On the critical role of the sensorimotor loop on the design of interaction techniques and interactive devices

Thomas Pietrzak

To cite this version:

Thomas Pietrzak. On the critical role of the sensorimotor loop on the design of interaction techniques and interactive devices. Computer Science [cs]. Université de Lille (2018-..), 2022. tel-03758501v2

HAL Id: tel-03758501
https://hal.inria.fr/tel-03758501v2

Submitted on 30 Aug 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

Faculté de Sciences et Technologies
Département Informatique

École Doctorale MADIS

HABILITATION À DIRIGER DES RECHERCHES

présentée par

THOMAS PIETRZAK

Discipline : Informatique

Spécialité : Interaction Humain-Machine

On the critical role of the sensorimotor loop on the design of interaction techniques and interactive devices

8 Juillet 2022

Rapporteuses et rapporteurs

Karon MacLean	Professeure	University of British Columbia
Laurence Nigay	Professeure	Université Grenoble Alpes
Philippe Palanque	Professeur	Université Toulouse III - Paul Sabatier

Examinatrices et examinateurs

Michel Beaudouin-Lafon	Professeur	Université Paris-Saclay
Dominique Bechmann	Professeure	Université de Strasbourg
Christian Duriez	Directeur de Recherche	Inria

Garant

Stéphane Huot	Directeur de Recherche	Inria

Habilitation à diriger des recherches préparée au sein de l'équipe projet LOKI
commune à l'Université de Lille et Inria
au sein du centre Inria de l'Université de Lille et du laboratoire CRIStAL — UMR CNRS 9189

Abstract

People interact with their environment thanks to their perceptual and motor skills. This is the way they both use objects around them and perceive the world around them. Interactive systems are examples of such objects. Therefore to design such objects, we must understand how people perceive them and manipulate them. For example, haptics is both related to the human sense of touch and what I call the motor ability. I address a number of research questions related to the design and implementation of haptic, gestural, and touch interfaces and present examples of contributions on these topics. More interestingly, perception, cognition, and action are not separated processes, but an integrated combination of them called the sensorimotor loop. Interactive systems follow the same overall scheme, with differences that make the complementarity of humans and machines. The interaction phenomenon is a set of connections between human sensorimotor loops, and interactive systems execution loops. It connects inputs with outputs, users and systems, and the physical world with cognition and computing in what I call the Human-System loop. This model provides a complete overview of the interaction phenomenon. It helps to identify the limiting factors of interaction that we can address to improve the design of interaction techniques and interactive devices.

Les humains interagissent avec leur environnement grâce à leurs capacités perceptives et motrices. C’est ainsi qu’ils utilisent les objets qui les entourent et perçoivent le monde autour d’eux. Les systèmes interactifs sont des exemples de tels objets. Par conséquent, pour concevoir de tels objets, nous devons comprendre comment les gens les perçoivent et les manipulent. Par exemple, l’haptique est à la fois liée au sens du toucher et à ce que j’appelle la capacité motrice. J’aborde un certain nombre de questions de recherche liées à la conception et à la mise en œuvre d’interfaces haptiques, gestuelles et tactiles et je présente des exemples de contributions sur ces sujets. Plus intéressant encore, la perception, la cognition et l’action ne sont pas des processus séparés, mais une combinaison intégrée d’entre eux appelée la boucle sensorimotrice. Les systèmes interactifs suivent le même schéma global, avec des différences qui forme la complémentarité des humains et des machines. Le phénomène d’interaction est un ensemble de connexions entre les boucles sensorimotrices humaines et les boucles d’exécution des systèmes interactifs. Il relie les entrées aux sorties, les utilisateurs aux systèmes, et le monde physique à la cognition et au calcul dans ce que j’appelle la boucle Humain-Système. Ce modèle fournit un aperçu complet du phénomène d’interaction. Il permet d’identifier les facteurs limitatifs de l’interaction que nous pouvons aborder pour améliorer la conception des techniques d’interaction et des dispositifs interactifs.

Contents

List of Figures

Acknowledgements

In this document, I discuss the results of a successful decade of work. However, my main achievement in this period is my family. I met my wife Milena, our son Jimmy was born, and we got married. We are all here for the other when one needs help or attention, whether to get through difficult times or to celebrate the highlights. I love you both more than anything else in this universe. The true achievement of my career is certainly my ability to keep a good work-life balance.

I am also grateful to my parents Paul and Chantal, as well as my brother Julien. They always supported me and there was no chance I was there today without them. I also thank the rest of my family and my step-family for providing a kind and warm environment. It is always a pleasure to have a good time together.

I also have the chance to have the most faithful and cheerful friends one hopes to have. They are always ready to hang around, help, or do anything else one of us needs. The frontier between friends and colleagues is sometimes difficult to draw. I am so grateful to the Telecom team that welcomed me for my postdoc in Paris. You helped me to take off my research, and we spent many good moments both in and outside the lab. You all are one of the reasons why I achieved this step today, and I am proud to count you as a friend as well.

Throughout my postdocs in Paris, Toronto, and my position at the Univerité de Lille I had the honor and pleasure to work with many talented people. These stimulating environments pulled me to the heights I reached today. I was delighted to share this part of my journey with you. Notably, I owe a life debt to all my team leaders: Éric, Ravin, Laurent, Nicolas, and Stéphane. Not only you all provided me with an outstanding environment that helped me to perform research at the highest levels and helped me to shape my vision of my research. But thanks to you the day I will lead a team myself, I will have five different leadership examples to get inspiration from. In particular, I truly thank Stéphane for being my HDR sponsor. He was for me the best person to guide me through this exercise, and in particular, push me to go further in my thinking.

All these welcoming and successful research environments are also combinations of very talented researchers. The ideas and visions presented in this document are the consequences of discussions, teamwork, and inspiration from my direct and indirect collaborators. Specifically, I thank my most frequent co-authors Géry, Nicolas, Sylvain, and Gilles, and most frequent co-advisors Audrey and Marcelo. More generally, I express my deepest gratitude to all my co-authors and more generally all the other team members and members of the HCI research community.

Beyond the research teams, I have been working in stimulating and welcoming environments, whether in the Computer Science department of the Faculté des Sciences et Technologies, the Inria Lille research center, or the CRIStAL laboratory. I thank them all. I would also like to give a special thanks to all the support teams at Inria, and more than anybody else our team assistants Karine, Julie, and Anne. Without talented, organized, and kind people like you, our job would not be possible.

I would like also to thank all the committee members for accepting to review my work. All of you inspired my research since I started research 17 years ago. Therefore, it is an honor for me to be able to present to you what I have been working on during the last decade.

À Milena et Jimmy,

Introduction

> Was there anything more exciting in life than seeking answers?
>
> Isaac Asimov

My research domain is Human-Computer Interaction (*HCI*). I study the interaction phenomenon between people and interactive systems. To produce such phenomenons I design and implement interaction techniques and interactive devices. Most of the work in this document is related to haptics in the broad sense. The word *haptics* comes from the ancient Greek word ἁπτικός or *haptikós* which means to touch, contact, manipulate[1]. Therefore I not only refer to haptics as a way to stimulate the human sense of touch, but also the human ability to perform motor actions. I discuss how my focus was initially on these two notions separately, and how I finally combined these two approaches and now focus on the sensorimotor loop. Therefore, haptics is in the end an occasion to discuss the sensorimotor loop, which is a higher-level notion and the true focus of my work today.

I describe my research since I got my Maître de Conférences position at the University of Lille in September 2011. I started my research at CRIStAL and Inria in the Mint project team. The Mint research themes were tactile and gestural interaction. Then, I joined the Mjolnir research group with other Mint members. We worked on tools to empower people, in the vein of Engelbart and Bush's visions [44, 73]. Finally, this research group became Loki, with a focus on technology and knowledge for interaction. These themes, and more generally the exceptional working environment I have been working in for 11 years had and still have a large impact on my work and my vision of my research field I will describe in the next chapters.

I have been thinking for years about the form this document should take. I decided to write a document that I hope to be useful for anybody who wants to design, implement, or evaluate interactive systems, in particular, but not limited to haptics. My core expertise is in Computer Science, but research in HCI is typically interdisciplinary. HCI experts will probably not be surprised to read elements of experimental psychology, electronics, robotics, or design for example. Other readers must understand that the force of our research domain, is to allow studying research problems in a holistic and systemic way. This is my favorite approach, and the way we practice our research in the Loki team (and previously Mjolnir). We do not focus on a particular technology or methodology for example. On the opposite, we are

---

[1] https://en.wiktionary.org/wiki/haptic

open to any problem, that we must characterize before searching for appropriate solutions. Hence, we rather need a large overview of methodologies, technologies, and knowledge. Thanks to this kind of expertise, we are used to making bridges between research communities, and keeping an overview of the large picture rather than focusing on a tiny part of the problem. For example, in 2019 with Marcelo Wanderley we revived the HAID workshop[2] that brings together researchers and practitioners in HCI, haptics, and music. Therefore, this document rather covers the basics of many notions I consider as important rather than focusing on particular points. Indeed, each of these notions is extensively studied in the literature. The interested reader will find entry points to this literature in this document.

Chapter I – The sense of touch

The Chapter I extends my Ph.D. work during which I used haptics to help visually impaired children at school. I describe a haptics rendering pipeline that details the different steps on both the system and human side, and both the hardware and software parts. This pipeline highlights the possible causes of alteration of the original message between the software and the user's mind. I explain the basics of haptic technologies and the sense of touch, then I discuss the challenges of their design. The first challenge is about the output vocabulary and how we encode information with haptic cues. The second one is about the intertwined relationship between engineering and the evaluation of haptic devices. The third challenge addresses the restoration of missing haptic properties of physical controls in multi-touch and gestural interaction. The fourth challenge is about the haptic properties of tangible controls and their effect on interaction. Then I discuss a number of contributions to these challenges, about Tactons, tactile textures, printed vibrotactile widgets, and actuated computer peripherals.

Chapter II – The motor ability

The Chapter II presents a mirror vision of the work presented in Chapter I. The input pipeline I discuss is almost identical to the haptic rendering pipeline, except that the user is initiating the action. In a similar way, this pipeline describes the possible causes of alteration of the system's interpretation of the user's intentions. I could not find an equivalent term of *senses* for human outputs. For lack of anything better, I defined the term *ability* and in particular the *motor ability*. I describe the way it works, as well as the way input systems work. Then I present challenges for the design of input systems. The first challenge is about the sensing and interpretation of human abilities. The second one addresses the design of input vocabularies. The third challenge is about the relevance of unnatural inputs and the limitation of only replicating the physical world into the digital world, which misses many opportunities to augment what human can do with digital tools. To address these challenges I present several contributions about latency measurement, flexible pens, finger identification, and interaction techniques in Virtual Reality.

---

[2] http://haid2019.lille.inria.fr/

Chapter III – The Sensorimotor loop

While Chapter II and Chapter I described the vision of HCI I had at the beginning of my research career. This vision evolved as the sensorimotor loop kept having greater importance in my work. The point of Chapter III is to explain that the first two chapters should have never been separated. This chapter starts with studies that failed at improving interaction substantially because they followed the vision of the two first chapters. Then I adddress these issues with contributions that leverage the sensorimotor loop for gestural interaction, and the sense of embodiment in immersive virtual reality. I conclude with a discussion about the connections between computing and the sensorimotor loop through models of human behavior, system architectures, and interaction paradigms. I propose a new model of a Human-System loop that takes into account both the users and the system, the inputs and the outputs, and the physical world with the software world. This model helps me keep a holistic view of interaction, and identify the bottlenecks that limit interaction.

The sense of touch

> Education is not something you can finish.
>
> Isaac Asimov

We discuss haptics as the sense of touch, and the implication for the design and implementation of haptic systems. To do so, we present the haptic pipeline that illustrates the hardware and software parts of both interactive systems and users. First, it shows the diversity of disciplines involved in the design and implementation of haptic systems. Second, it reveals pitfalls that potentially alter the message transmitted to users through touch at every stage of the pipeline. We present the main general research questions that guided my research: the output vocabulary, the engineering and evaluation of haptic devices, the haptic properties of physical objects, and the use of tangible objects for haptic interaction. We illustrate these research questions with several research projects: vibrotactile Tactons for activity monitoring, tactile textures with programmable friction, printed vibrotactile widgets, and actuated computer peripherals.

Haptics is generally seen as an output modality, and most haptic systems are designed for providing *haptic feedback*. Despite the usual user-centered approach in HCI, this is the typical convention in computer science and robotics, with a system-centered point of view. This is not the case for all scientific disciplines. For example, I had misunderstandings with a postdoc researcher with a background in cognitive sciences when he joined the Mint group a few years ago. What he referred to as input and output was the opposite of the convention I use. Initially, it created confusion between us, but soon after it helped us to understand each other's points of view. It also helped me to characterize the symmetries and asymmetries between humans and systems that I will discuss in Chapter III. This chapter will cover haptics as a way to stimulate the sense of touch.

There is a large diversity of haptic sensations and perceptions. When I use vocabulary related to haptic, I use Oakley *et al.*'s' definitions [189]. As far as I know, this was the first time these terms were clearly defined in the Human-Computer Interaction community. Following these definitions, *haptics* is a general term that refers to anything related to the sense of touch. *Kinesthetic* refers to the feeling of motion, resulting from sensations produced in muscles, tendons, and joints. *Force Feedback* is about the "mechanical production of information sensed by the human kinesthetic system". *Tactile* relates to pressure sensations sensed by the skin.

The design of haptic systems requires joint efforts between specialists in several scientific domains as depicted in Figure I.1. This pipeline has two steps on the system side and two steps on the human side. Both the system and humans have a step in the physical world and one outside the physical world. The objective of this pipeline is to transmit information to users through their sense of touch. This is indeed a simplified pipeline, which nevertheless shows that several scientific disciplines are interested in this topic. While each step of the pipeline is essentially studied by one or two scientific fields, the role of Human-Computer Interaction is to connect them in a meaningful and useful way for people. It is an illustration of our holistic approach, that enables us to study the interaction phenomenon that not only takes into account the system, but also the users, their tasks, and the environment.

Haptic systems

The objective of the pipeline is to render virtual objects or pieces of information into haptic sensations. The pipeline starts with the software part of the system, which uses the data associated with this information to compute a command or a signal. This is the system side, out of the physical world. Here we consider, relatively speaking, that the software is the computer's mind. These commands activate various kinds of actuators to produce a mechanical effect.

Since there is a dichotomy between kinesthetic and tactile sensation, there is also a dichotomy between force-feedback and tactile systems. Force-feedback systems typically use one of two major types of controls. The usual way to control a force-feedback system is to measure motion and compute a force, this is called *impedance control* [224]. This is the most common technique, mostly because there are many easy and cheap ways to measure motion. A function that computes a force

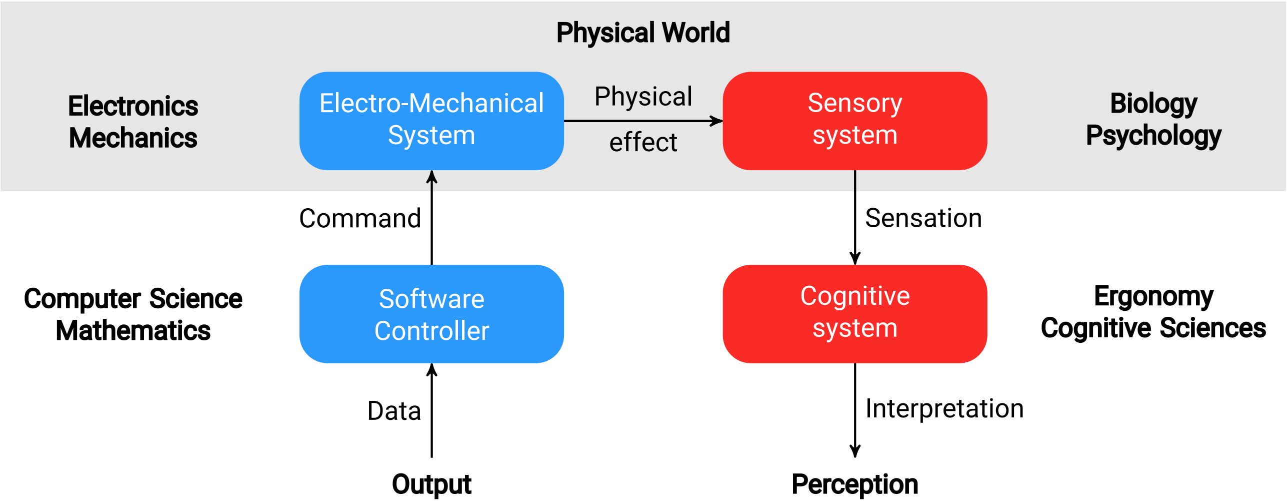


Figure I.1 – Haptic rendering pipeline with the system side and user side from the specified outputs to their perception. Both have a hardware and a software aspect.

depending on movement is called a force model. Other systems sense forces and compute an output motion, this is called *admittance control* [18, 253]. This is less used because forces are harder to measure, especially if precision is required. The advantage of such systems is their much higher stiffness than inductance systems. It is also worth mentioning Electrical Muscular Stimulation (EMS) which directly stimulates muscles with electrical signals [247].

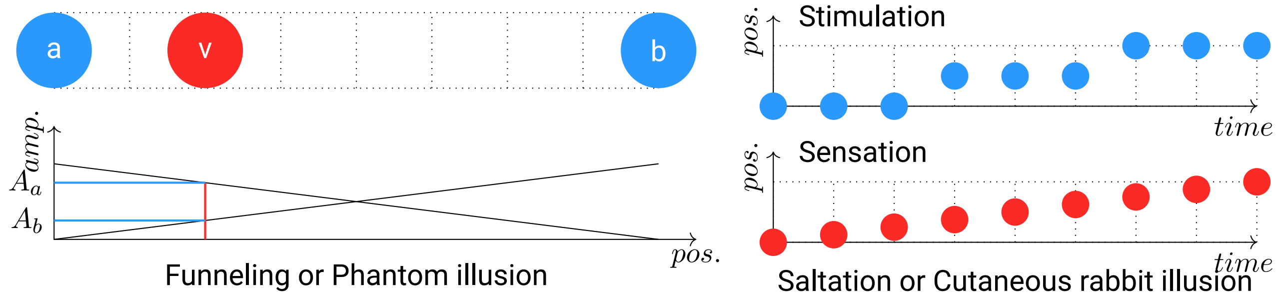
There are many ways to compute tactile signals, especially due to the diversity of actuation mechanisms and associated effects. Vibrotactile feedback is certainly the most common type of haptic feedback, because of its low price and simplicity. *Eccentric Rotating Mass* (ERM) actuators are widely used for these specific reasons. The mechanical effect results from the centrifugal force of the rotating mass. There is a delay before the motor spins fast enough, and inertia when the signal stops. Therefore there is no easy way to control both the frequency and amplitude of the produced mechanical effect with these actuators. *Voice coil* actuators work similarly to speakers, and they can actually be controlled with sound systems. The only difference is the frequency range, which is 1-1000Hz, compared to 200-20000Hz for sound. They produce vibrations with a coil and a permanent magnet [171, 266]. The precision of these actuators enables fine control of both frequency and amplitude. Such precision is either used for encoding abstract messages called Tactons [37, 120], or reproduce tactile effects such as button clicks [157, 176].

Pin arrays are essentially used to render patterns [**198**, **202**]. Each pin is controlled individually, either up or down. *Variable friction* technologies change the perceived friction of a surface. Two methods can produce this effect: *electro-vibration* and *squeeze-film effect*. The electro-vibration mechanism uses a high voltage (hundreds to thousand Volts) to stick the user's finger onto the interactive surface [243]. The signal is a sinusoid with controllable amplitude and frequency [20]. The squeeze film effect uses a high-frequency signal (tens of thousand Hertz) that we cannot perceive directly. This vibration creates an air cushion between the finger and the surface so that this surface feels smoother [29]. Finally, there are non-contact tactile technologies that transmit tactile feedback through the air. One technique uses an array of ultrasound actuators. The interferences of ultrasound waves create a stress field that triggers the sense of touch [48, 124] Another technique uses air vortexes [114, 238]. The displacement of a large membrane inside a box moves the air inside, which can escape through a small circular hole. The vortex is created with the pressure and moves forward on several meters before dissipating.

Sense of touch

The objective when producing a physical effect with a haptic system is to stimulate the users' sense of touch. On Figure I.1, I separate the pure sensing part from the interpretation part. The body has sensors in the skin, muscles, tendons, and articulations [32]. These sensors are different types of nerves and similarly to electronic sensors, they transform physical effects into electric signals. This document will not cover this in detail. The interested reader will find explanations in many Ph.D. manuscripts about haptics [40, 63, 119, 188], including mine [**196**]. We will just mention cornerstone studies about the perception of touch that established the limits of tactile perception [96]. Interestingly, the perception of touch depends on the body part, sex, and laterality [261]. In particular, the authors established the range of frequencies humans can perceive (roughly up to 1000Hz), with a peak perception around 250Hz. This study is the motivation why most vibrotactile systems use a 250 Hz signal, sometimes modulated with another frequency [41]. My opinion on this choice is that other frequencies provide different and interesting sensations. They just require a higher amplitude to be perceivable. However, the amplitude produced with usual vibrotactile systems is sufficient for a large range of frequencies. This is why we typically used different frequencies in our studies [**112**, **113**].

