Hilliges, R. Banks, S. Taylor, and A. Sellen. At home with surface computing? In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, page 159–168, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2207699. URL http://doi.acm.org/10.1145/2207676.2207699.

D. Klinkhammer, M. Nitsche, M. Specht, and H. Reiterer. Adaptive personal territories for co-located tabletop interaction in a museum setting. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '11, page 107–110, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0871-7. doi: 10.1145/2076354.2076375. URL http://doi.acm.org/10.1145/2076354.2076375.

S. Kovach. What is a smart TV?, Dec. 2010. URL http://www.businessinsider.com/what-is-a-smart-tv-2010-12.

P. O. Kristensson and K. Vertanen. The potential of dwell-free eye-typing for fast assistive gaze communication. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, page 241–244, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1221-9. doi: 10.1145/2168556.2168605. URL http://doi.acm.org/10.1145/2168556.2168605.

P.-O. Kristensson and S. Zhai. SHARK2: A large vocabulary shorthand writing system for pen-based computers. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, UIST '04, page 43–52, New York, NY, USA, 2004. ACM. ISBN 1-58113-957-8. doi: 10.1145/1029632.1029640. URL http://doi.acm.org/10.1145/1029632.1029640.

P. O. Kristensson, O. Arnell, A. Björk, N. Dahlbäck, J. Pennerup, E. Prytz, J. Wikman, and N. Åström. InfoTouch: an explorative multi-touch visualization interface for tagged photo collections. In *Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges*, NordiCHI '08, page 491–494, New York, NY, USA, 2008. ACM. ISBN

978-1-59593-704-9. doi: 10.1145/1463160.1463227. URL http://doi.acm.org/10.1145/1463160.1463227.

M. Kumar. *Gaze-Enhanced User Interface Design*. PhD thesis, Stanford University, 2007. URL http://hci.stanford.edu/research/GUIDe/publications.html.

B. Leung, M. Yates, P. Duez, and T. Chau. Text entry via character stroke disambiguation for an adolescent with severe motor impairment and cortical visual impairment. *Assistive Technology: The Official Journal of RESNA*, 22(4):223–235, 2010. ISSN 1040-0435. doi: 10.1080/10400435.2010.518580.

A. Leveille, J. Kiernan, J. A. Goodwin, and J. Antel. Eye movements in Amyotrophic Lateral Sclerosis. *Archives of Neurology*, 39(11):684–686, Nov. 1982. ISSN 0003-9942. URL http://www.ncbi.nlm.nih.gov/pubmed/7125995.

F. C. Y. Li, R. T. Guy, K. Yatani, and K. N. Truong. The 1line keyboard: a QWERTY layout in a single line. In *Symposium on User Interface Software and Technology*, UIST '11, pages 461–470. ACM, 2011. ISBN 978-1-4503-0716-1. doi: 10.1145/2047196.2047257. URL http://doi.acm.org/10.1145/2047196.2047257.

V. Lobato, G. López, and V. M. Peláez. MHP interactive applications: Combining visual and speech user interaction modes. In *EuroITV '09*, pages 10–13, 2009.

A. Lucero, J. Holopainen, and T. Jokela. Pass-them-around: collaborative use of mobile phones for photo sharing. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, page 1787–1796, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1979201. URL http://doi.acm.org/10.1145/1978942.1979201.

F. Lui, S. Fonda, L. Merlini, and R. Corazza. Saccadic eye movements are impaired in Duchenne Muscular Dystrophy. *Documenta Ophthalmologica. Advances in Ophthalmology*, 103(3):219–228, Nov. 2001. ISSN 0012-4486. URL http://www.ncbi.nlm.nih.gov/pubmed/11824659.

K. Luyten, K. Thys, S. Huypens, and K. Coninx. Telebuddies: Social stitching with interactive television. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pages 1049–1054, New York, NY, USA, 2006. ACM. ISBN 1-59593-298-4. doi: 10.1145/1125451.1125651. URL http://doi.acm.org/10.1145/1125451.1125651.

I. S. Mackenzie and T. Felzer. SAK: scanning ambiguous keyboard for efficient one-key text entry. *ACM Trans. Comput.-Hum. Interact.*, 17(3):11:1–11:39, July 2010. ISSN 1073-0516. doi: 10.1145/1806923.1806925. URL http://doi.acm.org/10.1145/1806923.1806925.