The perception of touch is nevertheless not only a matter of received signal. Gibson showed that people are more efficient at tactile exploration with active touch [91]. This means the brain combines sensations with other information, including exploratory movements that resulted in these sensations. Given the diversity of touch sensations: weight, shape, temperature, etc., Lederman and Klatzky describe specific exploratory procedures that enable people to perceive these types of sensations [144]. This chapter will not get into details regarding the relation between action and perception, and the Figure I.1 ignores this phenomenon. However we will cover this specific topic in chapter III.



Funneling or Phantom illusion

Saltation or Cutaneous rabbit illusion

Figure I.2 – Examples of tactile illusions. Left: funneling, or phantom illusion; right: saltation or cutaneous rabbit illusion.

Independently of active touch, our perception of haptic signals is an interpretation of the brain. For example, people may perceive only one vibration while there are two distinct stimulation if they are too close in space [261] or time [102]. We can use such interpretations to create tactile illusions. For example, the funneling illusion, also called phantom sensations, produces a virtual vibration in between two points[3, 24]. To do so, one must place two actuators, one at each end. The amplitude of the signal on both actuators is a ratio corresponding to the desired position of the virtual vibration. For example, let A be the maximum amplitude and two actuators a and b placed 4 cm apart, as depicted on the left part of Figure I.2. If the amplitude of actuator a is $A(a) = \frac{A}{4}$ and the amplitude of actuator b is $A(b) = \frac{3A}{4}$, then the users feel a vibration between a and b, 1 cm away from a and 3 cm away from b. Tactile saltation (also called the cutaneous rabbit illusion) gives the illusion of a sequence of equally spaced vibration [88]. The tactile stimulation however is a repeated vibration on a smaller subset of locations. For example, with three actuators a, b, and c equally spaced on a straight line, the stimulation is three vibrations on a, three on b, and three on c, as shown on the right part of Figure I.2. The person feels nine equally spaced stimulations between a and c. Israr *et al.* combined both funneling and saltation to produce 2D tactile motions, not only in a straight line but also on curves [129].

1 Research questions

The haptic rendering pipeline described above illustrates the multidisciplinary aspect of such research. It also reveals the many pitfalls at all levels that can make users perceive something different than what was intended. Any step in this pipeline potentially introduces a drift with respect to the original message. For example, the resolution of commands could be too low, the mechanical response to the command can be non-linear, the quality of the contact between the haptic system and the user can be sub-optimal, the mechanical effects can be out of the perceptual range, and haptic illusions can alter the interpretation of sensations. While the work of researchers in specialized domains mentioned in Figure I.1 is to avoid, or at least minimize such drifts in their part of the pipeline, my work as an HCI researcher consists in selecting, adjusting, combining, designing, implementing, and evaluating such parts to build *interactive systems*. My research is therefore fundamentally interdisciplinary, and I combine, complement or replace the expertise of different domains, depending on the needs of each research project, and the expertise of my collaborators. My objective is to make interactive systems better than the sum of their parts.

1.1 Output vocabulary

From a practical point of view, when a new haptic technology emerges, their designers need information about their specifications: the size of actuators, voltage, required amplitude, resonant frequency, maximum force (continuous and peak), etc. They often simulate the behavior of their actuators and evaluate the relation between the command and the physical effect [205, 206]. However, they need to know the desired parameters and values of the physical effects they can produce in order to propose a suitable output vocabulary. Therefore we perform iterations of prototyping and evaluation of human factors in order to understand these requirements, and design *interaction techniques*. Interaction techniques are combinations of a device (part of the electro-mechanical system) and an interactive language (part of the software controller) [183]. We presented technologies for haptic devices above. Interactive languages are systematic descriptions of the inputs and outputs of an interactive device. Similar to programming languages, they have three levels: lexical, syntactic, and semantic.

Figure I.3 – Four parameters of the vibrotactile output vocabulary: frequency, amplitude, duration and shape.

The *lexical level* defines the basic vocabulary of the device. For example, we can control vibrations in frequency, amplitude, shape, and duration (Figure I.3). Hence the command for a vibrotactile actuator is an electrical signal made of these elements. The research challenge here is to bridge the knowledge gap between the engineering of actuators and human factors to find tactile parameters that users can perceive and distinguish. Such research is typically useful to guide the design of tactile actuators. We will illustrate this kind of research in section I.2.2.

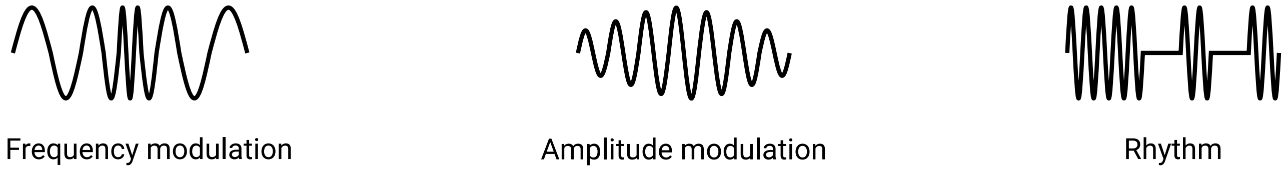


Figure I.4 – Three examples of haptic phrases: frequency modulation, amplitude modulation and rhythm.

The *syntactic level* combines lexical items to form haptic phrases. They eventually combine different modalities. The Figure I.4 shows several examples of such haptic phrases with vibrotactile feedback. Frequency and amplitude modulation create new kinds of feedback, that Brown *et al.* describe as roughness [37]. They also use sequences of vibrations to form rhythms. The challenge here is to find combinations of parameters that users are able to interpret together. I conducted such kind of research during my Ph.D. [**198**, **200**, **201**, **202**]. The difficulty is not only to identify such haptic parameters but also to evaluate them. By increasing the number of parameters and the number of values for each parameter, we quickly reach a point where it is impossible to evaluate every combination. In this case, we use other methods that basically consist of probing this space. We will discuss this in section I.2.2.

The *semantic level* represents the mapping between the haptic effect and its associated meaning. For example, if we create Tactons for meeting alerts, we can encode the kind of meeting with a rhythm, the importance with roughness, and the delay with spatial location. The combination of both parameters enables encoding every level of urgency for every caller ID. Figure I.5 illustrates this example of mapping between multi-parameters Tactons and hierarchical information.

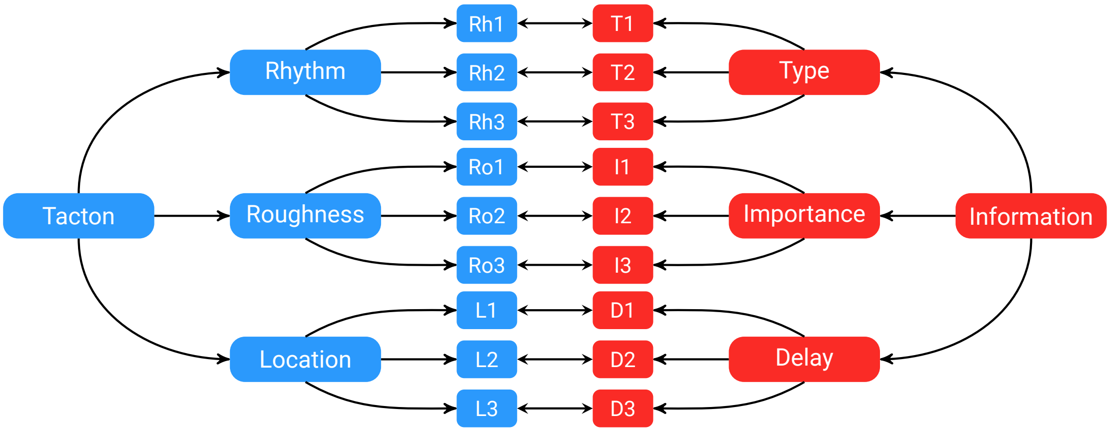


Figure I.5 – Illustration of a semantic mapping between 3-parameters Tactons and a 3-level information, adapted from [41]. 3 values of rhythm are mapped to 3 types of messages, 3 values of roughness are mapped to 3 degrees of importance, and 3 spatial locations are mapped to 3 values of delay.

Some studies about Tactons associate information to Tactons while evaluating the syntactic level [41, 120]. Therefore these studies either assume that the interpretation of the mapping is trivial, or they cannot make the distinction between the part of the metrics associated with the syntactic and the semantic level. Some of the Tactons studies I conducted during my Ph.D. also combined the syntactic and semantic level [**198**, **202**]. The Tactons parameters I used were mapped to direction and distance. However, the design rationale of these Tactons was to suggest an analogy between the Tacton parameters and the associated information. These Tactons used pin-arrays and coded the directions with lines or L shapes pointing towards the corresponding location. Unfortunately, we did not evaluate to which extent the semantic helped the interpretation.

In section I.2.1, I will address another issue with the semantic level of the haptic vocabulary. Most, if not all, haptic devices we use daily such as mobile phones or smartwatches have low-quality vibrotactile actuators, and typically only have one. In this study we addressed two challenges. First, we designed Tactons with an off-the-shelf smartwatch, which has only one actuator of poor quality. Second, we studied the interpretation of Tactons while users performed their daily activities.

1.2 Engineering and evaluation of haptic devices

Not only the design of interaction techniques depends on technological factors (the electromechanical system), but it also depends on human factors (both sensory and cognitive systems). The expansion of the haptic interactive vocabulary goes in both directions. On the one hand, the diversity of haptic sensations and body parts that sense them [145] motivate the design of a large diversity of haptic devices [231]. On the other hand, the frequent emergence of new haptic technologies expands the diversity of haptic sensations we leverage for the design of interaction techniques. This diversity of sensations and technologies is a richness, but also a limitation for the development and integration of haptic systems. Indeed, the sense of touch combines all these sensations, but no single haptic technology can produce all of them. Therefore, interactive systems which leverage the sense of touch only focus on the subset of haptic sensations relevant to their context of use. This means that the same type of research must be conducted for every new haptic technology. Further, the users' perception highly depends on the implementation, which is affected by many environmental factors such as temperature or moisture. The amplitude and randomness of such undesirable effects are generally higher with research prototypes than with commercial products. While these difficulties are essentially technical, they are difficult to quantify, and they have a strong effect on the precision, validity, and replicability of scientific results. In section I.2.2, I will discuss a project in which we investigated a way to mitigate this issue with physical objects reproducing a similar effect as the effect produced by a programmable friction haptic device.

1.3 Haptic properties of physical objects

Haptic devices are designed to stimulate the sense of touch such that we can control the haptic sensations that users will perceive. We described above the design rationale and pitfalls of such systems. Beyond these haptic systems, physical objects have their own haptic properties. They have a weight, mobile parts with resistance, tactile textures, etc. For example, physical buttons provide a haptic click and detents sensations. These haptic properties are structural and mechanical. For example, the size, shape, and layout of keyboard keys, as well as the keyboard slope, height, and profile are carefully designed [153]. The mechanical force required to push the keys and the click sensation are also systematically adjusted. The objective is to reduce the effects of fatigue and muscle strain, avoid incidental activations, and give immediate feedback to the user's action on the device.

The haptic properties of physical interfaces are however typically missing on multi-touch interfaces. Every widget feels like a flat surface. However, humans rely on haptic feedback all the time for everyday interactions with physical objects[161]. Efforts were made to restore this missing haptic feedback. For example, vibrotactile actuators can reproduce the clicking sensation of buttons [157, 176]. Several variable friction technologies can reproduce texture sensations [4, 20, 152]. Not only this haptic feedback improves subjective assessment of comfort, but it also improves performance in some cases. The clicking sensation on the Apple Magic Trackpad is impressive. It feels like a physical button. But we notice it is generated with a vibrotactile actuator because this haptic effect disappears when the trackpad is powered off. Further, studies show that vibrotactile feedback increases typing performance [121] on touch keyboards. I will discuss in I.2.3 the design of vibrotactile widgets for a touch dashboard.

1.4 Haptics and tangible interaction

Interaction with physical objects is interesting beyond its inspiration for bringing desirable haptic properties to touch interfaces. We interact with many objects in our environment all day. The haptic aspect of our interaction with everyday objects is richer than with interactive systems. All the objects in our environment have haptic properties, and the design of these objects leverage these properties. We can identify the key on the door on a keyring in our pocket, just by touch. Thanks to the tactile sensitivity of our fingers we can perceive the shape, size, and material of the key to some extent. We can identify ripe fruits and vegetables based on their hardness. We can use a TV remote in the dark because we locate the keys with proprioception. We can tell if an opaque bottle is empty, full, or in between because we feel its weight, and we can feel the liquid splashing inside.

To leverage these haptic properties of everyday objects, there is indeed a compelling intersection with *tangible interaction*. Ullmer and Ishii described Tangible User Interfaces (TUI) this way: “TUIs will augment the real physical world by coupling digital information to everyday physical objects and environments.” [128]. Tangible interaction not necessarily focuses on haptic properties. However, the fact that TUIs use everyday objects means that TUIs inherit their haptic properties. Further, physical objects have interesting properties, among which the possibility to change their shape. They can fold, squeeze, some parts are mobile. Changing the shape of objects can change the way we use them, therefore their function. For example, changing the shape of a knob changes the way we grip it [168]. A knob can even become a slider [136]. In section I.2.4, I will discuss how the actuation of computer peripherals can bring them new interactive properties.

2 Contributions

The previous section presented challenges, but also opportunities for the design and implementation of haptic interactive systems. These are questions I regularly address in my research. The way I describe the research projects below does not necessarily follow the narrative of the papers that resulted from these works. I rather discuss the contribution of these studies to the research questions above.

2.1 Tactons for activity monitoring

Using haptics as an output vocabulary to transmit information to users directly stems from my Master and Ph.D. work. I had the privilege and pleasure to collaborate with Stephen Brewster, who is also the inventor of the concept of Tactons [37]. I worked on several Tacton sets at the time: active [**200**] and passive [**201**] force feedback, as well as pin-array Tactons [**198**, **202**]. Here, I will discuss another project on Tactons called Activibe, which was a collaboration with Jessica Cauchard, James Landay, and Janette Cheng from Stanford University [**55**].

The idea was to design haptic feedback for activity monitoring. Fitness trackers became mainstream in the last decade. People use wearable devices such as smartwatches or bracelets to measure their activities, such as their daily steps [60]. They typically define daily objectives and need regular reminders to help them meet their goal. These activity trackers provide visual information with small screens or LEDs. Therefore users have to actively look at their devices to know their progress. This is a limitation of the incentive aspect of activity monitoring towards behavior change. Still today, the solution in consumer electronics products is a simple vibration that encourages users to look at their device to check for information visually. In this project we decided to provide progress information with vibrotactile feedback, thus not requiring users to look at the device while they perform their daily activities. We describe below the design and evaluation of these Tactons.

2.1.1 Tacton design

The context of activity monitoring implies several limitations on the design of the Tactons. We believed it would be difficult for us to motivate activity manufacturers to include a better tactile actuator than the existing ones. Therefore we decided to use an off-the-shelf smartwatch: the Pebble[1]. It includes an eccentric rotating mass (ERM) actuator. This low-resolution DC motor constrains the choice of tactile parameters to amplitude, duration, and rhythm. Given that the perception of amplitude varies from one user to another or the way the smartwatch is fastened, we use the duration and rhythm parameters only.

Information

ActiVibe was designed as a set of vibrotactile icons to represent progress. As activity performance is generally evaluated as a percentage or as a value on a scale, we created vibrations corresponding to the values 1 to 10, intending to represent 10% increments. Since there was no prior encoding of discrete numbers found in the literature using duration and rhythm only, we first had to determine the best encoding pattern for ActiVibe.

Figure I.6 shows a visual representation of the pattern sets that we evaluated. Each individual squiggly line represents a single short pulse, while a long line represents a longer vibration. We first designed the series of vibration sets (A-E) that were evaluated in a first laboratory setting. The results of the

---

[1] https://en.wikipedia.org/wiki/Pebble_(watch)

first study helped us to design pattern F, which was then compared in a second laboratory study to the best sets from the first study (A, C, and E).

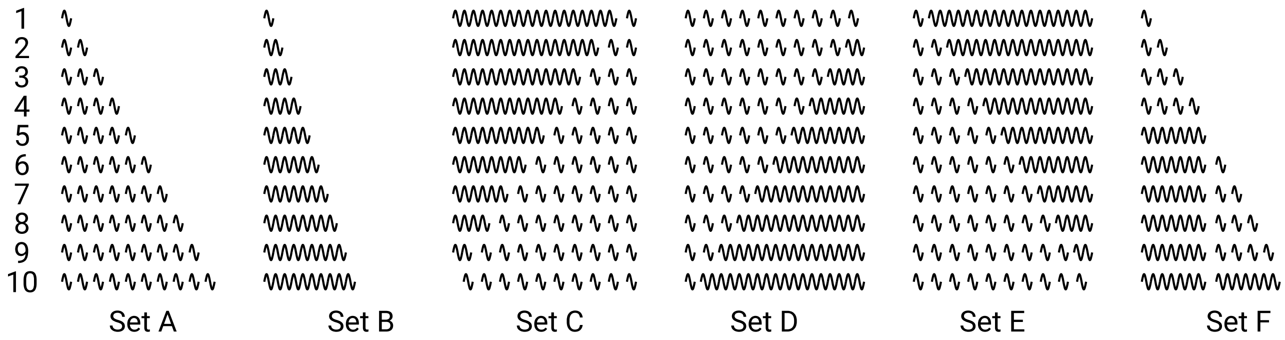


Figure I.6 – Visual representation of the 6 pattern sets we evaluated in two laboratory studies, and a longitudinal study.

Design Rationale

Our design is driven by the semantics of the values we want to convey. We intend to represent discrete values of a progression. We explored two possibilities: 1) represent the actual value only; 2) represent the value as well as the scale.

When representing the actual value only, the duration of the vibrotactile pattern depends on the value it represents (Figure I.6, sets A & B). The pattern has a short average duration, and thus it may be hard to understand the distance to the end of the event represented. We distinguish two variations. In set A, each value is represented by a series of short pulses, separated by short pauses. In set B, a continuous vibration represents each value with the duration corresponding to the value.

The disadvantage of representing only the value is that even if the user has an idea of the current value, there is no clue about the distance between this value and the maximum value. Introducing a scale enables the positioning of a value relative to the beginning and the end of a progression. Sets C-E represent both the value and the scale of a progression in several ways. Either the current value is represented by a series of short vibrations and the scale by filling the sequence with a long vibration, either before or after the value (Figure I.6, sets C & E), or the current value is represented by a long vibration and the scale is represented by filling the sequence with a series of short vibrations (Figure I.6, set D). Set F was defined using the results from the first laboratory study described in Section I.2.1.2 as a combination of short pulses and long vibrations.

Table I.1 summarizes the design space of the vibration pattern sets we developed for our laboratory studies. We were interested in knowing which sets of icons were the most suitable for representing progression values, and what is the best precision we can obtain using these representations. We can clearly see that we did not explore all the possible combinations. This is mostly due to the limited number of sets we could reasonably evaluate in a laboratory study. Other dimensions are also relevant, like adding a warning vibration before the actual code, to help users paying attention to the

Set	Information	Representation	Padding
A	Value only	Shorts	
B	Value only	Long	
C	Value + scale	Shorts (padding: Long)	Before
D	Value + scale	Long (padding: Shorts)	Before
E	Value + scale	Shorts (padding: Long)	After
F	Value + scale	Long + Shorts	

Table I.1 – Design space of the Activibe pattern sets.

signal while doing their daily activities. However, in some cases, the padding vibration more or less plays this role. In any case, we analyzed this aspect in the longitudinal study.

Parameter values

We ran short pilot studies to estimate the shortest vibration pulse that users are able to perceive with the apparatus (Pebble watch), as well as the shortest pause between two vibrations so that users can distinguish multiple short pulses. We obtained 100ms for the pulse and 150ms for the break between two pulses. Therefore, the longest patterns last less than 3 seconds.

In the longitudinal study, we increased the vibrations durations given that participants will not be paying as much attention to the vibration as they had in the laboratory studies. Short vibration lasted 150 ms, long vibration 600 ms, and pauses were 200 ms long. Finally, the pre-vibration was long vibrations of 700 ms.

2.1.2 Evaluation

I will not give the details about the evaluations in this study. I will rather discuss the main findings with hindsight. Readers interested in the details about the experimental protocols and the results should refer to [**55**].

Counting VS duration

The first findings are related to the participants' strategy for interpreting the patterns. Either they counted the short vibrations or estimated the duration of the long ones. The results showed that it is easier to count vibrations than to estimate the duration. This is at least true within the tested range: 1-10, and 3 s maximum. Indeed, we believe it would be much more complicated to keep the count with higher values. Durations are much complex to interpret, in particular because of the perception of time. Other modalities can influence the perception of the duration of haptic signals [103].

Now, sets C-E contained both short and long vibrations that potentially both encoded the target number. Participants clearly stated that they counted the short vibrations to identify the answer. Therefore, with set D in which the padding sequence of small vibrations before the long vibration supposedly represented the answer, participants reported counting backward.

The set F was designed after the first user study. The rationale was to reduce the counting task by using a long vibration for marking 5, and additional short vibrations to represent increments. Therefore, users have to count up to 5 short vibrations, and they do not have to estimate the duration of long vibrations. We compared this set with sets A, C, and E in a second laboratory study. Participants identified these patterns with the same or better performance than the patterns of the other sets.

Distance between input and answer

In the context of activity monitoring, the distance between the correct value and the user answer matters. Indeed, interpreting a 60 % progression instead of 70 % is a lesser concern than interpreting 20 % instead of 70 %. Therefore we analyzed this precision as a measure of distance between the input and the actual answer. We observed differences in terms of precision across pattern sets. For example with sets A and B, the precision gets worse as the number is high. However, the proportions are larger with set B, which confirms the idea that counting short vibrations is easier than estimating the duration of long vibrations.

Interestingly, with set D the participants' precision was higher on the middle-range values. We explain this with confusion due to the numbers being represented by the duration of the long vibrations at the end of the patterns. Since participants reported they counted backward, this result shows that they sometimes forgot to do so. It had a strong influence on the result, which is another reason to reject this pattern set.

Pre-attentive signals

Many participants mentioned that with the pattern set C, they liked the long vibrations, which helped them focus on the upcoming short vibrations and count them. We believed this aspect would be important in a real context when users receive information while they perform their daily activities. Therefore we conducted a 28-days longitudinal study. Participants were presented with random numbers of set F. Half of them received a pre-attentive vibration of 700 ms between the actual pattern. In order to make sure participants do not expect the notifications, they were sent twelve vibrations per day at semi-random times within a one-hour window between 7am and 8pm. This schedule helped to cover different activities from the users such as their commute, when they bring their kids to school, their day at work, and some evening activities.

The results did not show a significant difference in terms of answer accurracy rate. However, when discussing with participants after the experiment, most of the groups who did not have the vibration answered that a pre-vibration would have been useful. Almost all of the participants in the group who had it answered that the pre-vibration was useful to them. So despite a lack of quantitative improvements, we recommend providing a pre-attentive vibration because users prefer to have it.

2.1.3 Discussion

Participants identified the patterns with high accuracy both in the laboratory (96%) and in-situ (89%). We note that in the longitudinal study, the patterns were presented in random order. However, in an activity monitoring scenario, they would be presented in increasing order. Hence we conjecture that participants could identify them better.

The main limitation of this work is certainly its scalability. The most efficient technique for interpreting the patterns is counting the vibrations, and distinguishing short and long vibrations. We only used 10 patterns in each set, but if we would like to expand it to larger sets, we will need another way of coding numbers.

Finally, our starting constraint of this work was using an off-the-shelf wearable device. We used a Pebble watch, which includes a low-quality ERM actuator. Indeed, it strongly limited the output vocabulary we could use. However haptic technologies are evolving, and we can expect better actuators even in consumer electronics products in the future. Therefore it would be interesting to push this work forward with better vibrotactile actuators [171, 266], and use other tactile parameters such as frequency.

2.2 Tactile textures with programmable friction

This next contribution is also about the extension of the haptic output vocabulary. However, contrary to the previous contribution, in this project we studied the output vocabulary for a new kind of tactile technology: programmable friction. This work was mainly done by Ludovic Potier who was a postdoc with a cognitive sciences background that I co-supervised with Nicolas Roussel and Géry Casiez when we were part of the Mint team. The design and implementation of the tactile device we used are the result of years of research by the experts in electrical engineering in the Mint team: Frédéric Giraud, Michel Amberg, Betty Semail, and their students.

Two different technologies exist for changing the perceived friction of a surface. The first one is electrovibration. It uses a high voltage signal on an electrode to stick the user's finger on the surface [243]. Therefore this technology increases the perceived friction. The second technology leverages the squeeze film effect to reduce the perceived friction [29]. Technically, the whole surface vibrates at a high frequency, typically tens of thousand Hertz [4]. This frequency is way beyond the tactile sensitivity range. However, it creates an air cushion between the surface and the finger, which makes the surface feel more slippery. This project has two orthogonal challenges. First, the technology was new and developed by colleagues with expertise in electrical engineering. Therefore we had to work with research prototypes, called Stimtac, that were evolving in parallel to our own studies. Second, we had to characterize the tactile sensations produced with this technology in order to define the output vocabulary. Our results guided the design of the devices and the improvements of the devices enabled better sensations. The co-evolution of both research activities pushed forward this technology, and it is today developed by Hap2U[2].

---

[2] https://www.hap2u.net/

2.2.1 Technical considerations

Our colleagues built several prototypes. They glued an array of piezoelectric ceramics under the tactile surface. Their objective is to control the actuation of the ceramics to deform the whole surface consistently [29]. The ceramics are activated by two electrical signals with a phase shift between each other. The amplitude is maximized with a 180° phase shift, and zero with a 0° phase shift. The perceived friction depends on the vibration amplitude of the whole surface. Therefore the device cannot produce different tactile sensations on several fingers touching the surface. It is however possible to change the amplitude over time. In particular, the prototypes have sensors to locate the contact points so that we can adjust the tactile feedback depending on the finger position on the tactile surface.

The typical feedback with this technology consists of steps between two slickness values [28]. Previous work shows for example that using sticky targets increases target selection performance [52]. Authors showed that the increase in performance is due to the return of information, and not to a physical effect.

2.2.2 Perception of programmable friction

The connection between the command and what users perceive through their fingers is not trivial though. This is a typical example of the phenomenon discussed at the beginning of this chapter and illustrated on Figure I.1. Both the relation between the command and the vibration amplitude and the relation between the amplitude and the sensation of slickness are not linear [29]. They are affected by many environmental factors such as air moisture, temperature; surface properties such as material, cleanliness; and fingers aspect: tribology, cleanliness, moisture. There are differences in performance between prototypes. Some of these differences are due to their hand-made nature, especially the ceramics gluing. Others are just due to the updates resulting from lessons learned with each prototype. The major consequence of this is that the actual values of psychophysics studies results are tied to a particular prototype. They are hard to generalize. However, lessons learned are useful for improving the technology and understand its benefits and limitations.

The first study in which I was involved consisted in evaluating the Just Noticeable Difference (JND) of friction between two adjacent zones. The objective is to measure the useful resolution of the device. With this information, our colleagues could optimize the power consumption of the device, and we could design more complex patterns such as textures. This work was not published, therefore I will get in more details below with a new statistical analysis.

Methodology

We used a one-up/two-down adaptative method, which enables a faster convergence to the JND value [148]. This method consists in reducing the intensity of the signal after two good answers and augmenting it after one wrong answer (Figure I.7). The evaluation of each trial used a 3-alternatives forced-choice (3-AFC). This means we presented participants with 3 configurations. One of them had a signal: a difference of friction between the left and right sides of the device. This difference of friction corresponds to the intensity of the signal. The friction was uniform on the whole surface for the two other configurations. Participants had to tell which configuration presented a signal. This procedure reduces the chance level to 33 % and avoids anticipation biases. The position of the signal among the 3 configurations was random, and the side presenting the reference level was random as well.

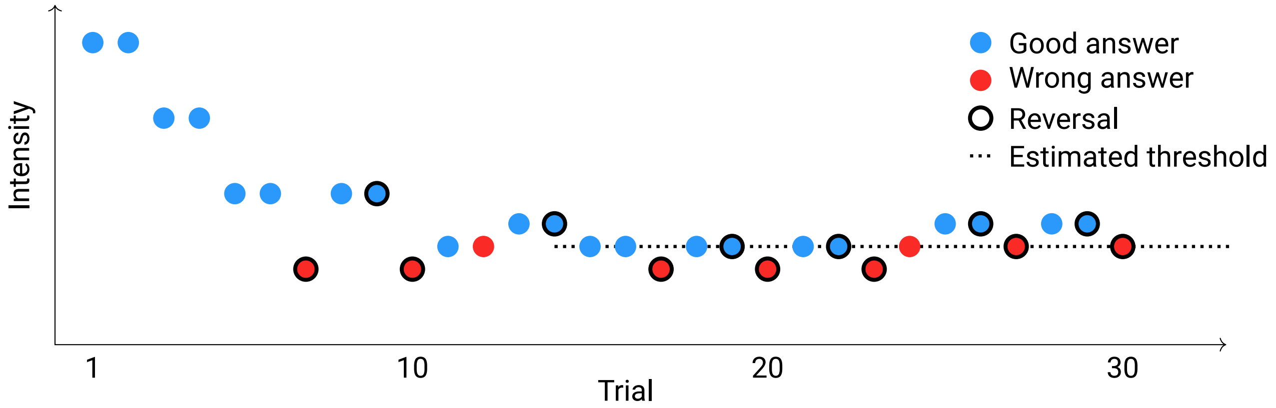


Figure I.7 – Example of an adaptative procedure.

The first trial of each block used the largest difference for the chosen reference value. For example, with a reference level of 0° the other value was 180°. Then, the value decreased by 1.8 dB after two good answers in a row and increased by 1.8 dB after a wrong answer. After three reversals we reduced the increment/decrement to 1.2 dB since participants went closer to the perception threshold. The block ended at the 13th reversal. We estimated the threshold by averaging the values corresponding to the last 10 reversals.

We used six reference levels, one for each block. The values were spread out linearly between the minimum and maximum command (0° to 180° phase shift). For the first three values (0°, 36°, and 72°) we searched the minimum greater value for which participant could feel a step. For the last three values (108°, 144°, and 180°) we searched the minimum lower value for which participant could feel a step.

The Stimtac device version we used in this experiment is depicted on Figure I.8. The surface is 75 × 40 mm, made of Copper-Beryllium, covered with a *Filmolux® easy clear matt* plastic sheet. 36 piezo-electric ceramics are glued underneath. 35 of them are actuated to vibrate the surface, and the last one is used as a sensor to measure the vibration amplitude. The device was connected to a PC through USB, on which the experimental application was running. A plastic cover, not visible on the picture, was placed over the surface to reduce the interactive surface to 70 × 20 mm to reduce the variability of amplitude and guide the user on a lateral movement.

Figure I.8 – The Stimtac device produces a variable friction haptic feedback.

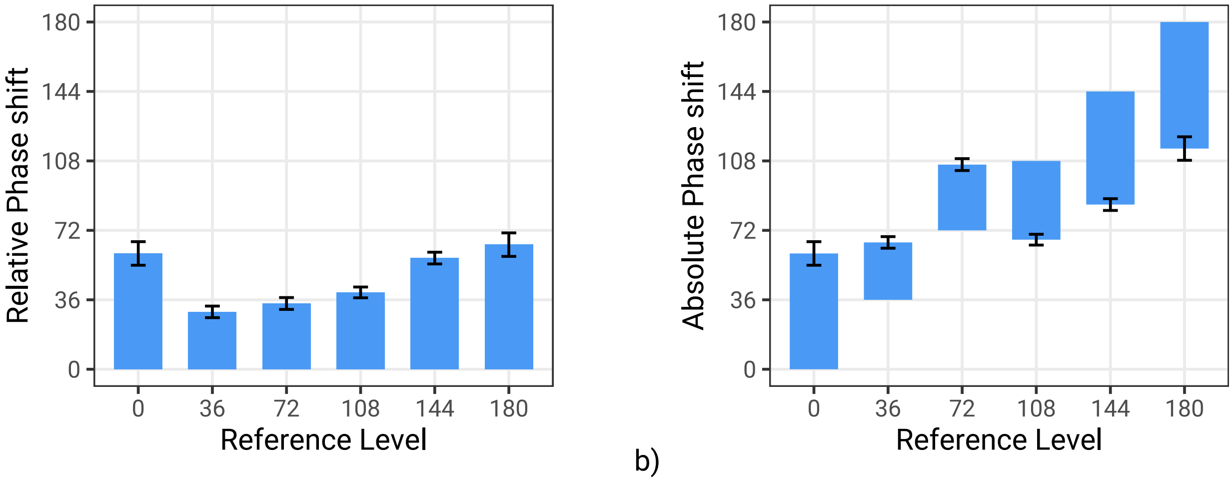
We instructed participants to use the index finger of their dominant hand only. They wore a noise-canceling headset to avoid guesses with audio cues. We cleaned the surface with isopropyl alcohol before each block. To make sure participants felt the signal in both directions, we instructed them to perform at least 3 back-and-forths between the left and right zones of the surface. The experiment application showed a pressure bar and participants were instructed to remain in a specified range.

The experiment started with a training phase in which signals of different intensities were presented to participants. We presented them with a low-intensity signal to make sure they will be able to perform all 6 blocks. All the participants felt this signal. The application showed three visually identical items representing the three configurations of the trial. Participants could switch between them with the `Space` key but could not go back to the previous ones. The current item was highlighted, and we limited the exploration time for each item to 12 s. After exploring the three items, participants had to indicate which one presented the signal with the `1` `2` `3` keys. Participants did not receive any feedback regarding whether their answer was good or wrong.

12 participants took part in this experiment, all of them were right-handed, their mean age was 27.7 years old, and none of them had a known tactile sensitivity issue. Participants performed 2 sessions of 6 blocks with pauses between blocks and the two sessions happened on a different day. Each block typically consisted of 30 to 50 trials and lasted about 20 minutes. The order of blocks was balanced with a Latin square.

Results and discussion

In this analysis, we used 10 Reversals × 6 Levels × 12 Participants = 720 trials. Table I.2 shows the mean JND value and standard deviation for each reference level. Figure I.9 shows two representations of the results. On the left, for each level, the bars represent the JND value. On the right, for each level, the bottom of the bars represents the reference level and the height of the bar is the JND value. It gives an absolute view of the phase shift associated with the reference level and the JND. For both charts, the error bars are 95% confidence intervals. On the right side, the error bar is on the variable end because the reference level did not change.



a) b)

Figure I.9 – Results of the experiment about the JND between two adjacent zones. a) The relative phase shift JND corresponding to the difference between the reference level and the variable value. b) The absolute values of the reference level and the JND. Error bars are 95% confidence intervals.

Reference level	Mean JND	Standard Deviation
0°	60.0°	33.8°
36°	29.8°	16.6°
72°	34.1°	17.0°
108°	39.8°	15.5°
144°	57.6°	16.8°
180°	64.6°	33.7°

Table I.2 – Mean JND values and standard deviation for each reference level.

A Shapiro-Wilk normality test shows the data does not follow a normal distribution ($p < 0.0001$). We performed a boxcox correction ($\lambda = 0.18$), however the data distribution remained not normal ($p < 0.0001$). Therefore we analyzed our data with a Kruskal-Wallis rank-sum test, which showed significant differences ($\chi^2(5) = 204.45, p < 0.0001$) The posthoc analysis with pairwise Wilcoxon rank-sum tests shows two groups of reference levels with significant differences between the groups. The difference between the JND and the 0°, 144°, and 180° reference levels was significantly higher than with the 36°, 72° and 104° ($p < 0.0001$ for all differences, except between 72° and 108° for which $p < 0.05$).

First of all, it is important to remind that the squeeze film effect reduces friction. Therefore, when the command is high, the surface is more slippery than when the command is low. On the figures, the left and bottom sides correspond to high friction, and the right and top sides correspond to low friction. Hence, we explain the fact that the higher difference between the reference levels and the JND are on the edge values (0°, 144°, and 180°) with the non-linearity of both the effect produced by the command (effect on 0°) and the perception of the mechanical effect (effect on 144°, and 180°).

The overall and conservative recommendation here is to use differences of phase shift greater than 100° to create patterns with this implementation of Stimtac (larger mean JND plus standard deviation). In any case, it is unlikely users can perceive a difference of phase shift lower than 10° (lower mean JND minus standard deviation). Unfortunately, at the time I am writing this I have no access to the amplitude measurement data. This makes it impossible to draw more general recommendations. The measurement with a laser vibrometer of a 180° command gives a 2.2 µm amplitude of vibration. It is however difficult to measure an actual friction value because it depends on the force applied on the surface, which varies when users move their fingers to explore the surface.

2.2.3 *Output vocabulary: Tactile Textures*

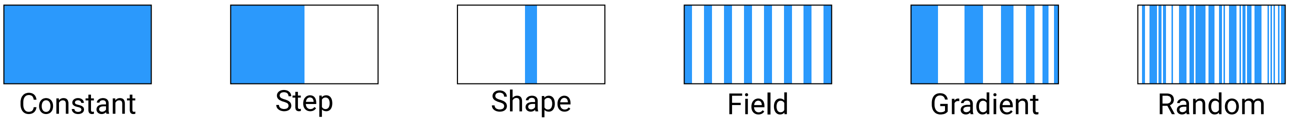
The exploration of low-level parameters such as the difference of friction between two adjacent zones as discussed in the previous section enables the design of higher-level tactile parameters such as textures. The literature on the perception of tactile textures is vast. It is an active research topic in psychology for decades. However, we struggled to find an exact definition of a tactile texture. Many studies like [146, 147, 268] focus on roughness. Roughness can be seen as a fine-grain texture for which users do not feel individual elements. There is research in the physical world, with studies on fabrics [192, 195]. In the digital world, there is a notion of patterns [89], that are sometimes direct translations of visual textures [127]. Schellingerhout studied repeated patterns and describes 3 configurations [227]. The first one is *smooth textures*, which corresponds to a uniform signal without repetition. The second one is a *homogeneous texture*, which is a repeated pattern without variations. The last one is a *gradient texture*, which is a repeated pattern of different scales. Therefore the multiscale aspect nature of texture is an important element to consider [122].

Definition

We took inspiration from this research topic to propose a definition of tactile patterns and texture as follows [**208**, **209**].

Definition 1 We define a *tactile pattern* as a spatial or temporal sequence of shapes distinguishable from a background, periodic or not.

Figure I.10 depicts examples of 1D tactile patterns of increased complexity. The first one is a *constant* pattern, for which the device uses the same friction command over the whole surface. Like every other pattern type, we can adjust the friction level. The second pattern is a *step*. It corresponds to a frontier between two adjacent zones which have a different friction value. It is the smallest building block we will use later for designing more complex patterns and textures. We can adjust its position and direction. The third pattern is a *shape*: a localized pattern distinct from the background. It has two steps so that the user can move through it. Its parameters are position and size (width). The fourth pattern is a *field*, a regular repetition of a shape. With a sufficiently low size and a high number of repetitions, users are not able to count the item while exploring the surface. They rather have a sensation of roughness. One of the parameters is the duty cycle, which is the ratio between the signal size and the period size. We can also adjust the number of repetitions and the width, which together controls the density of the pattern. The fifth pattern is a *gradient*, repetition of a shape, with a variable size. There are many ways to increase or decrease the size of shapes of a gradient: linearly, exponentially, etc. Finally, the sixth pattern is a *random* series of shapes. We can control the minimum and maximum sizes of shapes.



Constant Step Shape Field Gradient Random

Figure I.10 – Examples of tactile patterns with increased complexity. The background is white and the pattern is blue.

Definition 2 We define a *Tactile texture* as combination of one of several patterns on several dimensions or at different scales.

Textures can have patterns of different scales, providing different levels of detail. For example, physical objects can have a rugosity due to the microstructure of their material and a higher grain structure due to the carving of their surface. Figure I.11 shows two examples of tactile textures. The first one is a sequence of tactile patterns of different types: field, constant, gradient, and random. Users can most likely identify different parts in the textures, without necessarily locate their boundaries. The second one repeats a field-type pattern, which is itself a repetition of a shape. We can also see this texture as the frequency modulation of two signals. We can obviously use the same structure and parameters than with vibrotactile Tactons (see Figure I.3 and I.4). However, the range of friction we can produce with this device is not as wide as the range of vibration amplitudes we can produce with state-of-the-art vibrotactile actuators. Therefore we cannot simply translate vibrotactile Tactons into tactile textures with current programmable friction devices. It would be interesting to perform JND studies with newer devices that produces a higher quality variable friction feedback, such as devices made by Hap2U[3].

Contrary to vibrotactile Tactons, tactile texture can easily extend to multiple dimensions. Figure I.10 shows 1D textures, but it can extend to 2D, 3D textures or even 4D with a temporal dimension. Other technologies are indeed required for three spatial dimensions, like ultrasound actuators [48]. The examples of tactile patterns and textures we presented do not necessarily cover all possibilities. However, it is an illustration of the expressive power of tactile textures made of elementary steps. We will next discuss an evaluation of users' perception of tactile patterns.

---

[3] https://www.hap2u.net/

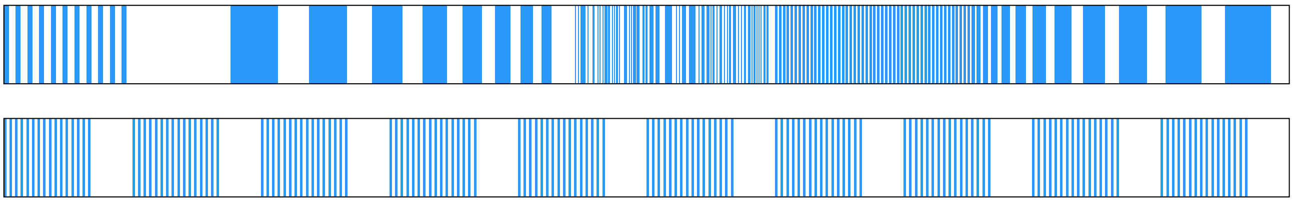


Figure I.11 – Examples of tactile textures. The first one is the combination of a series of various pattern types. It starts with a field, then a constant, a decreasing gradient, a random pattern, another field and finishes with an increasing gradient. The second one is a repetition of a field-type tactile pattern.