I. S. MacKenzie and R. W. Soukoreff. A character-level error analysis technique for evaluating text entry methods. In *Proceedings of the second Nordic conference on Human-computer interaction*, NordiCHI '02, page 243–246, New York, NY, USA, 2002. ACM. ISBN 1-58113-616-1. doi: 10.1145/572020.572056. URL http://doi.acm.org/10.1145/572020.572056.

I. S. MacKenzie and R. W. Soukoreff. Phrase sets for evaluating text entry techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, page 754–755, New York, NY, USA, 2003. ACM. ISBN 1-58113-637-4. doi: 10.1145/765891.765971. URL http://doi.acm.org/10.1145/765891.765971.

I. S. MacKenzie and X. Zhang. Eye typing using word and letter prediction and a fixation algorithm. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, ETRA '08, page 55–58, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-982-1. doi: 10.1145/1344471.1344484. URL http://doi.acm.org/10.1145/1344471.1344484.

I. S. MacKenzie, R. W. Soukoreff, and J. Helga. 1 thumb, 4 buttons, 20 words per minute: design and evaluation of H4-writer. In *Symposium on User Interface Software and Technology*, UIST '11, pages 471–480. ACM, 2011. ISBN 978-1-4503-0716-1. doi: 10.1145/2047196.2047258. URL http://doi.acm.org/10.1145/2047196.2047258.

P. Majaranta, U.-K. Ahola, and O. Špakov. Fast gaze typing with an adjustable dwell time. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, page 357–360, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. doi: 10.1145/1518701.1518758. URL http://doi.acm.org/10.1145/1518701.1518758.

N. Marquardt, T. Ballendat, S. Boring, S. Greenberg, and K. Hinckley. Gradual engagement: Facilitating information exchange between digital devices as a function of proximity. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '12, pages 31–40, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1209-7. doi: 10.1145/2396636.2396642. URL http://doi.acm.org/10.1145/2396636.2396642.

M. McGill, J. Williamson, and S. A. Brewster. How to lose friends & alienate people: Sharing control of a single-user TV system. In *Proceedings of the 2014 ACM International Conference on Interactive Experiences for TV and Online Video*, TVX '14, pages 147–154, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2838-8. doi: 10.1145/2602299.2602318. URL http://doi.acm.org/10.1145/2602299.2602318.

Medical News Today. What is motor neuron disease? What is Amyotrophic Lateral Sclerosis (ALS), or Lou Gehrig's Disease?, 2009. URL

http://www.medicalnewstoday.com/articles/164342.php. Accessed 17 September 2013.

C. H. Morimoto and A. Amir. Context switching for fast key selection in text entry applications. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ETRA '10, page 271–274, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-994-7. doi: 10.1145/1743666.1743730. URL http://doi.acm.org/10.1145/1743666.1743730.

S. Morris and A. Smith-Chaigneau. *Interactive TV Standards*. Focal Press, 2005.

M. Möllers and J. Borchers. TaPS widgets: interacting with tangible private spaces. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '11, page 75–78, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0871-7. doi: 10.1145/2076354.2076369. URL http://doi.acm.org/10.1145/2076354.2076369.

T. Nakajima. How to reuse exisiting interactive applications in ubiquitous computing environments? In *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*, pages 1127–1133, New York, NY, USA, 2006. ACM. ISBN 1-59593-108-2. doi: http://doi.acm.org/10.1145/1141277.1141546.

Y. Nakatoh, H. Kuwano, T. Kanamori, and M. Hoshimi. Speech Recognition Interface System for Digital TV Control — Special Issue Applied Systems. *Acoustical science and technology*, 28(3):165–171, 2007.

National Human Genome Research Institute. Learning about duchenne muscular dystrophy, 2013. URL http://www.genome.gov/19518854.

National Institute of Neurological Disorders and Stroke. Motor neuron diseases fact sheet, 2012. URL http://www.ninds.nih.gov/disorders/motor_neuron_diseases/detail_motor_neuron_diseases.htm. Accessed 17 September 2013.

M. Orsini, A. M. d. S. Catharino, F. M. C. Catharino, M. P. Mello, M. R. G. d. Freitas, M. A. A. Leite, and O. J. M. Nascimento. Man-in-the-barrel syndrome, a symmetrical proximal brachial amyotrophic diplegia related to motor neuron diseases: A survey of nine cases. *Revista da Associação Médica Brasileira*, 55(6):712–715, Jan. 2009. ISSN 0104-4230. doi: 10.1590/S0104-42302009000600016. URL http://www.scielo.br/scielo.php?pid=S0104-42302009000600016&script=sci_abstract&tlng=pt.