Evaluation of tactile patterns

The first interesting research question is whether users can distinguish the patterns. The way of evaluating this is not trivial though because the design space is large and it is difficult, not to say impossible, to present all of them with sufficient repetitions for an accurate analysis. Therefore we must evaluate a subset of all possible patterns. Recent work addressed this issue with a new method for sampling the design space [67]. At the time, we used multidimensional scaling (MDS) because of the multidimensional nature of our patterns. This method consists in asking participants to group items and use the number of times they are in the same group as a similarity metric. Then we compute the number of required dimensions and map each item to a position in space so that the distance between two items is proportional to their dissimilarity. Finally, we identify clusters of similar patterns. This method was already successfully used to evaluate Tactons [74]. Figure I.12 shows the patterns we evaluated in our study **[208]**. They are made of 5 shapes and 7 densities, for a total of 34 patterns (the lowest density vertical line and squares are identical).

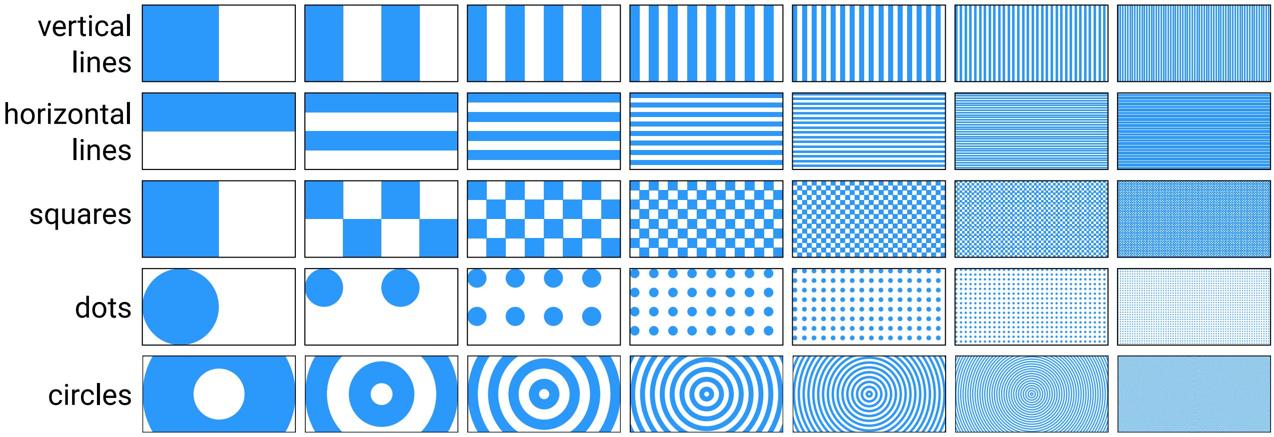


Figure I.12 – Tactile patterns used in the MDS experiment. They are made of 5 shapes and 7 densities.

The secondary research question is whether the perception of tactile textures with a variable friction device is similar to the perception of a similar pattern on a physical surface. We can expect differences because this technology cannot produce edges that we can feel under the fingertip. With a squeeze film effect device, edges are produced by changing the friction of the whole surface depending on the finger position. Therefore in our experiment, we compared tactile patterns rendered on a Stimtac device, and 250g dull coated paper cards as tangible surfaces. The stimulation on the paper is created with transparent ink, which feels stickier than the paper. The stickiness depends on the number of printed layers, printed with an HP Indigo Digital Press® printer. We performed a pilot study with 3 users to define the number of layers required to match the stimulation with the STIMTAC. We opted for 30 layers, which makes a 0.05 mm thickness. Both STIMTAC and cards have the same size: 8cm wide and 4cm high.

Discussion

Interested readers will find more details on the experimental details and the full analysis in the corresponding paper [**208**]. Here I will focus on the main findings. We designed the patterns with several shapes and densities. Therefore we expect participants to group patterns of the same shapes or densities. However, the MDS analysis fails at identifying a good set of clusters with 2 or 3 dimensions. This means that this is unlikely there is a consensual grouping strategy among users and conditions. We observed participants often grouped vertical and horizontal lines per shape in both conditions. We hypothesize this is due to the fact that the sensation with these two shapes is different whether you explore them horizontally or vertically.

Next, we analyzed the composition of pattern groups made by participants. We observed if groups contained the same SHAPE, same DENSITY or if users MIXED shapes and densities. In the paper condition, participants used more often a SHAPE or DENSITY strategy than a MIXED strategy. In particular, when they made groups of 6 or 7 items, they dominantly used a SHAPE strategy. Interestingly, in the Stimtac condition participants mostly used a MIXED strategy, and made more groups of 2 items. Because the grouping strategy was different in the two conditions, we assume that they perceived the patterns differently with the paper cards and Stimtac.

2.2.4 Conclusion

This project is a typically interdisciplinary project in which research was made on two fronts at the same time. On one side, our Electrical engineering colleagues designed, simulated, and implemented the device itself. On our side, we evaluated the perception of the haptic effect by users. Both research activities benefitted from the other. Our main contribution was the evaluation of variable friction parameters for the design of an output vocabulary with programmable friction devices.

We first worked on the lexical level and evaluated the JND of steps for six reference values of friction. We observed differences of perception between reference values. We attributed these differences to both the non-linearity of the mechanical effect resulting from the command and the perception of the mechanical effect. As a result, the signal commands on the device were adapted.

Then we worked on the syntactic level and proposed definitions of tactile patterns and textures. We compared the users' estimation of similarity of tactile patterns made of different shapes and densities implemented with a Stimtac device and with coated paper cards. We observed differences in grouping strategy across conditions, suggesting differences of perception between the haptic device and a similar physical representation of textures. It means that results about perception made with research prototypes are hard to generalize. Using physical props is an alternative to evaluate best-case scenarios. However, devices cannot reproduce all haptic properties of the physical world.

2.3 Printed vibrotactile widgets

In the previous section, we investigated the output vocabulary for a new device providing a new type of haptic feedback. We did not explore a particular context or application but rather studied the possibilities and limitations of the technology. In this project, we are interested in vibrotactile feedback, which is well covered in the literature. We were however interested in a particular case: restoring haptic feedback on touchscreens. Indeed touchscreens have many advantages compared to physical interfaces. They can be updated. They have no mechanical parts that wear over time. They are flat, so they are easy to clean. However, physical interfaces such as buttons and sliders have interesting interactive properties. They have a relief that enables people to locate them with touch. They provide haptic feedback when they are operated: click sensation, detents, stops. These properties guarantee invaluable usability benefits, such as giving continuous feedback of users' actions and the discoverability of interactive elements.

This work was part of the H2020 Happiness project. I was the leader of the Human Factors work package, as well as the leader for Inria that was represented by the Mjolnir and Hybrid research groups. In this project I supervised Christian Frisson during his postdoc in the group, as well as Julien Decaudin who was the engineer who implemented the software library, demos, and electronic prototypes.

The manufacturing process of touch interfaces such as dashboards augmented with haptic feedback is complex. Mechanical actuators must be attached underneath such that the vibration transmits to the interactive places of the surfaces. In this project, we investigate a new kind of actuator. They are printed on a flexible substrate with a piezoelectric ink. Therefore, these actuators can be embedded in plastic injection molds when dashboard parts are produced. They can even be integrated into curved interactive surfaces. Our colleagues at CEA LITEN[4] designed and implemented the actuators, and Walterpack[5] worked on the integration of actuators in the plastic injection mold. We prototyped the driving electronics and clamping system, designed tactile widgets, and implemented the software part and driving electronics for a dashboard prototype that was showcased at the Geneva Motor Show 2017[6] (Figure I.13).

---

[4] Pictic platform

[5] http://www.walterpack.com/

[6] Mojave concept car, produced by the Esperra Sbarro school

Figure I.13 – The Mojave concept car designed by the Esperra Sbarro school, showcased at the Geneva Motor Show 2017. We implemented the software part of the dashboard and the driving electronics.

2.3.1 Experimental platform

The design of these actuators is a trade-off between their size and thickness, the number of layers, and the signal voltage. Our colleagues who designed and implemented these actuators performed FEM simulations and measured the response to signal with laser vibrometers [205, 206]. They gave us several prototypes that we could use to design vibrotactile widgets. The first prototype (top of Figure I.14) had six buttons embedded in a molded plastic dashboard part. The clamping area was a circle around each actuator. It defines the area on which the vibration propagates, therefore this vibrotactile feedback is localized. The second prototype was a bare flexible substrate with seven actuators. We designed a clamping frame around the actuators so that the vibration could propagate on the whole length of the slider. We added tension strings to tighten the frame so that the clamping area could resonate. This is the same approach as a drum head on a drum shell. The capacitive touch sensor, printed with silver ink, is under the actuators. The setup is depicted at the bottom of the Figure I.14.

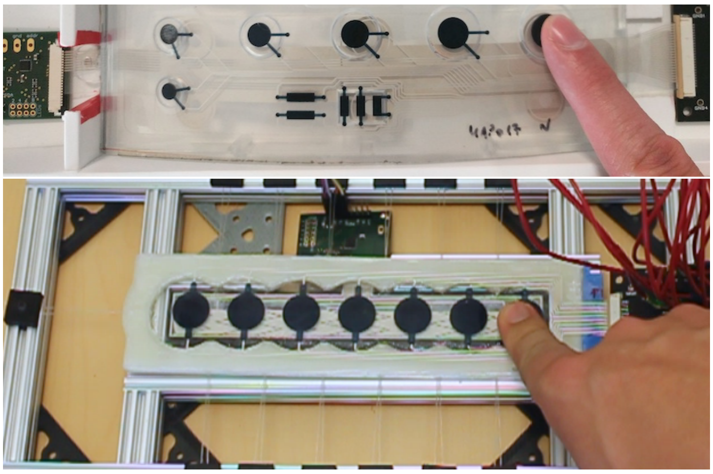


Figure I.14 – Top: Printed buttons. They are clamped individually. Bottom: Printed slider on a clamping frame. The slider is made of 7 piezo actuators. Tension strings make sure the substrate is tight.

We built a custom PCB with piezo drivers[7] (one per actuator), and the capacitive sensing chip [8]. The chips communicate with the mainboard that runs the application[9] through an I2C bus. The drivers have two modes of operation. Either we play pre-recorded signals, or we provide an audio signal in the tactile frequency range. The pre-recorded signal solution is easier because it does not require additional hardware. The audio solution requires sound generation hardware (typically a sound card with one channel per driver), but the signal generation is more flexible. We chose to use the audio mode for designing iteratively the tactile signals. We generated the audio signals with Purr-Data [43]. Then when used the other mode when the tactile signals were validated.

2.3.2 Vibrotactile widgets

The idea of using vibrations to simulate button clicks on touch surfaces started with vibrotactile actuators attached to PDAs [84, 176, 210], then mobile phones [38, 121]. Current mobile phones use low quality actuators (ERM or LRA). Vibrations are still mostly used for messages and call notifications. Mobile phones sometimes use vibrations for button presses, but the quality of this feedback is poor. Indeed other actuators provide sharper vibrotactile feedback. We discussed this at the beginning of this chapter. Voice coil actuators provide precise and strong vibrations [266], and piezo actuators have a smaller form factor and are convenient for implementing buttons [157, 248]. Our challenge is that printed actuators are thin (between 4 and 10 µm). Therefore the vibration propagation requires a careful design.

Kim and Lee analyzed force-displacement curves of physical buttons and designed high-quality vibrotactile feedback that emulates button presses [138]. They split force-displacement curves into two types of sections: slopes and jumps, which are delimited by tactile points (Figure I.15). Slopes are spring-like sensations that correspond to material resistance. When pressing a button, we feel first the resistance before the click, then another one when the button reaches its end. When releasing a button, we feel another resistance before the click sensation. The tactile points (1) and (3) represent the click sensations when the button physically switches, and the bottom-out (2) represents the end of the button. We adapted and simplified these curves for the design and implementation of our buttons. We describe further our implementation of buttons, sliders, and touchpads in [**81**, **82**].

The design of the tactile feedback for slopes and points we described in the previous paragraph requires an iterative process. Several tools were designed to support the design of vibrotactile animations [229, 230], but they are not necessarily designed for the design of haptic feedback for tactile widgets. As we discussed in section I.2.3.1, the haptic drivers provide an audio mode in which we can provide an audio signal to drive the actuators. We leveraged this feature with Purr Data, a web-based audio synthesis programming language that we used to design vibrotactile feedback [**83**]. We describe the design of the vibrotactile feedback for tactile buttons in [**82**].

---

[7] TI DRV2667

[8] Microchip CAP1188 for buttons and Microchip CAP1214 for sliders

[9] Raspberry Pi 3B

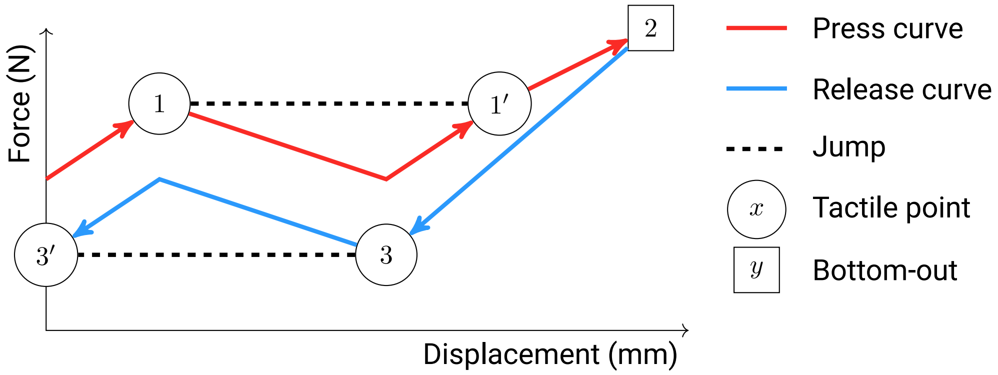


Figure I.15 – Force-displacement curve for a tactile button, adapted and simplified from [138].

2.3.3 Discussion

This work required interdisciplinary skills and knowledge to put together existing building blocks to design a whole system. This is what I consider a strength of Human-Computer Interaction research, at least the way I practice it. We have skills in many research domains, which allows us to design, implement and evaluate interactive systems. When more expertise is required, we can efficiently collaborate with experts in other domains. In this project, we collaborated with experts in material science who designed the tactile actuators, modeled and simulated their vibrations in a theoretical environment.

In these simulations, there was no finger, and the clamping of the surface was perfect, without tension. In practice, the finger dampens the vibration, and it is unsure other vibrations disrupt the perceptions of the vibrations produced by the device. In particular, the main scenario of the project[10] was a car dashboard. Cars produce vibrations of various frequencies and amplitudes because of the engine and the road. Other partners of the project actually evaluated the perception of this tactile feedback in a real-case scenario [178]. On our side, we noticed the clamping of research prototypes is not trivial. It is difficult to glue the flexible substrate on a rigid surface with a clamping area so that the substrate remains perfectly flat on the clamping area. It creates a bi-stable condition on the surface that produces an undesirable click sensation when touching it with a finger. Therefore we must apply tension to hold the flexible substrate flat on the rigid surface. It is well known that adjusting the tension of a drum head on a drum shell changes its resonant frequency. This is precisely the way we tune a drum. The same principle applies to our setup. However, the vibration was still perceptible, with no bi-stable condition.

A difficulty of this work was the long manufacturing process of actuators. It required weeks of planning, and the actuators were printed in a clean white room. This long process limited the number of iterations we could perform for designing the actuator layout and properties such as thickness, shapes, and sizes. Therefore at each iteration, we printed several configurations then tested them to select the most appropriate one. However, it limited the type of user studies we could perform.

---

[10] H2020 Happiness, grant agreement #645145

One of the major differences between this vibrotactile technology and other piezo-based actuators is that their thickness is very low. It is an advantage because it uses less materials. It is also a drawback because the amplitude of vibration is much lower. Therefore the vibration hardly transmits to thick surfaces like a 1mm thick plastic dashboard. The alternative is placing the actuators on holes that define a clamping area so that the substrate resonates like a drum shell. This solution brings an interesting property that other technologies do not have. The vibration is localized to the clamping area, therefore it is localized. Hence with a dashboard with several buttons and sliders, it is possible to vibrate buttons individually.

2.4 Actuated computer peripherals

The previous project addressed technological issues due to the replacement of physical interfaces by touch interfaces. In particular, we studied a haptic technology that could restore the haptic feedback of physical controls. This project is the exact opposite. We embrace physical controls and their haptic properties, and we study how we can use them differently. It introduces the idea we will develop in Chapter II and Chapter III that haptics is not only about the sense of touch, but also about manipulation.

This project was a collaboration with Gilles Bailly during his postdoc at Deutsch Telekom, and his debut as a CNRS researcher. In the first part of this project I designed and implemented the Métamorphe prototype, and we collaborated with my former colleagues at the University of Toronto for the user studies: Daniel Wigdor and Jonathan Deber. In the second part of the project we worked with Sylvain Malacria just before he joined the Mjolnir team, and Sidarth Sahdev, and electrical engineering master student who designed and implemented the Living Desktop hardware before he joined the Univerity of Toronto as a Ph.D. student.

In this work, we focused on desktop interaction with the augmentation of desktop peripherals. There are many examples of extensions of desktop peripherals in past research, in particular keyboards. For example, additional sensors enable contact sensing on the keys of a keyboard [213], gesture on the whole keyboard surface [31, 134, 249, 271], or force sensing on keys [69]. In other works, actuators are embedded in each key to make them harder to press [118, 226]. Mice were also augmented, with shape-changing features to enable notifications or provide an elastic input device for continuous rate control [137]. More generally, the idea of changing the shape of a physical controller is to provide different affordances [136, 168]. The work we have done with Métamorphe explores the augmentation of keyboards for increasing its input vocabulary [**12**]. We embedded solenoids in keys such that they can either be raised or lowered. However, they could be pressed in both positions. Not only it changes the geometry of the keyboard, but it also gives users access to the sides of the keys. We will discuss the augmentation of desktop interaction at the device level in the next section.

Beyond the augmentation of devices, the desktop itself can also be augmented. The earlier works on the extension of the desktop combined the best of paper and computing [6, 263]. Later, other systems combined projection and touch interaction to give users access to additional information and offer them a larger interactive surface [26, 241]. Actuation is another modality that augments desktop interaction. For example, a fleet of small robots can represent data dynamically [143]. However, the combination of ubiquitous displays with actuation brings another dimension that increases the interaction vocabulary [220]. Our approach with Living Desktop also considers the desktop workstation as a whole that should be integrated into its environment [**13**]. However, our primary focus is on augmenting devices and their interaction by leveraging their physical properties.

These two research projects are complementary, and they focus on different levels. Métamorphe focuses on the device level while Living Desktop focuses on the desktop level. In both cases, we use actuation as a mechanism to provide new features, with two paradigms in mind. The first one is shape-changing interaction: the shape of an object is one of the signifiers of its affordances. Hence changing the shape of a device is an interesting way of providing and advertising an extended interactive vocabulary. The second paradigm is tangible interaction, which "augments the real physical world by coupling digital information to everyday physical objects and environments." [128]. Here we see desktop peripherals as everyday objects that we manipulate as such.

2.4.1 Device level

Motion is an essential aspect of interaction with peripherals. Pointing devices rely on movement measurements. Keyboards use binary key positions as input data. In the Métamorphe project [**12**], we actuated the keys so that they can either be up or down (Figure I.16, left). However, contrary to other augmented keyboards [118, 226], users can press its keys, whether the key is up or down.

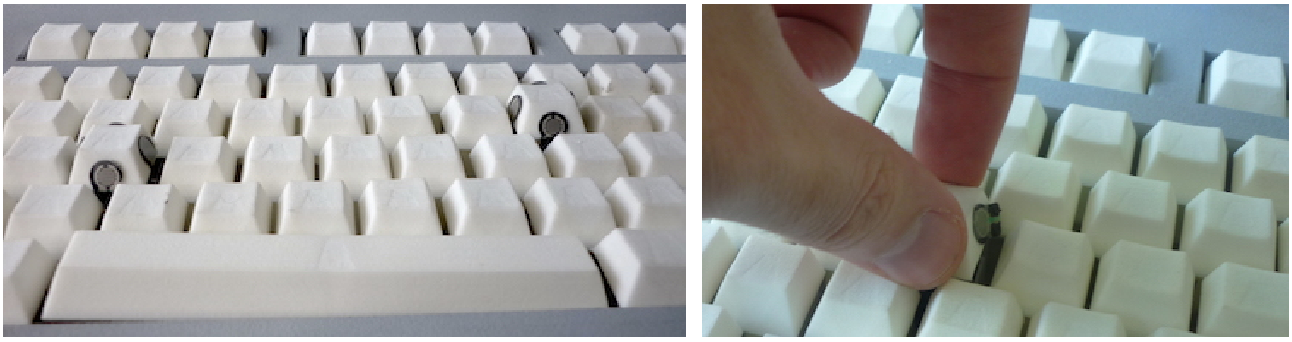


Figure I.16 – Métamorphe is a keyboard with actuated keys, which can either be up or down. Left: view of the keyboard with two keys up. Right: raised keys have new interaction possibilities. For example, they can be pushed or pinched.

This shape-changing keyboard provides new interactive properties compared to regular keyboards. First of all, it changes the haptic properties of the keyboard. When users scan the keyboard with their hands, they can distinguish raised keys. Not only it helps users to locate raised keys, but also the surrounding keys because the raised keys provide a reference point. This is the same phenomenon as the little notches on the `F` and `J` keys that help touch typists placing their hand on the keyboard, at a different scale, and with more flexibility. We experimented this phenomenon with a user study [**12**]. It is indeed an interesting property for eyes-free interaction. We can imagine for example that keys corresponding to a keyboard shortcut are raised when the `Ctrl` key is pressed. This was the original idea behind the CtrlMouse project [**199**] that I will not discuss in this document.

Beyond haptic properties, this new mechanism provides other benefits. When a key is up, users can push it in four directions, pinch it (Figure I.16, right), or even rotate it. With a force sensor all around it, we can turn the key into an isometric pointing device such as a trackpoint. Previous work showed examples of interactions we can perform with an array of actuated rods [80, 130, 149]. Our prototype only actuated eight keys for technical reasons, and we kept the layout of traditional keyboards because we intended to augment traditional keyboards. However, if we put technical limitations aside, we can envision a combination of these two concepts: a shape-changing surface, and a new input vocabulary brought by controllers that pop out of the surface.

2.4.2 *Desktop level*

In this project, we observed people when they use a desktop computer. We identified situations in which they move the peripherals but not for interacting with the computer. For example, we observed people turning their screens to avoid sun reflections. Other users turned their screen either to show visual content to somebody. It is also frequent to give other people the mouse or keyboard to give them control over the computer. In the Living Desktop project, we actuated a mouse, a keyboard, and a screen (Figure I.17). The mouse and the keyboard can translate in the x, y plane directions. The keyboard can also rotate. The screen can rotate and translate in the x axis direction. The details of the apparatus are described in [**13**].

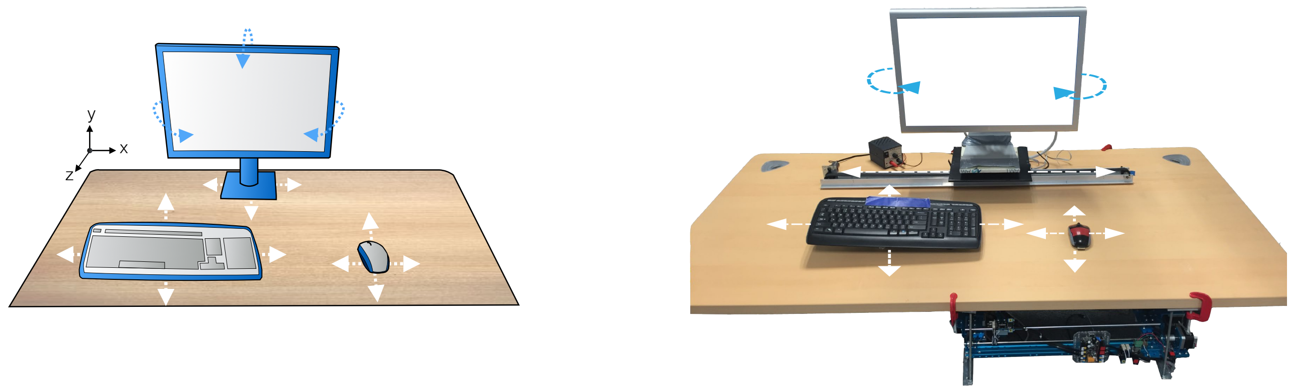


Figure I.17 – The Living Desktop is a concept in which desktop peripherals can move around on the desk.

With these capabilities, devices can move on their own without requiring the user to move them. The interesting question here is the degree of control the user has over their devices. Beaudouin-Lafon defines two interaction paradigm: *computer-as-a-tool* and *computer-as-a-partner* [21]. In the first case, the system only reacts to the users' actions. In the second case, the users delegate tasks to the system. Here I argue for a continuum between full control and full automation. We discuss examples of application scenarios for four degrees of control.

Full control

When users have full control over the actuated devices, they can move them around physically or remotely. For example, when video-conferencing with a desktop computer, the camera is usually affixed to the screen. We can manipulate it with full control to adjust the field of view. The problem is when remote users would like to show an object they manipulate outside the camera field of view. They have to move the screen at the same time that they are manipulating the object. In this scenario, we take the control over the remote screen to adjust the field of view and make sure we can see what the remote users would like to show.

Constraint control

Even with a large screen or multiple screens, the interactive screen estate is limited. We propose to use an actuated screen as a peephole display in a larger working space. In this scenario, the screen moves on the x axis and the pixels show the content in this area in the physical space. The screen is like a moving physical window. We can imagine combining this scenario with a projection that provides a low-resolution image around the screen [132].

Constraint automation

We sometimes need to watch information on our screen while moving in a room, like when working on a whiteboard. We implemented a scenario in which the monitor orientation follows users in their office It displays notifications such as new emails, agenda alerts, missed calls, etc. It also uses proxemic interaction [222] by adapting the text size so that it is readable regardless of the distance.

Full automation

Being well seated is essential for healthy office work. It reduces fatigue and pain. It is however difficult to pay attention to our posture all day. In this scenario, the mouse and the keyboard move away if we are not seated correctly on the chair. The user has no control over the devices in this situation.

2.4.3 Discussion

This work on actuated devices is indeed different from projects discussed in the previous sections. The previous projects followed the typical view on haptic, which we commonly designate as “haptic feedback”. In these previous projects, we encoded information or rendered haptic properties of objects with forces and vibrations. Here we take a step back and consider the haptic properties of physical objects. The properties are due to the objects’ shape, size, weight, material, position, etc. By actuating objects, desktop peripherals in our case, we modified some of these properties, typically the shape and position. Our focus here was not on how users perceive these haptic properties but how we can leverage them to propose new interaction properties.

These projects introduce two concepts that we will develop in Chapter II and Chapter III. The first one is the idea that haptics does not only cover the sense of touch but also our ability to manipulate. In our examples, the haptic properties of objects enabled different kinds of manipulation. The situation is reversed, and we can see the user as a haptic device that produces forces on a manipulated entity. The second concept is the idea that we cannot separate haptic as output and haptic as input. These are two sides of the same coin that forms interaction. This concept has many consequences, such as the continuum between control and automation, and it is linked to several fundamental paradigms of the literature that we will discuss in Chapter III.

3 Conclusion

The work that was presented in this chapter illustrates the systematic approach in my research. The objective is not to focus on a particular technology, a particular problem, or a particular context. Rather, I search for appropriate technologies for a given problem in a given context. It gives me the flexibility to address many different types of research questions depending on the expertise of my collaborators and the objectives of the research project. This is to me the key factor to conduct interdisciplinary research. For example in the *Activibe* project, I collaborated with other researchers in HCI who were interested in using haptic cues for promoting behavior change. The focus was not on designing a new device or even new haptic technologies, but to design a way to encode information with an off-the-shelf device. Hence, we worked on the design, implementation, and evaluation of tactons. The approach was different with *tactile textures* and *printgets*. In these cases, I collaborated with researchers with a background in engineering who designed a new kind of haptic technology. The two situations were different though. In the first case, I also worked with a postdoc who had a background in cognitive sciences. Therefore our studies focused on the perception and interpretation of tactile textures. In the second case, I worked with a postdoc who had a background in audio and music technologies. As a consequence, our work was directed to the signal generation, authoring tools, and the technical apparatus. Finally, with the *actuated devices* project I mainly worked with HCI colleagues and a master intern with an electrical engineering background. The intern designed and implemented most of the robotics part of Living Desktop. I worked mostly on the hardware part of Métamorphe, the software part of both, and the observation study All of us worked on user studies and the application scenarios. The complementarity of expertise in my research project is an important aspect that not only guides my own contribution, but also the whole direction of the project.

These research projects have a few limitations. Acknowledging them is not only important regarding these projects, but this is also useful hindsight for the evolution of my research practices on the same type of projects. While the longitudinal study we performed with *Activibe* is rare and insightful on this kind of topic, there is room for improvement on the tacton design methodology. The design rationale was not systematic, so even if we found a satisfactory solution there are probably many other possible designs. The difficulty is that the design space is huge, and there are many points of view. The main consequence of the results is the difficulty to generalize these results to bigger sets. The design of large tactons sets is hard in general. The works of *tactile textures* and *printgets* were interesting examples of projects in which the research on the technology and the research on the usage of the technology feed each other. They are illustrations of Huot's *designeering* concept [126]. One of the limitations is that research prototypes are built upon simulations. However, real-world conditions can have a huge impact on the device performance, and the user's perception. This is why an iterative process is efficient at taking into account both the technological and interaction concerns. However, the other limitation is the slow iteration process due to the long manufacturing of prototypes. HCI uses low-fidelity prototypes to bootstrap the first cycles of iterative design. It is important in such projects to identify alternative technologies that can be used for prototyping. This way, as HCI researchers, we can give our engineering colleagues specifications at the early stage of the project rather than waiting for their early prototypes that already commit to choices that cannot be changed. In the *actuated devices* project, we built working prototypes that enabled us to explore the scenarios in real contexts. The Métamorphe was robust enough to evaluate interactive properties, such as the users' ability to locate raised keys. However, technical limitations prevented us from actuating all keys, or add all the sensors we would need to explore some of the new input we imagined, typically rotating the keys. The Living Desktop was functional but with technical limitations on speed or forces. Therefore a user evaluation with the prototype could have been biased due to these technical limitations. This is why we evaluated the scenarios with videos. We would certainly have collected more valuable data with more robust prototypes.

In this part of my research activities, I leveraged the sense of touch for the design of interactive systems. Most of this work focused on tactile sensations, but I remained interested in kinesthetic sensations as well. The design of force feedback devices is complex due to both the mechanical constraints and signal generation of the device side, and the perception and interpretation of the mechanical effect on the user side (see Figure I.1). My next projects in this area will cover an evaluation of the effects of the haptic loop frequency on the perception of haptic feedback. It will require the design and implementation of a simple but fast force feedback device.

This chapter focused on haptics as the sense of touch, which is the usual point of view on haptics. However, as we have seen with the actuated devices in particular, haptic is also about manipulation. This is the subject of the next chapter.

II The motor ability

The true delight is in the finding out rather than in the knowing.

Isaac Asimov

After considering haptics as the sense of touch, we discuss here haptics as the human ability to touch and manipulate the environments and the objects it contains. We present the motor sensing pipeline that is the mirror of the haptic rendering pipeline discussed in the previous chapter. It reveals the research questions I addressed in my research: the sensing and interpretation of the users' gestures, the input vocabulary, and the design of interaction techniques for unnatural actions. Then I discuss four contributions: a system latency measurement methodology and tool, flexion as a new degree of freedom for pen interaction, finger identification as a new property of multi-touch interaction, and interaction techniques in virtual reality.

In the previous chapter, we discussed haptics as the sense of touch. This is the common interpretation of the word "haptic". However, haptics also refers to our ability to touch and manipulate. In this chapter, we will focus on gestures we perform with our fingers, hands, and arms. It covers most of the commonly used input modalities: button, pointing interfaces, touch, and gestural interaction. All these modalities rely on gestures, but they leverage different parameters. Typically, buttons only sense binary contacts regardless of which finger pressed it or the force or speed of actuation. Pointing interfaces such as mice or touchpads sense 2D movements in addition to contacts. Multi-touch interfaces can even sense the contacts and movements of several contact points. Finally, gestural interaction can be sensed in 3D in the air without contact.

The relation between gesture input in the broad sense and haptics may seem far-fetched at first sight. However, if we look at Figure I.1 in the previous chapter and switch the user and the system, we obtain the Figure II.1. The user produces a mechanical effect that the system will sense and interpret. Therefore the user will play the same role as a haptic device. With this in mind, this is not surprising that cognitive scientists call this phenomenon output, as opposed to input for computer scientists. The Figure II.1 depicts both the user and the system, which both have a hardware part, in the physical world, and a software part. The software part of users corresponds to ideas, or the mind in general, whereas the software part of the system refers to the code and its execution.

The purpose of the modalities mentioned above is to sense and interpret the gestures of the fingers, hands, and arms. This is a much more complex task than it seems at first sight. The hand alone has 21 degrees of freedom [72] and the arm adds 7 more [175]. The movement range of each of these degrees of freedom depends on multiple factors, including morphology, physical condition, age, and gender [175]. It is therefore not surprising that input systems only sense a small part of the possible human movements.

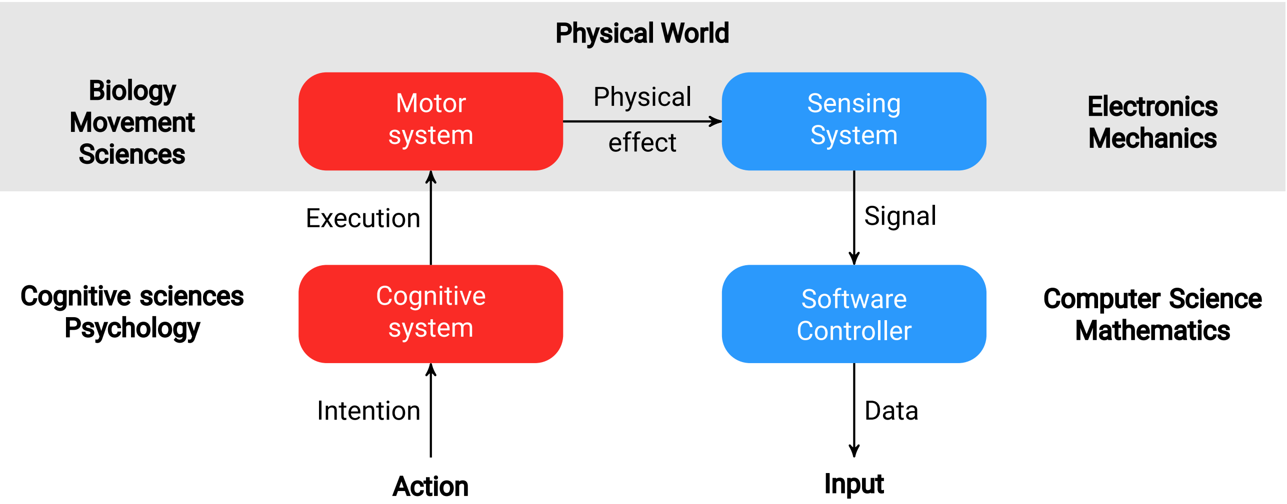


Figure II.1 – Motor sensing pipeline with the user side and system side from the user's action to the inputs in the system. Both have a hardware and a software aspect.

Motor ability

Definition 1 Human *abilities* refer to the human capacities to act on their environment, similar to the human *senses* that refer to the human capacities to get information from their environment.

In this chapter, we focus on the *motor ability*, which leverages the *motor system* to touch and manipulate the environment and objects it contains. As depicted on Figure II.1, this notion comprises both the motor system and the associated part of the cognitive system. Specifically, when users should like to perform an action they form an intention that the cognitive system will turn into execution commands for the motor system. The motor system, typically muscles, tendons, and articulations actuate the body, which in turn produces physical effects. These physical effects enable the manipulation of objects in contact with the mobile parts of the body. They take the form of movements or forces. The hand has 21 degrees of freedom: 5 for the thumb and 4 for each of the four other fingers [72]. The arms have 7 degrees of freedom: 3 for the shoulders, 1 for the elbow, 1 for the forearm, and 2 for the wrist [175]. Knowledge about the range of movements and maximum forces is necessary for the design and layout of workstations. For example, NASA documents these values with and without gravity, or with pressurization because they need to provide precise and documented specifications for spacecrafts [175]. They also use these specifications to design clothes that astronauts can wear comfortably to perform routine tasks.

In our case, we design input systems. Therefore we need to know the range and precision of movements we have to sense. However, these are not the only human factors that have an impact on input systems. Users may ignore the way to perform a specific action. The discoverability [186, 207] and learnability [58, 106] of interaction are typical research problems in the HCI field related to this point. Then the physical actions users have to perform can require high motor skills or training [228]. For example, performing some multiple-key keyboard shortcuts with one hand can be challenging [**199**].

Input systems

Input systems capture our movements with sensors that measure directly or indirectly parts of our movements. There are several families of technologies for this. The list below is not necessarily meant to be exhaustive. Rather, the idea is to give readers an overview of today's most frequent technologies. The first family is electro-mechanical components. In this family, various types of switches and push buttons sense contacts. Keyboards and buttons use variations of this technology. Linear and rotary potentiometers sense continuous movements. Joysticks typically contain two of them to sense rotations in two directions. Encoders sense discrete movements. Ball mice used two of them to sense positions, and mice wheels still use one. The second family of input sensing technologies is electrical sensors, such as resistive or capacitive sensors. They sense the position of one or more contact points. They are used in current touchscreens, including mobile phones and tablets. The third family is vision-based technologies. Such sensors use cameras and vision algorithms to detect movements. For example, optical mice use high-frequency and low-resolution cameras. Old-generations tabletops used infrared lights and cameras to detect contact points. Other devices use one or more RGB or infrared cameras to detect markers or a projected pattern. They are used for body motion capture or for tracking objects. VR headsets also use this technology to track the controllers. The last family of technologies is Microelectromechanical systems (*MEMS*). They sense many kinds of physical phenomenons such as acceleration, rotations, magnetic fields, or fluid pressure. Input systems typically use combinations of accelerometers, gyroscopes, and magnetometers called Inertial measurement units (*IMU*).

The signal coming from these sensors requires several transformations. Contact inputs such as buttons require a software or hardware *debouncing* mechanism to avoid unwanted multiple activations. Threshold-based input such as capacitive sensing not only requires adjusting a sensitivity and threshold value, but they often require an *hysteresis* mechanism to avoid multiple activations as well. Analog signals must be transformed to digital values with an Analog-to-digital converter (*ADC*). Input values often have noise that must be *filtered*. Many possible filters remove noise, at the cost of latency [53]. Some kinds of input require further transformation. In particular, pointing input requires a *transfer functions* that computes the movement of the cursor on the screen depending on the physical movements of the input device. These transfer functions usually take into account the ballistic-then-corrective nature of our movements [167]. Vision-based technologies are sensible to occlusions. Therefore the software part of the pipeline extrapolates data to fill gaps in the input streams. The combination of several sources of inputs is challenging as well but provides more precision in some cases. For example, data from accelerometers require mathematical integrations for position sensing. Not only it requires calibration, but it is also sensible to drift due to the data precision. The fusion of accelerometers, gyroscopes, and magnetometers provides better tracking, at the cost of increased processing complexity.

The software processes described in the previous paragraph are either computed on the device or the host computer. For example, the device systematically debounces inputs with analog low-pass filters. They also implement hysteresis effects with a Schmitt trigger[1]. The ADCs are either microcontrollers peripherals or dedicated components. Filters are commonly implemented either on the device firmware or the host drivers. For example, mouse and touchpad movements are filtered on the device. On the opposite, inputs of depth-cameras are filtered on the host side, because the host retrieves raw data and computes a skeleton for example [233]. The transfer function is typically computed on the host because it requires information about the display.

When devices are integrated into interactive systems, they are connected to hosts with a simple bus like SPI or I2C[2]. In this case, the device implements a communication protocol that the host has to follow. There is no standard protocol, but the overall idea is usually similar. These buses use no correction codes, therefore they are fast but sensible to interferences. Thus, devices that users can plug

---

[1] https://en.wikipedia.org/wiki/Schmitt_trigger

[2] https://en.wikipedia.org/wiki/Serial_Peripheral_Interface https://en.wikipedia.org/wiki/I^{2}C

use more robust buses such as USB with the Human Interface Devices (*HID*) class[3]. This class defines a standard communication protocol for interactive devices. When the device is plugged, it sends descriptors that list its features. In particular, the HID descriptor details the format and semantic of the data packets the device will send at a fixed frequency. With this protocol, the host can interpret virtually any HID device with a generic driver. Regardless of the communication method between the host and the device, the drivers of the operating system create *input events* that applications will interpret for their own use.

1 Research questions

Similar to the haptic pipeline, the motor pipeline reveals pitfalls that could lead systems to behave differently than what users had in mind. Users may ignore the way to perform what they intend to do. They can know the action they have to do but it is challenging to perform. There can be obstacles in the physical world that prevent systems to sense these actions correctly. The range of physical effects can be out of the sensing range of the system. The system may interpret what it sensed incorrectly. This pipeline is therefore a profuse source of HCI research questions. I focus here on three categories of research questions that I addressed in my research in the last decade: sensing and interpretation, input vocabulary, and unnatural input.

1.1 Sensing and interpretation

Interactive systems have a limited set of sensors, which sense a limited subset of the users' actions features. This is sometimes a desired property: a button is pressed regardless of the actual motion that produced its activation. When users lift a mouse, its position is not tracked anymore. This behavior enables *clutching*, which extends the motion range of the mouse cursor with the same required physical area. Sensing limitations are sometimes constraints for the interaction though. For example, occlusion phenomenons or the limited field of view of vision-based sensors can impede gesture recognition. Many factors can have a negative influence on the processing of input signals. For example, gesture recognition failures provoke interaction errors. Beyond errors, the *segmentation* of gestures is a key challenge of gestural interaction because typical gesture sensors observe users all the time, even when users do not want to interact with the system. The main consequence is the *Midas touch*[4] effect that forces users to interact at all times. In the next chapter, section III.2.1.1, I will discuss this issue and a mitigation strategy for this specific problem. In the section II.2.1 of this chapter, I will discuss a method and measurement system we designed for characterizing the latency of input systems.