Patient.co.uk. Duchenne muscular dystrophy, 2013. URL http://www.patient.co.uk/health/duchenne-muscular-dystrophy-leaflet.

D. Pedrosa and M. d. G. Pimentel. Text entry using a foot for severely motor-impaired individuals. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, SAC '14, New York, NY, USA, 2014. ACM.

D. Pedrosa, J. A. C. M. Jr, E. Melo, and M. d. G. C. Pimentel. Componente de interação multimodal no ginga. In *Proceedings of 16th Brazilian Symposium on Multimedia and the Web*, volume 2, pages 197–202, Belo Horizonte, 2010a.

D. Pedrosa, J. A. C. Martins, Jr., E. L. Melo, and C. A. C. Teixeira. A multimodal interaction component for digital television. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, SAC '11, pages 1253–1258, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0113-8. doi: http://doi.acm.org/10.1145/1982185.1982459. URL `http://doi.acm.org/10.1145/1982185.1982459`.

D. Pedrosa, R. L. Guimarães, M. da Graça Pimentel, D. C. A. Bulterman, and P. Cesar. Interactive coffee table for exploration of personal photos and videos. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, pages 967–974, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1656-9. doi: 10.1145/2480362.2480548. URL `http://doi.acm.org/10.1145/2480362.2480548`.

D. Pedrosa, M. d. G. Campos Pimentel, A. Wright, and K. Truong. Filteryedping: Design challenges and user performance of dwell-free eye-typing. *(submitted to) ACM Trans. Access. Comput.*, X(X):X:1–X:37, Oct. 2014. ISSN 1936-7228. doi: XXX. URL `XXX`.

D. Pedrosa, M. d. G. Campos Pimentel, and K. Truong. Filteryedping: A dwell-free eye-typing technique. In *(submitted to) Proceedings of the Extended Abstracts of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI EA '15, pages X:1–X:4, New York, NY, USA, 2015. ACM.

D. d. C. Pedrosa, D. A. Vega-Oliveros, R. P. d. M. Fortes, and M. d. G. C. Pimentel. Text input in digital television: a component prototype. In *EuroITV '10: Proceedings of the 8th European conference on Changing Television Environments*, 2010b. Submetido.

C. Peng. *Digital Television Applications*. PhD thesis, Helsinki University of Technology, November 2002.

J. Perrinet, X. G. Pañeda, S. Cabrero, D. Melendi, R. García, and V. García. Evaluation of virtual keyboards for interactive digital television applications. *International Journal of Human-Computer Interaction*, 27(8):703–728, Mar. 2011. ISSN 1044-7318. doi: 10.1080/10447318.2011.555305. URL `http://www.tandfonline.com/doi/abs/10.1080/10447318.2011.555305`.

M. d. G. C. Pimentel, R. G. Cattelan, E. L. Melo, A. F. Prado, and C. A. C. Teixeira. End-user live editing of iTV programmes. *Int. J. Adv. Media Commun.*, 4(1):78–103, 2010. ISSN 1462-4613. doi: http://dx.doi.org/10.1504/IJAMC.2010.030007.

T. Regan and I. Todd. Media center buddies: Instant messaging around a media center. In *Proceedings of the Third Nordic Conference on Human-computer Interaction*, NordiCHI '04, pages 141–144, New York, NY, USA, 2004. ACM. ISBN 1-58113-857-1. doi: 10.1145/1028014.1028036. URL http://doi.acm.org/10.1145/1028014.1028036.

G. Ren and E. O'Neill. Freehand gestural text entry for interactive TV. In *Proceedings of the 11th European Conference on Interactive TV and Video*, EuroITV '13, pages 121–130, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1951-5. doi: 10.1145/2465958.2465966. URL http://doi.acm.org/10.1145/2465958.2465966.

M. Rice and N. Alm. Designing new interfaces for digital interactive television usable by older adults. *Comput. Entertain.*, 6(1):1–20, 2008. ISSN 1544-3574. doi: http://doi.acm.org/10.1145/1350843.1350849.

A. Roibás, R. Sala, S. Ahmad, and M. Radman. Beyond the remote control: Going the extra mile to enhance iTV access via mobile devices & humanizing navigation experience for those with special needs. In *EuroITV '2005*, pages 133–141, 2005. ISBN 978-3-540-69477-9.

D. Rough, K. Vertanen, and P. O. Kristensson. An evaluation of Dasher with a high-performance language model as a gaze communication method. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, AVI '14, page 169–176, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2775-6. doi: 10.1145/2598153.2598157. URL http://doi.acm.org/10.1145/2598153.2598157.

K.-J. Räihä and S. Ovaska. An exploratory study of eye typing fundamentals: dwell time, text entry rate, errors, and workload. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 3001–3010, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2208711. URL http://doi.acm.org/10.1145/2207676.2208711.

S. Sarcar, P. Panwar, and T. Chakraborty. EyeK: An efficient dwell-free eye gaze-based text entry system. In *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction*, APCHI '13, page 215–220, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2253-9. doi: 10.1145/2525194.2525288. URL http://doi.acm.org/10.1145/2525194.2525288.

J. Scott, D. Dearman, K. Yatani, and K. N. Truong. Sensing foot gestures from the pocket. In *Symposium on User Interface Software and Technology*, UIST '10, pages 199–208. ACM,

2010. ISBN 978-1-4503-0271-5. doi: 10.1145/1866029.1866063. URL http://doi.acm.org/10.1145/1866029.1866063.

S. D. Scott, M. Sheelagh, T. Carpendale, and K. M. Inkpen. Territoriality in collaborative tabletop workspaces. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, CSCW '04, page 294–303, New York, NY, USA, 2004. ACM. ISBN 1-58113-810-5. doi: 10.1145/1031607.1031655. URL http://doi.acm.org/10.1145/1031607.1031655.

T. Seifried, M. Haller, S. D. Scott, F. Perteneder, C. Rendl, D. Sakamoto, and M. Inami. CRISTAL: a collaborative home media and device controller based on a multi-touch display. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, page 33–40, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-733-2. doi: 10.1145/1731903.1731911. URL http://doi.acm.org/10.1145/1731903.1731911.

C. Shen, N. Lesh, and F. Vernier. Personal digital historian: story sharing around the table. *interactions*, 10(2):15–22, Mar. 2003. ISSN 1072-5520. doi: 10.1145/637848.637856. URL http://doi.acm.org/10.1145/637848.637856.

M. Silfverberg, I. S. MacKenzie, and P. Korhonen. Predicting text entry speed on mobile phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, pages 9–16, New York, NY, USA, 2000. ACM. ISBN 1-58113-216-6. doi: 10.1145/332040.332044. URL http://doi.acm.org/10.1145/332040.332044.

L. D. N. Silva, C. E. C. F. Batista, L. E. C. Leite, and G. L. Souza Filho. Suporte para desenvolvimento de aplicações multiusuário e multidispositivo para TV Digital com Ginga. *T&C Amazônia*, 12:75–84, 2007.

L. D. N. Silva, T. A. Tavares, and G. L. Souza. Desenvolvimento de programas de TVDI explorando as funções inovadoras do Ginga-J. In *WebMedia '2008*, pages 26–29, 2008. ISBN 85-7669-100-0.

R. Simpson. Modeling one-switch row-column scanning with errors and error correction methods. *The Open Rehabilitation Journal*, 4(1):1–12, 2011. ISSN 18749437. doi: 10.2174/1874943701104010001. URL http://dx.doi.org/10.2174/1874943701104010001.

L. F. G. Soares, R. M. Costa, M. F. Moreno, and M. F. Moreno. Multiple exhibition devices in DTV systems. In *ACM MM'2009*, pages 281–290, 2009. ISBN 978-1-60558-608-3. doi: http://doi.acm.org/10.1145/1631272.1631312.

A. Soro, S. A. Iacolina, R. Scateni, and S. Uras. Evaluation of user gestures in multi-touch interaction: a case study in pair-programming. In *Proceedings of the 13th international conference on multimodal interfaces*, ICMI '11, page 161–168, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0641-6. doi: 10.1145/2070481.2070508. URL http://doi.acm.org/10.1145/2070481.2070508.

R. W. Soukoreff and I. S. MacKenzie. Metrics for text entry research: an evaluation of MSD and KSPC, and a new unified error metric. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, page 113–120, New York, NY, USA, 2003. ACM. ISBN 1-58113-630-7. doi: 10.1145/642611.642632. URL http://doi.acm.org/10.1145/642611.642632.