---

[3] https://www.usb.org/hid

[4] https://en.wikipedia.org/wiki/Midas#Golden_Touch

1.2 Input vocabulary

The structure of the input vocabulary is similar to the structure of the output vocabulary. We describe it with the lexical, syntactic, and semantic levels of languages. I typically explain this with the example of the computer mouse because everybody used it and knows its interactive language. Its lexical elements are *degrees of freedom* (DOF): x-y motion and button presses and releases. They have associated data: relative displacement value on both axes, and the ID of the button that was pressed or released. The syntactic level consists in assembling these elements to form input phrases. A click is a button press followed by the release of the same button. There are many variations, typically a double click that repeats a click twice, or half-clicks that replace the release by dwell. Finally, the semantic level associates input phrases to commands and parameters. For example, a press on an object followed by a movement and a release on another object is typically interpreted as moving the first object into the second one.

When designing a new input device, we have to specify its DOFs, their type, range, precision, and if they are integrated with each other [160]. For example, the x and y DOFs of a mouse are integrated because users control them together with a single movement. The design process of an input device consists of cycles between engineering and evaluation in order to choose and document this information related to DOFs. In Section II.2.2 I will describe the design of flexible pens, and in Section II.2.3 I will explain how we leveraged finger identification for multi-touch interaction.

1.3 Unnatural input

The term *natural interaction* was already used before multitouch and gestural interaction became mainstream [33]. These interaction styles are integrated into ubiquitous consumer electronics products such as smartphones and tablets for more than a decade. They are nowadays referred to as natural user interfaces or NUI [264]. The first question is: what do we mean by “natural”? It primarily refers to what exists in the nature. However, gestures are not objects that can be found in the wild. It also refers to what comes to mind, or prior skills. People have skills in manipulating physical objects that we can leverage in interactive systems. Yet, people have different backgrounds, skills, and cultures. Therefore what comes to somebody’s mind is not necessarily the same as what somebody else is thinking. The second question is: is it better because it is natural? This is a vast question because it depends on what we would like to improve: performance, learnability, guessability, etc. There is no general guarantee that gestures improve any of these measures. They have to be carefully designed for this. This is the reason why decades have passed between the first gestural interaction systems and the first successful commercial products based on gestural interaction. Norman discussed the concept of NUIs and concluded that what matters is not whether these interactions are natural or not [184]. What matters is that it is an alternative interaction modality for the design of interactive systems, and the same usability rules apply to them.

Now, if the natural essence of gestural interaction is not an essential benefit for the design of interactive systems, we can think about this modality differently. The digital world does not have some of the limitations of the physical world [131]. For example, we can easily teleport inside a virtual environment. We can move through objects or even fly. We can manipulate objects remotely and independently to their weight, size, or shape. Hence, when designing interactive systems with gestural interaction there is no necessity to reproduce the physical world. We should rather focus on what we would like users to achieve, and improve performance, learnability, and so on. From a practical point of view, we care about the input vocabulary, the properties we would like to manipulate in the virtual environment, and the mapping between them. For example, in Section II.2.4 we present two interaction techniques for immersive virtual environments. The first one is a selection technique for distant objects. The second one enables users to select facial expressions for their avatar without the need to perform the same expression with their own face.

2 Contributions

With these research questions in mind, now I will describe some of my contributions leveraging the users' motor abilities for the design of interactive systems and interaction techniques. Instead of presenting the research below the way it was described in research papers, I will focus on hindsight and how these contributions influenced my research.

2.1 System latency measurement

The first contribution I will present is not an interaction technique or an interactive system. It is a methodology and study about the latency of touch-based interactive systems in the general sense. Every interactive system has a delay between the moment users perform an action and when the system produces a response. This delay is referred to as *end-to-end latency*. This latency is known to cause performance and usability issues [65, 133, 250, 258]. Therefore, there are research studies about strategies to mitigate these effects or reduce latency [54, 174].

This work was part of the ANR Turbotouch, coordinated by Géry Casiez. He was involved in this work, along with Nicolas Roussel, and Mathieu Falce who was an engineer in the team. I designed and implemented the Lagmeter device with the help of Damien Marchal, who is a permanent CNRS research engineer at the CRIStAL laboratory.

Before reducing latency or mitigating its effects, it is important to measure it and understand the contribution of each part of the system to it. There are many possible sources of latency: input and output, software and hardware, device and host. Typical latency measurement methods consisted in counting frames on videos made with a high-speed camera [177, 240, 250]. This method is simple but tedious. It is difficult to make large series of measurements and it is impossible to identify the main sources of latency. Other methods enable repetitive measures, but they are adapted to specific input types [49, 64].

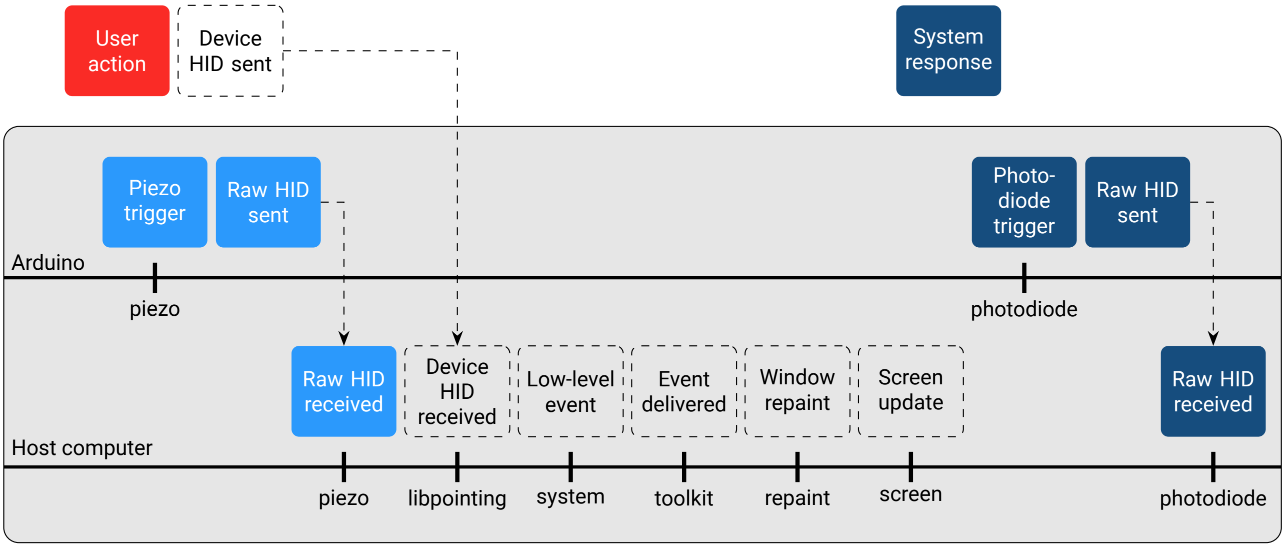


Figure II.2 – LagMeter measurement process.

2.1.1 Measurement apparatus

We designed LagMeter, a latency measurement tool that facilitates repetitive measures, and enables the slicing of latency between several parts of the interactive system [**50**]. It measures latency for any tap-based input such as keyboards, mice, touchpads, touchscreens. The details are described in the paper, and the overall idea is depicted on Figure II.2. The light blue items correspond to input stages and dark blue items to output stages. The dashed items are optional steps that provide finer grain information for latency slicing. They depend on the system running on the host computer.

The moment users touch the input device, a piezoelectric vibration sensor[5] attached to the fingertip detects the contact. The idea is that all touch-based inputs require an initial contact. We connected this sensor to a custom electronic board connected to an Arduino Leonardo[6] microcontroller board. The board enables the adjustment of the detection threshold since the voltage we get from the piezo sensor is typically lower than 0.5 V. The Arduino sends a Raw HID message to the host computer when it detects contacts. In parallel, the input device sends an event HID packet to the host computer. A custom application gets the RawHID packet with HIDAPI[7], and the device HID packet with libpointing [51]. It also has callbacks on the system and the toolkit input events. When the application received the input event, the application toggles the screen between black and white. It has callbacks that trigger when the screen repaint is requested and done. Then, a photodiode on the electronic board detects changes on the screen and the Arduino sends another Raw HID message to the application.

---

[5] http://www.te.com/usa-en/product-CAT- PFS0011.html
[6] https://www.arduino.cc/en/Main/Arduino_BoardLeonardo
[7] https://github.com/signal11/hidapi

2.1.2 Results and discussion

What I realized with this project is that nothing in interaction can be considered instantaneous. When we measured the responsiveness of the piezoelectric vibration sensor, we noticed that even the mechanical contact of a physical button creates latency at the millisecond scale. It is both due to the elasticity of the finger, and the mechanical actuation of the button switch. We made this measurement with an aluminum foil around a finger, tapping on a copper tape pasted on a mouse button. The finger also had a piezoelectric sensor strapped around its tip. The Figure II.3 shows a measurement with three signals: the finger contact in blue, the button press in red, and the piezo signal in green. We notice it takes about 2 ms for the button to switch. The delay of the piezo trigger depends on the threshold, but it typically takes between 1 ms and 3 ms. The paper contains more detailed data about these measures [**50**].

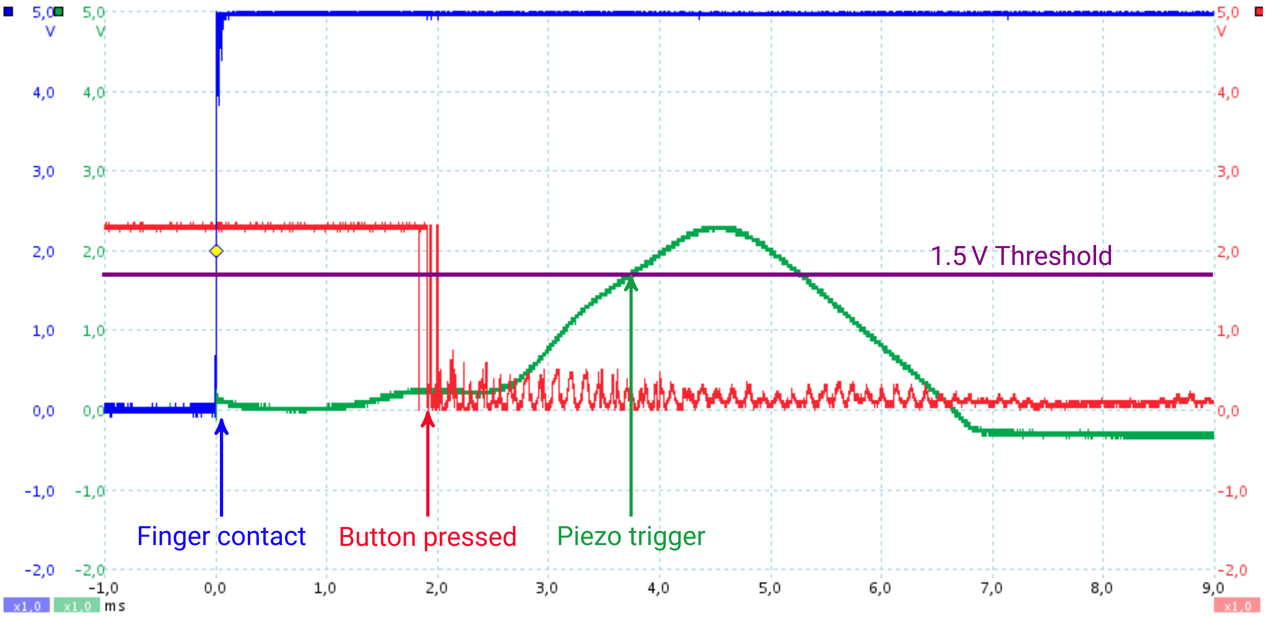


Figure II.3 – Latency measurement of a button press. The blue line goes to 5V when the finger touches the surface. The red lines goes to 0V when the button is pressed. The green curves shows the piezoelectric vibratio sensor signal. The violet line shows a $1.5V$ threshold.

The measurement of the screen change was more accurate. The idea was to switch the screen between black and white so that we can detect the change with a simple luminosity sensor. We used a photodiode because it is fast. Figure II.4 shows the signal when the screen turned white. Interestingly, the measure is so fast that we can observe the lines being drawn around the sensor. Therefore the position of the photodiode on the screen has an impact on the latency we measure. However, depending on the adjustment of the detection threshold, we can reliably measure the screen change in 4 µs. This is way sufficient for our purpose and makes the actual position of the sensor less critical.

Latency slicing requires having all measures with the same clock. This was an issue because part of the process was done on the Arduino board, and another part on the host computer. It means we needed a way to communicate in a fast and reliable way between the two. The problem is that

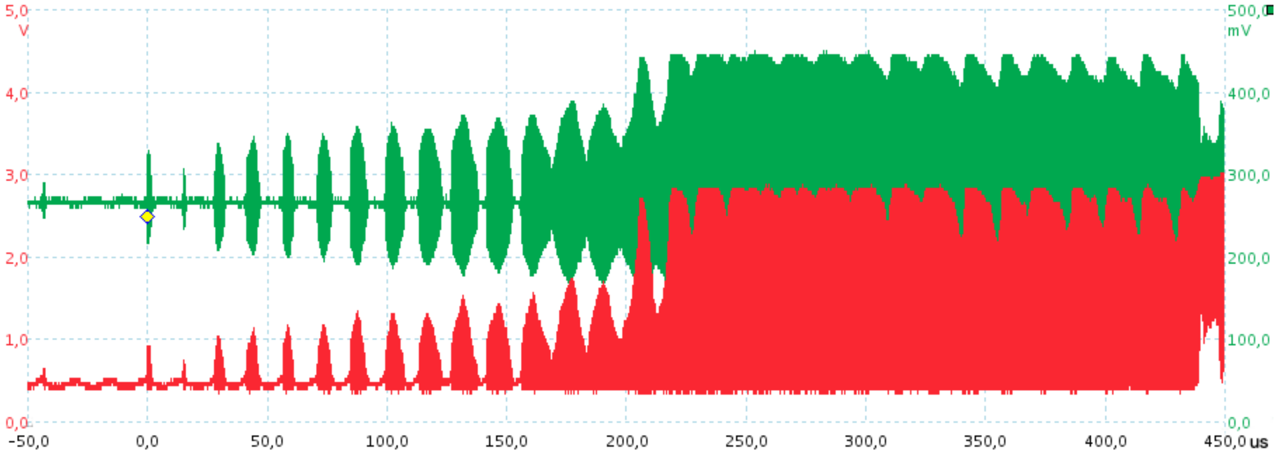


Figure II.4 – Photodiode measure. The green signal is the photodiode signal and the red signal is the amplified signal.

typical operating systems (Windows, Linux, and MacOS) cannot guarantee the execution of code with a 1 ms precision. One solution was to use a Raspberry Pi[8] as a host computer because it has GPIOs that can communicate with microcontrollers with low latency. However, measures with this platform would not necessarily be representative of the usage of other platforms. Thus we considered several communication channels between a microcontroller and a host computer, such as ethernet or a parallel port with a specific extension card. Nevertheless, we opted for raw HID messages because their latency was sufficiently low and consistent. The round-trip delay between the Arduino and the host computer was between 1 s and 2 s, depending on the operating system and the processor load.

We made multiple series of measures to study the impact of many types of factors. Concerning the input, we compared different input frequencies with a Logitech G9 mouse that enables the adjustment of the mouse frequency. The host computer received the device HID packet after 3.3 ms on average for 125 Hz and under a millisecond for frequencies above 250 Hz. The latency between the moment the device HID packet was received and the moment the application requested the repaint of the screen was about 3 ms. The end-to-end latency was around 60 ms, therefore the input part of the pipeline has little impact on latency. Most of the latency is due to the output. We observed several factors that impact latency: the operating system, graphic toolkit, and screen frequency. We used a minimal application for our measures, but real applications most likely introduce more latency because of their normal operations.

---

[8] https://www.raspberrypi.org/

2.2 Flexible pens

Latency is a technical limitation for interaction. On the oposite, limited output vocabularies can be improved with technical solutions. Pen interaction is a good example of the exploration of new degrees of freedom. This is certainly because pens are used in many contexts, in particular for artistic creation. Typical interactive pens sense the x-y position as well as proximity. Research explored additional sensing such as pressure [116], tilt [251] and roll [27]. Not only this extended input vocabulary can be used to map brush parameters. But it also enables selecting commands and offers richer interactions, whether it is with combinations of pen and touch interactions [117] or by leveraging physical attributes of the pen [256]. In this work, we were interested in the bending of a flexible pen as additional degrees of freedom.

This project initiated my collaborations with Audrey Girouard from Carleton Univerity. The first part of this work was done by Nicholas Fellion who was a master student at Carleton. He did a 6 months internship in Lille to work on the Flexstylus prototype. The second part of this work was done by Alfrancis Guerrero, who was also a master student at Carleton University. He designed and implemented the Hyperbrush prototypes.

2.2.1 Prototypes

We built two series of prototypes, the first one being *FlexStylus* [77, **78**] (Figure II.5, up). It used a custom flexion sensor made with 4 eroded fiber optics, with an infrared LED on one end and phototransistors on the other end each. By measuring the amount of light sensed by the phototransistors we could infer two angles and a degree of absolute bending.

The choice of the flexible part did not follow a systematic rationale, but rather general design considerations. The idea was to have a diameter similar to the one of a drawing pen. It had to be flexible enough to avoid muscle strain, but stiff enough so that users could write and draw conveniently. Informal pilot studies showed that full-length flexible prototypes were inconvenient for precise manipulation. Therefore we chose to limit the flexible part to a few centimeters with a rigid part on both sides to keep the benefits of both flexible and rigid pens. The rigid part between the tip and the flexible part is long enough to enable users to grip the stylus there.

The second series of prototypes, called *HyperBrush* [**107**] (Figure II.5, down), used a consumer electronics bend sensor[9]. It provided more precise and reliable inputs. However, the interesting new property of these prototypes was the ability to change the flexural stiffness with interchangeable flexible components. Both the rigid and flexible parts of the pen are 3D printed, and the flexible part is threaded so that it is screwed to the rigid parts. The end sensor slips inside the 3D-printed stylus.

The FlexStylus prototypes had an orientation issue. Users had no cues on where to hold the stylus. Therefore the angles inputs were relative rather than absolute. This is the same issue Buxton reports with the iMac Round Mouse[10]. We addressed this issue with the HyperBrush prototypes by adding

---

[9] https://www.bendlabs.com/

[10] https://www.microsoft.com/buxtoncollection/detail.aspx?id=109

Figure II.5 – Two series of flexible pen prototypes. On the top: FlexStylus with a custom bend sensor made of eroded fiber optics. On the bottom: HyperBrush with a consumer electronics bend sensor, and interchangeable flexible components.

a fake button that users were encouraged to keep under their index fingers. The relief of this button helped users keep the stylus in a consistent orientation eyes-free.

2.2.2 Pen grips

Studies in the literature show that there is a strong connection between the way users hold a stylus and the way they use the stylus [116, 239]. They noticed that participants in their user studies often changed the way they gripped the stylus depending on the task they were performing. We took inspiration from this work, and we designed our prototypes for three kinds of grips.

- **Tool grip**: users hold the stylus the same way they hold a pen. This is actually a family of grips, but the idea is that users hold the rigid part of the pen close to the tip. They can press the pen with their thumb to bend it.
- **Menu grip**: users hold the stylus from the rigid part on the other side of the flexible part. The tip remains still on the surface thanks to friction. This way, users can control an orientation vector around the contact point.
- **In-air grip**: users hold the stylus in the air between their fingers. They can bend the stylus by squeezing their fingers, or rolling the pen.

When we evaluated our second series of prototypes some participants reported drawing with the menu grip, after the flexible part. Drawing this way gave them the impression of painting with a brush, hence the name *Hyperbrush*. This is a difficulty with this type of work. We need a design rationale to guide the implementation before we evaluate the device. We evaluate the interaction with the device to make sure that users can use it the way we wanted them to use it. However, we want to let such serendipitous behavior happen because it makes interaction with the device richer.

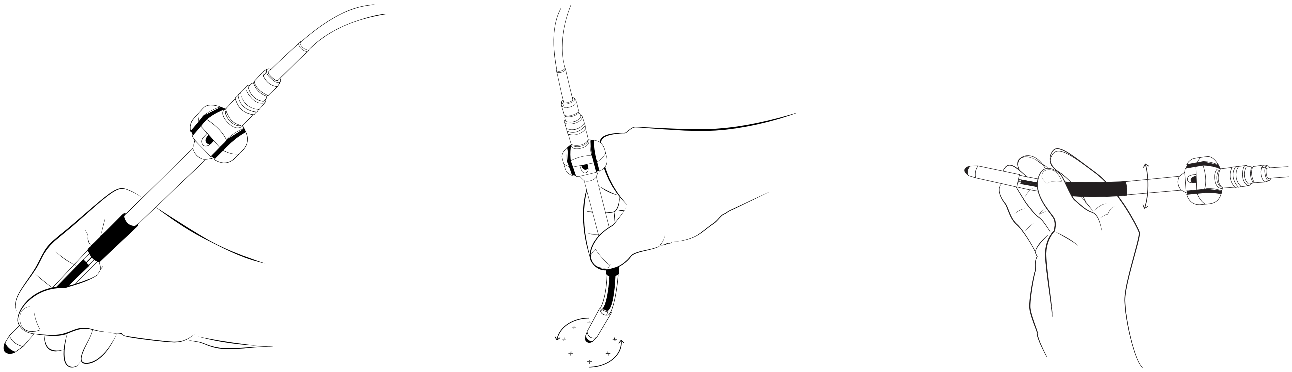


Figure II.6 – Three types of grips with a flexible pen. The pen grip, the joystick grip and the roll grip.

2.2.3 Evaluations and discussion

We evaluated the users' ability to control the bending of FlexStylus and compared it to pressure input [**78**]. We found out that the participants of our studies were more precise with the flexible pen than with pressure input. Pressure pens are isometric devices, therefore it is unsurprising that they do not perform well with position control [269, 270]. Flexstylus can be considered as an elastic device. Therefore position control can be an issue. However, the classification of a flexible pen between an isotonic or elastic device essentially depends on its flexural stiffness. A soft flexible pen is most likely to behave like an isotonic device while a hard one should act like an elastic or even isometric device. Therefore with the next generation of flexible styluses, HyperBrush, we wanted to compare different flexural stiffnesses.

We built several prototypes of different flexural stiffness (Figure II.7). The flexural stiffness is adjusted with the thickness of the flexible parts inner walls. The measure of flexural stiffness we used is the ratio of the amount of force applied per unit of deflection. We measured it with a hollowed cylinder cantilever beam test. We compared the performance and subjective preferences between these different flexural stiffnesses and a pressure pen for pie-menu item selection and brush stroke precision. The results do not show any clear better configuration. We also collected subjective preferences with a drawing task. The results suggest that the best is to provide users with several interactive pens that provide different kinds of input vocabularies and haptic feedback. Users will choose the one they feel is appropriate for their current task.

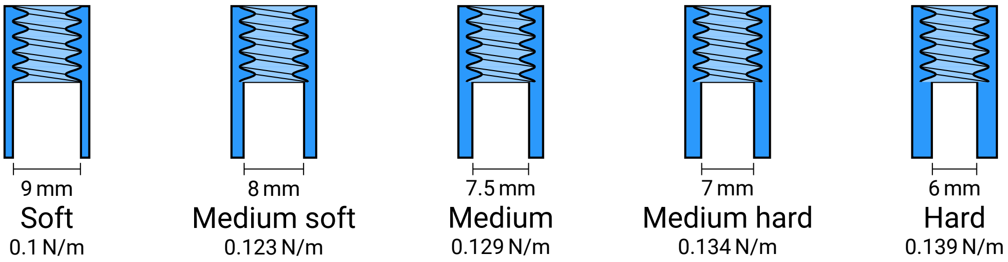


Figure II.7 – Flexural stiffnesses of the 5 prototypes of HyperBrush we used for our evaluations.

2.3 Finger identification

Hands and fingers have many degrees of freedom and most people have enough skills and dexterity to perform gestures and manipulate objects. Therefore hands are widely used for interacting with interactive systems. Yet, consumer electronics interactive devices only use a limited subset of gesture properties. Typically multi-touch devices essentially use the coordinates and movement of the contacts, number of contacts, and recently pressure. It already enables complex interaction techniques, in particular for command selection [11]. Further studies leverage the raw data of multi-touch surfaces, either from cameras or capacitive sensors. With this data, we can detect contact blobs instead of single coordinates, which increases the input vocabulary [221]. Other research investigated technologies that sense other gestures properties such as the part of the finger touching the surface [115], make a distinction between contacts from different users [70], or analyze the hand posture [173]. This has been an active research area in the last decades, therefore this is just a tiny overview of what has been done on this topic.

In this work, we were interested in *finger identification*. It was the master internship work of Alix Goguey, which I co-supervised with Géry Casiez. We collaborated with Fanny Chevalier and Nicolas Roussel in our research team, as well as Daniel Vogel from the University of Waterloo.

Finger identification is an additional hand gesture property, which says which finger of which hand produced a given contact point. There is still no technology that senses this property directly in consumer electronic products. A workaround with existing technologies is to ask users to press all fingers before releasing some of them [151]. Other projects use different processing of sensed data. For example, the old generations tabletops used an infrared projector under a translucent surface, pointed towards the user. An infrared camera sensed the reflection of the infrared light that created blobs at contact points. The multi-touch coordinates were computed as the centroid of these blobs. However, the whole hand is visible on this image and it is possible to identify fingers with image processing of this data [75]. Several other studies in the literature use heuristics to detect finger chords. They detect the hand palm and deduce the position of fingers [10] or make assumptions based on the relative position of contact points [90, 257]. Other research investigated finger identification with prototyping technologies: optical fibers under a touchpad that enable the capture of fingerprints [123], a glove with piezoelectric vibration sensors [165], or a glove with fiducial markers [164]. In the next section, we will discuss the finger identification sensing technologies we used in our studies.

One of the simplest applications of finger identification is probably mapping a different command to every finger. Research showed that this is already sufficient to increase the input throughput of touch interaction [223]. This is particularly useful because multitouch applications typically cannot provide as many commands as desktop applications. For example, in 2014 Wagner *et al.* reported that there were 648 commands in the menus of the desktop version of Adobe Photoshop compared to 35 commands on the tablet version [257]. Interaction techniques that use finger identification are a solution. However, the input space is huge: $2^{10} - 1$ fingers chords, to which we can add gestures.

We will discuss how we systematically leverage this new input space, in particular how we reduce this input space.

2.3.1 Prototyping apparatus

This project is a typical HCI project in which we investigate the benefits and limitations of an interaction paradigm before the technology is ready to be implemented in consumer electronics products. The advantage is that research for such technology will happen only if and when the benefits will balance the costs. Even if such technology is not ready yet, the HCI community regularly uses alternative technologies that necessarily make compromises. In some cases, we just imagine the technology is there and works perfectly. The wizard of Oz technique consists in having an operator executing the actions on the users' behalf, with any other technology. This is not always even necessary. In the case of finger identifications, the Glass+Skin study just indicated participants the finger they wanted them to use. Other methods make assumptions on the inputs [90, 257], but it limits the resulting interaction techniques. We can also use alternative technologies. Either they only work for specific devices, like projection-based tabletops, or they use invasive methods like markers. We prefer the last method since it allowed us to build prototypes for tabletops, tablets, and smartphones, without limitations on the input vocabulary.

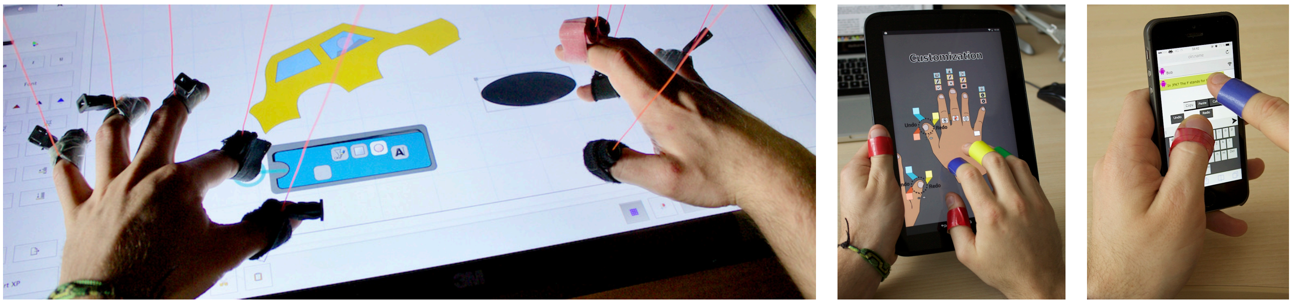


Figure II.8 – Three implementations of finger identification prototypes. The first one uses gametracks to locate fingers on a tabletop. The two other ones use color markers with a camera to track fingers on a tablet and a smartphone.

The Figure II.8 shows our three prototypes. The tabletop prototype combines two sources of information to identify fingers. The first one is the 2D coordinates of the contact points. The second one is the 3D position of every finger that we sense with GameTraks[11]. GameTracks sense a 3D position by measuring the length and angles of a string attached to a joystick. This method requires calibration and strings attached to every finger, but it is not sensible to occlusion for example. The tablet and smartphone prototypes used color markers on every finger, as well as an external camera. It requires calibration as well and it is sensible to occlusion. However, having no strings attached to the fingers is more convenient for interacting on smaller devices. These prototyping technologies are clearly not usable in consumer electronics products. But they are robust enough and add little constraints on multi-touch interaction, so they are convenient for exploring the input space of multi-touch interaction with finger identification.

---

[11] https://en.wikipedia.org/wiki/Gametrak

2.3.2 Input vocabulary

In this project we leveraged the input vocabulary of finger identification for command selection as well as the direct manipulation of the command parameters [**97**, **98**, **99**]. Without finger identification, the input space is just the number of contacts, between 1 and 10 for one user, and coordinates for each contact point. With finger identification, the theoretical input space is much larger: each finger can either be pressed or not, which leads to $2^{10} - 1$ possibilities. However, chords with too many fingers are less likely used. For a given number of fingers k, there are still C_k^{10} possible combinations of two-hands chords. This leads to 45 combinations of 2 fingers and 120 combinations of 3 fingers. People cannot perform all chords efficiently because of biomechanical constraints [90]. However, this input space remains still high therefore the design of a command selection technique requires a systematic approach.

The inspiration for our design rationale stems from command selection with keyboard shortcuts. Keyboard shortcuts use modifier keys and an alphanumeric key. Different combinations of modifier keys change the command activated with a given alphanumeric key. Then, when the associated action is being executed the users can apply constraints with modifier keys. The Figure II.9 shows how we applied this principle to multi-touch interaction with finger identification. The users start by touching the surface with their left hand. Some of the possible chords enable the selection of commands with the right hand. A crib sheet shows icons representing the five commands mapped to each of the five fingers of the right hand. In this example pressing both the thumb and index finger maps drawing commands to fingers. The user draws a rectangle by using the command associated with the middle finger. The initial contact point of the middle finger of the right-hand defines one corner of the rectangle, and the opposite corner is adjusted continuously with direct manipulation. During this step, the users can lift their left hand and continue to adjust the rectangle. Now, they may want to apply constraints. To do so, they touch the surface with the left hand. In this example, the middle finger sets the initial contact point as the center of the rectangle rather than the first corner. Another crib sheet shows an icon representing this constraint.

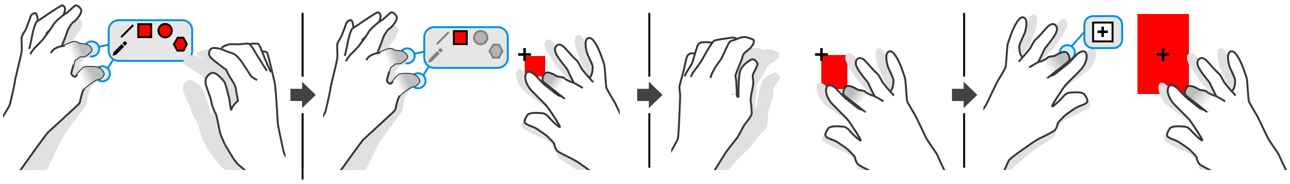


Figure II.9 – Command selection with finger identification. A chord with the left hand shows a crib sheet of commands users can select with the right hand. The user selects a command and adjusts the parameters with direct manipulation. The user can lift the left hand while adjusting the command parameters. The user can invoke additional constraints on the command parameters with the left hand.

2.3.3 Discussion

One of the key issues of this kind of system is the discoverability of the available commands. The Figure II.10 shows the visual feedforward items we use to promote the discoverability of commands. On the left, when all fingers touch the surface, bubbles show all the possible chords associated with modifiers or commands. On the left, crib sheets represent the visual icons of available commands or constraints for the current chord of the left hand. All this visual feedforward information is displayed around the corresponding contact points and follows them continuously when they move. The actual crib sheets are updated continuously as fingers touch or leave the surface.

Linear menus and toolbars in desktop applications show, in principle, all the available commands of an application. In our case, the visual representation of commands is only available on demand. The command mapping and constraints change in real-time as the users press or release fingers. It encourages users to explore the system and discover the available commands. This exploration would be an issue if the explorable input space was too large. However, we essentially use one-hand chords. There are therefore C_k^5 possible chords with k fingers, which gives 5 combinations of one or four fingers, and 10 combinations of 2 or 3 fingers. Therefore there are $5 + 10 + 10 + 5 = 30$ modifier chords for a maximum of 150 single-finger commands with the right hand. We assume this is a reasonable size for an explorable input space. However, we did not conduct a user study to evaluate this aspect and validate this claim.

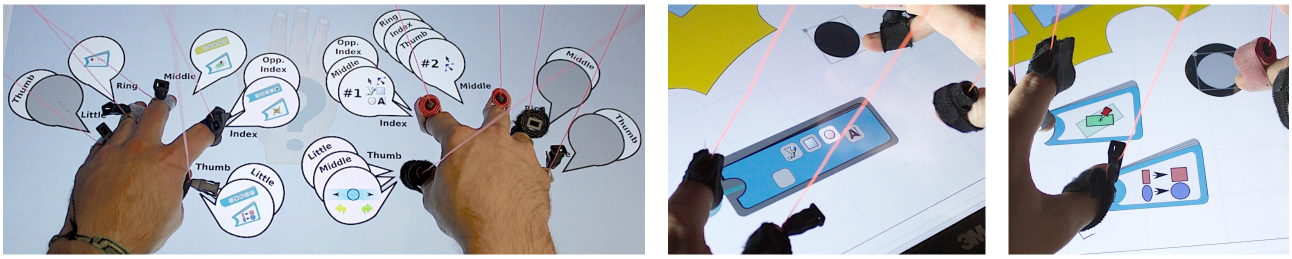


Figure II.10 – Discoverability features for multi-touch interaction with finger identification. Users can see all the available commands when touching the surface with all their fingers. Crib sheets show the available commands and constraints when fingers of the left hand are touching the surface.

Keyboard shortcuts with multiple modifier keys are sometimes difficult to execute because the position of the keys is fixed. Therefore sometimes it requires users to stretch their fingers to reach all the keys, especially if they want to keep one hand on the mouse. In this context, we studied the possibility to select modifiers on the mouse so that the left hand only has to select the alphanumeric key [**199**]. The input space remained limited, especially because we only had two modifier buttons on the mouse. In the case of our command selection technique for multi-touch interaction with finger identification, the advantage is that there is no actual button to press. Therefore, the location of the contact point does not matter. Hence, users can touch the surface with a comfortable posture.

2.4 Immersive Virtual Reality

Immersive virtual reality has been around since the beginning of personal computing [244, 245]. Early prototypes already included a stereovision headset with head tracking to match the users' movements with their position in the virtual environment. Yet, it took VR headsets much longer than personal computers to reach either households or professional environments. Even today that many different consumer electronics VR headsets are available at a reasonable price, their usage remains limited. Similarly to every interactive technology, the objective is not to replace completely concurrent technologies, but rather find the particular application scenarios for which this technology is more suitable than others. To do so, we need to know better how we interact in virtual environments.

The studies I will describe in this section focus on input methods for immersive virtual reality. This was Marc Baloup's work during his master internship that I co-supervised with Géry Casiez and his Ph.D. that I co-supervised with Géry Casiez and Martin Hachet. It was part of the Avatar project funded by Inria. In this work we take immersive virtuality reality as a context that constrains the input methods we can use, in which users have to perform specific tasks. One of these constraints is that the users cannot see their own bodies because the virtual environment covers their entire field of view. Therefore, either a motion capture system senses the users' body movements, or they hold input devices in their hands. Users interact with their environment mostly with gestures, but also with buttons, touchpads, and joysticks on handheld devices. As we discussed at the beginning of this chapter, the objective in virtual reality is not always to reproduce what exists in the physical world. We rather focus on either elementary or compound tasks and figure out ways to enable users to perform these tasks. For example, selecting an object is an elementary task that is usually part of more complex or compound tasks: manipulation, command selection, navigation, text entry, etc. We describe below the design and evaluation of a new 3D pointing technique for immersive virtual reality.

One of the main active research topics in this area is augmenting *immersion*, *presence*, and *embodiment* [135, 234, 265]. In short, these three concepts are related to the users' sensation of being in the virtual environment. I will cover this topic in the next chapter, section III.2.2. Yet, a typical way to increase the users' sensation of embodiment is to provide them an *avatar* that they can control. The users can typically control their avatar's hands and head because it is easy to map their position to the VR headset and the controller's position with inverse kinematics. The facial expression of the avatar is important for non-verbal communication. However, it is more complex to control than the head and hands because it requires many degrees of freedom. Therefore we describe below the design and evaluation of an interaction technique for the control of the facial expression of an avatar in immersive virtual environments.

We used a common general methodology for the design of these two interaction techniques. We identified the key design aspects of the techniques and we implemented and evaluated several alternatives. Then we implemented the best combination of these alternatives and compared it to equivalent techniques in the literature.

2.4.1 3D Pointing

In the physical world, it is difficult to interact with objects remotely because of physical constraints. In virtual worlds, such physical constraints do not exist. Users can interact remotely with objects, regardless of their size, shape, or weight. The most common technique for selecting objects is certainly *raycasting* [36]. The users control a ray with their hand, and the first intersected object can be selected with a validation action, such as a button press. It shares similarities with laser pointers, except that the light ray between the controller and the contact point is visible. The simplicity of this technique has a cost. Occluded targets cannot be selected, or require users to get around them. Moreover, even if theoretically there is no limit to the distance and size of the object the users would like to select, in practice hand tremor and input noise create such a limit. Therefore, there are many adaptations of this technique in the literature, well covered in [7]. Most of these techniques use a disambiguation step to let the users select a target among all the interested targets [66, 105, 141]. Other techniques use a cursor on the ray that users can either manipulate by manipulating the ray [105], or with another degree of freedom [217].

These solutions fix the issue of occluded targets but not the issue of small and distant targets. To do so, we propose RayCursor, an alternative of raycasting with a cursor that users can control that leverages proximity selection [**14**, **15**, **16**]. Proximity selection consists in selecting the nearest target from the cursor instead of the intersected target [104]. Initial studies were with 2D pointing. However, it was also studied for 3D pointing with a virtual hand [254], that is to say pointing at targets the users can reach around them. We combined this approach and raycasting with a cursor. For the design of this technique, we performed three initial evaluations. The first one was about the most appropriate visual feedforward and was inspired by a similar work for 2D proximity selection [109]. The second evaluation was about the transfer function for the movement of the cursor, and the last one was about the benefits of filtering the inputs that control the ray with a 1€ filter [53]. After this, we designed a semi-automatic RayCursor that combines RayCursor and raycasting (Figure II.11). Finally, we compared the performance of the two versions of RayCursor, raycasting, and another technique of the literature [217].

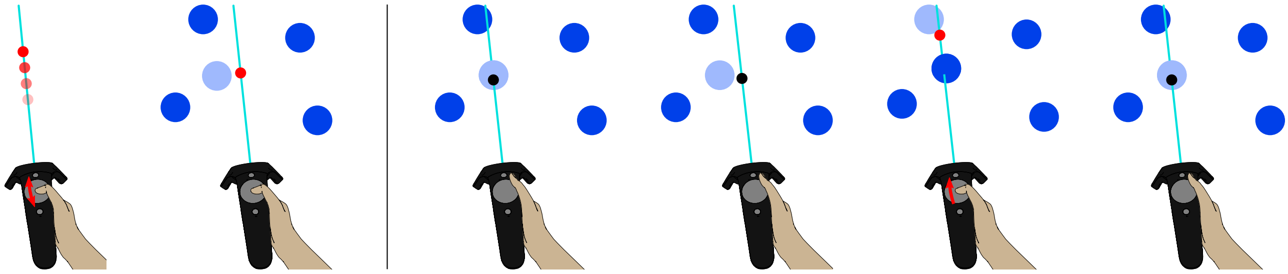


Figure II.11 – Two versions of Raycursor. The manual RayCursor (left) selects the nearest target from the cursor. The semi-automatic RayCursor acts like raycasting, a black cursor is positioned on the first intersected target. When the ray moves out of a target, it remains selected while it is the nearest target. The users can move the cursor manually with the touchpad, which turns red, to select another target. If the users lift their finger for more than 1s, the cursor switches back to its initial behavior.

2.4.2 Facial expression selection

Research on virtual reality and the possibility of creating immersive virtual environments inspired many science-fiction authors. Decades after Sutherland's work [245], novels like Neuromancer [94], Snow Crash [242], and more recently Ready Player One [57] describe immersive virtual worlds as alternate realities in which people can socialize, play, or even work. The new brand of Facebook, Meta, is a reference to Snow Crash's *metaverse* and shows the new focus of the company on an immersive social network, with Spaces then Horizon[12]. Similarly to other immersive social networks like Mozilla Hubs[13], VRChat[14], and RecRoom[15], the objective is to enable people to get together in a virtual environment and interact with each other as if they were in the same room.

Such immersive virtual environments require easy and usable ways to perform atomic actions such as the study presented in the previous section. But above all, communication is certainly the most important aspect of social networks in general. Text entry remains a difficult and tedious task in immersive virtual environments, and the current best solution is to simply use voice. However, non-verbal communication such as facial expressions is also an essential aspect of communication, whether for speech in the physical world or by writing [48]. One way to enable users to control the facial expression of their avatar is to detect their own facial expression, we call this isomorphic control. Vision-based techniques use either external depth cameras[156, 262] or cameras embedded in a VR headset [154, 246]. However, with such techniques facial expressions are limited to expressions users are able to perform. Moreover, users cannot give their avatar a different expression than their own. Therefore we investigated the non-isomorphic control of facial expressions, with interaction techniques [**17**].