A. J. Sporka, O. Polacek, and P. Slavik. Comparison of two text entry methods on interactive TV. In *Proceedings of the 10th European Conference on Interactive TV and Video*, EuroiTV '12, pages 49–52, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1107-6. doi: 10.1145/2325616.2325627. URL http://doi.acm.org/10.1145/2325616.2325627.

M. V. Springett and R. N. Griffiths. Innovation for inclusive design: an approach to exploring the idtv design space. In *UXTV '08: Proceeding of the 1st international conference on Designing interactive user experiences for TV and video*, pages 49–58, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-100-2. doi: http://doi.acm.org/10.1145/1453805.1453817.

H. Stelmaszewska, B. Fields, and A. Blandford. The roles of time, place, value and relationships in collocated photo sharing with camera phones. In *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction - Volume 1*, BCS-HCI '08, pages 141–150, Swinton, UK, UK, 2008. British Computer Society. ISBN 978-1-906124-04-5. URL http://dl.acm.org/citation.cfm?id=1531514.1531534.

M. Sugimoto, K. Hosoi, and H. Hashizume. Caretta: a system for supporting face-to-face collaboration by integrating personal and shared spaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, page 41–48, New York, NY, USA, 2004. ACM. ISBN 1-58113-702-8. doi: 10.1145/985692.985698. URL http://doi.acm.org/10.1145/985692.985698.

L. Swan and A. S. Taylor. Photo displays in the home. In *Proceedings of the 7th ACM conference on Designing interactive systems*, DIS '08, page 261–270, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-002-9. doi: 10.1145/1394445.1394473. URL http://doi.acm.org/10.1145/1394445.1394473.

K. Takashima, N. Aida, H. Yokoyama, and Y. Kitamura. TransformTable: A self-actuated shape-changing digital table. In *Proceedings of the 2013 ACM International Conference on*

Interactive Tabletops and Surfaces, ITS '13, pages 179–188, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2271-3. doi: 10.1145/2512349.2512818. URL http://doi.acm.org/10.1145/2512349.2512818.

W. Tangsuksant, C. Aekmunkhongpaisal, P. Cambua, T. Charoenpong, and T. Chanwimalueang. Directional eye movement detection system for virtual keyboard controller. In *Biomedical Engineering International Conference (BMEiCON), 2012*, pages 1–5, 2012. doi: 10.1109/BMEiCon.2012.6465432. URL http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6465432.

C. A. C. Teixeira, E. L. Melo, R. G. Cattelan, and M. d. G. C. Pimentel. User-media interaction with interactive tv. In *ACM SAC '2009*, pages 1829–1833, 2009. ISBN 978-1-60558-166-8. doi: http://doi.acm.org/10.1145/1529282.1529690.

C. A. C. Teixeira, E. L. Melo, G. B. Freitas, C. A. S. Santos, and M. d. G. C. Pimentel. Discrimination of media moments and media intervals: sticker-based watch-and-comment annotation. *Multimedia Tools and Applications*, 61(3):675–696, Dec. 2012. ISSN 1380-7501, 1573-7721. doi: 10.1007/s11042-011-0846-6. URL http://link.springer.com/article/10.1007/s11042-011-0846-6.

V. Tkotz. Frequência de ocorrência de letras no português, 2005. URL http://www.numaboa.com.br/criptografia/criptoanalise/310-Frequencia-no-Portugues. Accessed 17/Sep/2013.

United Nations. Convention on the rights of persons with disabilities, 2006. URL http://www.un.org/disabilities/convention/conventionfull.shtml. Convention on the rights of persons with disabilities.

M. H. Urbina and A. Huckauf. Dwell time free eye typing approaches. In *The 3rd Conference on Communication by Gaze Interaction – COGAIN 2007: Gaze-based Creativity and Interacting with Games and On-line Communities (COGAIN)*, pages 65–70, Leicester, UK, 2007. URL http://wiki.cogain.org/images/e/e5/COGAIN2007Proceedings.pdf.

U.S. National Library of Medicine. Duchenne muscular dystrophy, 2012. URL http://www.nlm.nih.gov/medlineplus/ency/article/000705.htm.

D. A. Vega-Oliveros, D. d. C. Pedrosa, M. d. G. C. Pimentel, and R. P. de Mattos Fortes. An approach based on multiple text input modes for interactive digital TV applications. In *Proceedings of the 28th ACM International Conference on Design of Communication*, SIGDOC '10, pages 191–198, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0403-0. doi: http://doi.acm.org/10.1145/1878450.1878483. URL http://doi.acm.org/10.1145/1878450.1878483.

K. Vertanen and P. O. Kristensson. A versatile dataset for text entry evaluations based on genuine mobile emails. In *Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pages 295–298, 2011. ISBN 978-1-4503-0541-9. doi: 10.1145/2037373.2037418. URL http://doi.acm.org/10.1145/2037373.2037418.

D. J. Ward, A. F. Blackwell, and D. J. C. MacKay. Dasher – a data entry interface using continuous gestures and language models. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, UIST '00, page 129–137, New York, NY, USA, 2000. ACM. ISBN 1-58113-212-3. doi: 10.1145/354401.354427. URL http://doi.acm.org/10.1145/354401.354427.

D. Wigdor, C. Shen, C. Forlines, and R. Balakrishnan. Table-centric interactive spaces for real-time collaboration. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '06, page 103–107, New York, NY, USA, 2006. ACM. ISBN 1-59593-353-0. doi: http://doi.acm.org/10.1145/1133265.1133286. URL http://doi.acm.org/10.1145/1133265.1133286.

A. D. Wilson and M. Agrawala. Text entry using a dual joystick game controller. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 475–478, New York, NY, USA, 2006. ACM. ISBN 1-59593-372-7. doi: http://doi.acm.org/10.1145/1124772.1124844.

K. Wittenburg, T. Lanning, D. Schwenke, H. Shubin, and A. Vetro. The prospects for unrestricted speech input for TV content search. In *AVI '06: Proc. Conf. Advanced Visual Interfaces*, pages 352–359, New York, NY, USA, 2006. ACM. ISBN 1-59593-353-0. doi: http://doi.acm.org/10.1145/1133265.1133338.

J. Wobbrock and B. Myers. Trackball text entry for people with motor impairments. In *Conference on Human Factors in Computing Systems*, CHI '06, pages 479–488, 2006. ISBN 1-59593-372-7. doi: 10.1145/1124772.1124845. URL http://doi.acm.org/10.1145/1124772.1124845.

J. O. Wobbrock and Brad A. Myers. Analyzing the input stream for character- level errors in unconstrained text entry evaluations. *ACM Trans. Comput.-Hum. Interact.*, 13(4):458–489, 2006. ISSN 1073-0516. doi: 10.1145/1188816.1188819. URL http://doi.acm.org/10.1145/1188816.1188819.

J. O. Wobbrock, B. A. Myers, and J. A. Kembel. EdgeWrite: a stylus-based text entry method designed for high accuracy and stability of motion. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, UIST '03, pages 61–70, New York, NY, USA, 2003. ACM. ISBN 1-58113-636-6. doi: 10.1145/964696.964703. URL http://doi.acm.org/10.1145/964696.964703.

J. O. Wobbrock, J. Rubinstein, M. W. Sawyer, and A. T. Duchowski. Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, ETRA '08, page 11–18, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-982-1. doi: 10.1145/1344471.1344475. URL http://doi.acm.org/10.1145/1344471.1344475.

C.-M. Wu, C. Y. Chuang, M.-C. Hsieh, and S.-H. Chang. An eye input device for persons with the motor neuron diseases. *Biomedical Engineering: Applications, Basis and Communications*, 25(01):1350006, Feb. 2013. ISSN 1016-2372, 1793-7132. doi: 10.4015/S1016237213500063. URL http://www.worldscientific.com/doi/abs/10.4015/S1016237213500063.

Y. Yoshimoto, T. H. Dang, A. Kimura, F. Shibata, and H. Tamura. Interaction design of 2D/3D map navigation on wall and tabletop displays. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '11, page 254–255, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0871-7. doi: 10.1145/2076354.2076402. URL http://doi.acm.org/10.1145/2076354.2076402.

X. A. Zhao, E. D. Guestrin, D. Sayenko, T. Simpson, M. Gauthier, and M. R. Popovic. Typing with eye-gaze and tooth-clicks. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, page 341–344, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1221-9. doi: 10.1145/2168556.2168632. URL http://doi.acm.org/10.1145/2168556.2168632.