The fine control of facial expressions requires many degrees of freedom. For example, the FACS standard defines 24 Action Units [71], and the MPEG-4 proposes 68 Facial Animation Parameters[191]. Therefore we propose to reduce the number of degrees of freedom by decomposing the selection of a facial expression into several sub-tasks, similarly to Bowman's decomposition of 3D interaction tasks [36]. The sub-tasks are: selecting a facial expression, its intensity, duration, and ending. The selection consists in choosing a facial expression among a list of pre-defined expressions. Each predefined expression is a configuration of FACS action units values. The Figure II.12 shows four of the facial expression selection techniques we designed, the fifth one uses voice commands. These techniques are essentially item selection techniques, similar to what we would use for command selection. Similar to command selection, some of the facial expressions share properties, which we leveraged to structure the layout of items in some of the techniques.

The visual representation of facial expressions in our techniques uses emojis. We made this choice because they are frequently used in messaging and social networks. Also, people can even identify small versions of them. If we restrict facial expressions corresponding to emotions, we can leverage

---

[12] https://www.facebook.com/spaces, https://www.oculus.com/facebookhorizon/

[13] https://hubs.mozilla.com/

[14] https://hello.vrchat.com/

[15] https://recroom.com/

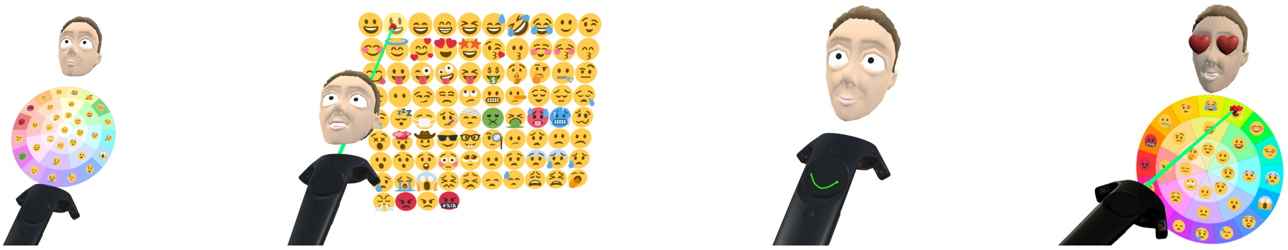


Figure II.12 – Four of the facial expression selection techniques: 2D circular menu arranged by emotion, raycasting 2D grid menu, touchpad gestures, raycasting on the 2D circular menu arranged by emotion.

models of emotions such as PAD [166] or Plutchik's wheel of emotions [204]. The layout of our pie menu selection technique is adapted from Plutchik's wheel, which organizes emotions along 8 axes representing 8 base emotions. The difference between Plutchik's wheel and our menu is that we mapped the maximum intensity to the edge of the circle rather than to the middle so that the center represents the neutral face. The grid menu allows the selection of any facial expression, regardless of whether they represent an emotion or not. In particular, users can select emojis with decorations such as hearts, tears, or glasses for example. In this case, we render these decorations on top of the avatar's face. Transitions between two facial expressions are made by interpolating the values of each FACS action unit. Our evaluations show that participants made more errors with the gesture and voice techniques, and they preferred the other techniques.

We designed several techniques for controlling the intensity, duration, and ending of facial expressions all at once. Users define the duration indirectly when they end the facial expression. The first technique maps the intensity to the controller trigger. The facial expression ends 1 s after the user released the trigger to avoid incidental endings. The other techniques below end when the user releases the selection button. With the second technique, users have to shake the controller and the intensity is mapped to the shaking speed. The third technique is similar, users must roll the controller and the intensity is mapped to the roll angle. The fourth technique creates a visible elastic band between the selected expression and the selection ray. The intensity is mapped to the length of the elastic band. This technique requires a ray to which the elastic band is attached, therefore it cannot be used with a subset of the facial expression selection techniques. The last technique only works with pie menu techniques and maps the intensity to the distance to the center of the menu. Our evaluations show a subjective preference of the elastic band, trigger, and orientation techniques over the shaking technique.

We evaluated the usability of the pie menu with a raycasting selection and intensity control with the elastic band. The results show that controlling their avatar's facial expression can disturb users while they are talking or listening to somebody else. However, the perceived interruptions were reasonable.

2.4.3 Discussion

More than fifty years after the first virtual reality headsets, the use of immersive virtual reality is still in its early stages. Social media companies recently started investing massively on this topic. When people evolve in such virtual environments, they are free of many constraints of the physical world. Therefore when we design interaction techniques to enable people to perform actions in such virtual environments, we also think about ways to perform actions they could not in the physical world. The two techniques we discussed in this section allow users to perform two distinct actions: select objects and show facial expressions. However, they both have in common that they enable users to perform these actions in ways they could not in the physical world. In the physical world, it is impossible to select an object remotely and show a facial expression different from their actual facial expression. The choice of using these techniques in virtual reality systems is indeed a matter of trade-offs.

Most of the benefits of Raycursor rely on proximity selection. The actual benefits of proximity selection were demonstrated with the bubble cursor in both 2D [104] and 3D with a virtual hand [254]. We extended this knowledge to remote selection in 3D. The consequence of proximity selection is that there is always a selectable object, and the area of influence of the targets is a Voronoi diagram. The issue we overlooked in this work is the exact way to interpret the area of influence of 3D objects, as opposed to points. We conducted evaluations with spheres and, as a 3D extension to the bubble cursor, the distance to the sphere is the distance to the center minus the sphere radius. With arbitrary objects, it is unclear which point should be considered. This choice has a major effect on the area of influence. We identified issues with convex and oblong objects, as well as dense clusters of small objects surrounded by large objects. Solutions remain to be investigated for such cases.

The main argument for using a non-isomorphic technique for selecting a facial expression is to enable users to show a facial expression different from their actual facial expression. This is typically what we do when we use emojis in messaging or social media apps. In particular, it allows people to show a specific facial expression even if they are not able to perform it at will. Not everybody is an actor. Now, isomorphic techniques also have benefits. They do not require conscious control and they do not disrupt or interrupt a primary task, like talking. Moreover, new VR headsets include cameras for face tracking[16], which will make it easier to implement such interactions techniques. Therefore it would be interesting to investigate the combination of isomorphic and non-isometric techniques to get the best of both and study the control/automation trade-off in the context of facial expression selection.

---

[16] https://www.vive.com/eu/accessory/facial-tracker/

3 Conclusion

When I started writing this chapter, the idea was to highlight the symmetry between output and input, in particular in the case of haptics. The previous chapter was about the sense of touch, one of the five senses. I was however surprised by the difficulty for me to find an equivalent of the word “senses”, as the human input systems, for the human output systems. I decided to use the word *ability*, which refers to the ways humans have to act on their environment, the same way their *abilities* enable them to get information from their environment. In fact, humans do not have many kinds of ways to act on their environment. This is maybe the reason why I could not find the word I was searching for. To the best of my knowledge, humans can do so with movements, voice, and fluid secretions.

This chapter focused on *motor abilities*, which are abilities that leverage the human motor system. The motor system enables people to touch and manipulate their environment, therefore it is the output part of the human haptic system. I described on Figure II.1 the process between the moment a user plans a manipulation action and the moment the system produced information based on the effects of this manipulation it sensed. Similar to the analogous figure in the previous chapter (Figure I.1), this description clarifies the software and hardware parts on both the human and system side, as well as the connection between humans and systems. This process revealed general research questions that I addressed in some of my research projects.

The first research question was about the way we design and evaluate the sensing and interpretation of physical effects resulting from human manipulation with an interactive system. This question is essentially related to the system. In particular, I discussed a methodology and tool we designed and implemented to measure and slice the latency of interactive systems.

The second research question was about the interactive input vocabulary. This is a whole research domain in itself. It involves both users and systems because the system produces the input vocabulary, and users need the physical ability to perform the appropriate actions. I discussed first the design of alternative input methods with flexible pens. Then I discussed the mapping and reduction of degrees of freedom with finger identification for multi-touch interaction.

The third research question was about unnatural input, which are ways to leverage the properties of virtual environments to perform actions that are difficult or impossible in the physical world. This research is essentially about interaction techniques, therefore it is both about the users and the system. The contributions I describe are interaction techniques for immersive virtual reality. The first one is a distant 3D pointing technique with proximity selection The second one is an interaction technique to select facial expressions for an avatar.

Indeed, all the studies we discussed in this chapter focus on hands dexterity and our capacity to touch and manipulate. However, the haptics as the sense of touch was barely used.

The latency measurement study assumed the output was visual. We performed measurements on Linux, MacOS, and Windows with several graphics library, that send data to a graphic card connected to a monitor. We showed that this part of the process was the major source of latency. Haptic systems are quite different, and there is no clear standard. There is typically a haptic look running around 1kHz that computes forces or vibrations faster than the human haptic sensitivity [225]. Audio-haptic systems with physical simulations use loops up to 10 kHz [150]. In the future, I would like to study the effect of the haptic loop on the perception of different kinds of force models. I will also measure the latency of several types of haptic systems, to compare them to visual systems.

The flexible pens we described in this chapter provide passive force feedback with their flexural stiffness. We build several prototypes of different flexural stiffnesses and studied their effect on both objective and subjective measures. However, we could also provide active vibrotactile feedback with an actuator. With controllable haptic feedback, we could give users a haptic immediate response when the users are bending the stylus. I hypothesize that such feedback, like haptic detents, would help users control the bend of the device with continuous gestures. We could also create discrete input with click sensations for activation gestures. The overall idea would be to use haptic feedback to support direct manipulation. We will discuss this concept in a different context with a haptic wristband in Section III.2.1.2 of Chapter III.

Our multi-touch interaction paradigm with finger identification heavily relies on visual cues. It uses visual feedforward, feedback, and uses additional visuals for promoting learnability and discoverability. Tactile feedback on touch surfaces is usually poor. At best smartphones have a low-quality tactile actuator. We started this project with tabletops, which are difficult to actuate. We discussed the design and evaluation of several potential technologies for this in Section I.2.2 and Section I.2.3. It would be interesting to see how such haptic feedback can reduce the visual clutter of our multi-touch interaction paradigm with finger identification.

Finally, we discussed the way that immersive virtual environments enable users to perform actions that are difficult or impossible to perform in the physical world. In particular, we discussed two contributions, the first one was a distant pointing technique with proximity selection and the second one was a facial expression selection technique. These techniques use basic haptic feedback, with simple clicks upon selection activation for example. However, it does not permit users to feel objects they touch in the environment. Haptic devices for immersive Virtual Reality is an active research topic [35]. However, solutions usually focus on particular manipulations. One of my current projects is about visuo-haptic integration and includes the design of an expressive vibrotactile VR controller.

III The Sensorimotor loop

> The most exciting phrase to hear in science, the one that heralds new discoveries, is not 'Eureka!' but 'That's funny...'.
>
> Isaac Asimov

We discuss the limits of the approach presented in the previous chapters that consist in focusing on either input or output. We present two users studies in which this approach is not sufficient to improve interaction. Then we discuss the critical role of the sensorimotor loop through models of human behavior, as well as system architectures and paradigms. We combine these approaches and discuss their positioning to the approach discussed in the first two chapters. Finally, we illustrate this approach with two contributions. The first one is alternative interaction paradigms that leverage the sensorimotor loop. The second one is a series of studies around the sense of embodiment in immersive virtual reality.

The previous chapters focused on the way we leverage haptics for the design of either input or output systems. The first chapter, about haptic as the sense of touch, barely mentions input. On the opposite, the second chapter, about haptic as the motor ability, barely mentions output. This separation sometimes seems to make sense. For example, Tactons used to convey information for notifications do not necessarily require user input. However, combining output with gestural input enable users to get more detailed information [193]. Multi-touch and gestural interfaces generally use visual feedback to compensate for the lack of physicality of controls. But the problem remains the same: whether the feedback is visual or haptic, it requires a systematic design.

The message of this chapter is that input and output have equivalent importance for the design of interactive systems, and must be designed together. This is not as trivial as it seems to be at first sight. To illustrate this, I describe below two parts of rejected submissions and explain what was wrong with them. It had to do with the motivations and the lack of focus on the sensorimotor loop. It made me realize that this is a large hole in the two previous chapters. Therefore I will discuss the sensorimotor loop, its connection to both human and computing models, and postion this approach to the approach of the first two paragraphs. I will conclude with contributions that illustrate the critical role of the sensorimotor loop in interaction.

1 The limits of the separated approach

During my postdoc at the University of Toronto under the supervision of Ravin Balakrishnan, I worked on the lack of haptic feedback in 3D gestural interaction. The Microsoft Kinect[1] was just released. It featured a motion capture system affordable for households. It followed the trend of *natural* user interfaces, and was advertised as "*You* are the controller"[2]. Besides the discussion about the natural aspect of user interfaces and their relevance in the previous chapter, this motivation to eliminate physical controllers also eliminated many useful, not to say essential, haptic properties of physical controllers. This was already an issue with touch interaction, but there was at least the passive haptics of the interactive surface. In the case of 3D gestural interaction, the users have no haptic feedback when they interact with virtual objects. Therefore, the immediate feedback they need for direct manipulation [232] is essentially visual or auditory. This contributes to the overload of these modalities. Therefore, I worked on a way to provide haptic feedback for 3D interaction.

The design rationale was to keep the users' hands free since this was the main motivation of this type of sensor. If users have to hold a device for haptic feedback, this device could also serve as an input controller. Therefore we opted for a wearable haptic device for the wrist.

---

[1] https://tiny.one/Kinect

[2] https://tiny.one/youAreTheController

Apparatus

There are several reasons to discuss the design and implementation of the prototype in this document. First, it is designed for expressivity with simplicity. Second, the iterative process is an interesting case study that led to guidelines regarding the implementation of interactive systems. It helped me with the ideation of the concept depicted in Figure I.1 in Chapter I.

We discussed the output vocabulary of vibrotactile feedback in Chapter I. The independent control of both the frequency and amplitude of a vibrotactile signal is necessary for an expressive output vocabulary. It requires precise vibrotactile actuators such as voice coil actuators. We used EAI C2 tactors [171]. The typical way to drive precise vibrotactile actuators is to use a sound generation system. This is convenient because the parameters of the signal are the same: frequency, amplitude, and shape. The main difference is the frequency range: 1–1000 Hz for haptics and 200–20 kHz for sound. Managing the amplitude is easier with vibrations because the required amplitude levels are much lower. In the end, the shape parameter is in my opinion the bottleneck of complexity for the implementation of vibrotactile devices because it imposes a much higher sampling rate. It makes the design of sound generation systems complex, especially with microcontrollers available at the time this project started. For the sake of simplicity, I rather opted for a straightforward design that enabled the precise control of both frequency and amplitude at the cost of a low control of the signal shape[3]. The idea is to control the frequency and amplitude with two PWM signals generated by the timers of a microcontroller (Figure III.1). The frequency signal typically ranges between 1–1000 Hz. The amplitude is controlled by the duty cycle of a high-frequency signal. We used voice coil actuators, therefore they behave like low-pass filters, which stabilize this high-frequency signal, hence reducing the amplitude of the actuator's movement. Our prototypes used 16 MHz controllers with 8 bits timers, which gives a 62.5 kHz loop with 256 levels of amplitude. It communicated with a host computer with a serial protocol over Bluetooth.

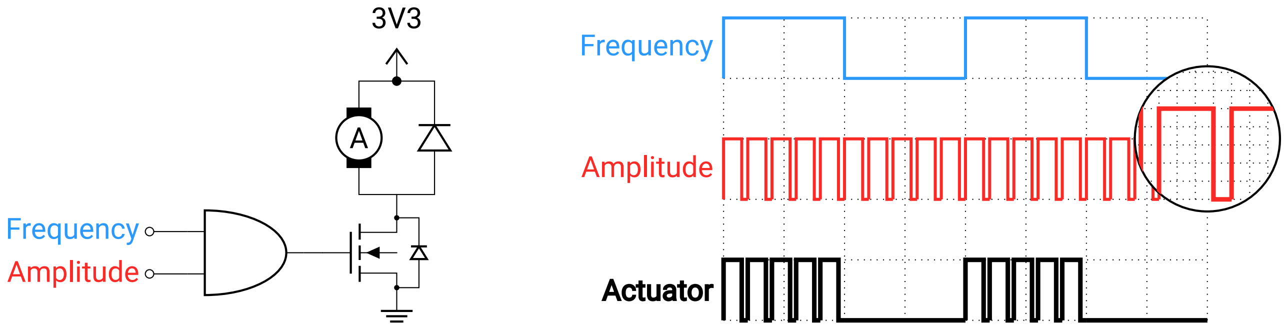


Figure III.1 – On the left: electrical diagram for driving the actuators. Two signals, one for Frequency and the other for Amplitude are modulated. On the right: a low-frequency signal with 50% duty cycle for Frequency, a high-frequency signal with an adjustable duty cycle for Amplitude ($\frac{3}{4}$ in this example), and the modulation of both signals to drive the actuator.

---

[3] Controlling the signal shape remains possible, with a software $\Delta\Sigma$ modulation for example https://tiny.one/DeltaSigma.

As shown on Figure III.2, the design of the prototypes was iterative. The first two prototypes used Arduino LilyPad microcontroller boards [42]. These boards are designed for wearables, so they made sense for this project. On the first prototypes, the components were sewn with conductive thread. It caused several issues though. First, the thin conductive thread had a non-negligible resistance. Therefore it was necessary to duplicate the connections to have enough power flowing in the circuit. Second, the elasticity of the wristband was convenient for comfort, but it caused short circuits that made the device unreliable. To alleviate these issues, I soldered the components on small protoboards, which I connected together with conductive thread. Now the issue was the connection between the thread and the pads. Using conductive glue was only a temporary fix because it would not stick long to the pads. Given the recurring technical issues, I designed a PCB for the third prototype, which was connected to the microcontroller, battery, and actuators with wires. This new prototype uses an Arduino Mini Pro microcontroller board[4], which includes the same microcontroller as the LilyPad: the Atmega 328P[5]. This is a convenient 8-bits microcontroller for small designs. The issue with it was that it only has six timer channels, and I needed eight to control four actuators in frequency and amplitude. Therefore I used a common frequency signal for all actuators, and an individual Amplitude signal. This choice made sense given the type of tactile effects we planned to use, which we will discuss in the next sections.

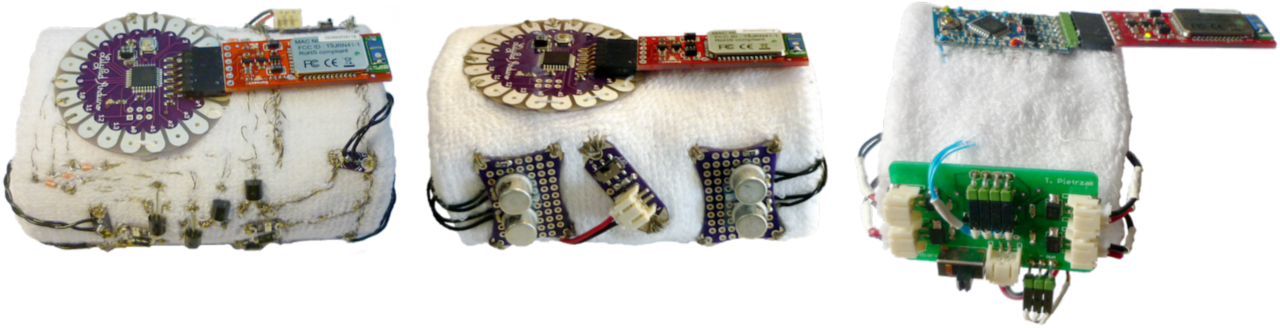


Figure III.2 – Three iterations of haptic wristband prototypes. On the first prototypes, components were connected with conductive thread. On the second prototype, the components were soldered on protoboards that were connected with conductive thread. The third prototype used a homemade PCB connected with wires.

The takeaway is that prototyping interactive devices requires both keeping in mind the design rationale and the technical constraints of their implementation. It is critical to balance the trade-offs and make informed compromises. For example, replacing conductive thread with wires was not a compromise after all, and make the prototype more robust. It was a constraint motivated by a bad choice of microcontroller board from the beginning. The restrictions on signal shape and the common frequency were actual compromises. But they enabled a simple and robust design while keeping an expressive output vocabulary.

---

[4] https://tiny.one/ArduinoMiniPro

[5] https://tiny.one/ATmega328P

1.1 Quantitative limitations

The first rejected submission on this work described several scenarios in which 3D gestural interaction was supposed to benefit from tactile feedback. I expected quantitative benefits, the same way tactile feedback improved typing performance on a mobile soft keyboard [121]. Therefore I described several scenarios for 3D gestural interaction in which I provided tactile feedback. I evaluated the quantitative benefits for some of them, in particular for target selection.

3D gestural interaction has two main issues. The first one is the precision of the movements, due to both the stability of the hand and the noise of the input system (see Figure II.1 in Chapter II). The second one is the lack of segmentation. Physical controls typically have at least two states among *out-of-range*, *tracking*, and *dragging* [45]. 3D gestural interaction only has a *tracking* state. Therefore, workarounds are necessary even for basic interactions such as target activation. Kinect applications typically use a dwell cursor, that requires users to remain still over buttons for a few seconds to activate them. We replicated these cursors as depicted on III.3. In Kinect applications, both the immediate feedback telling users they hover a button and the dwelling progress is indicated with visual feedback. At the time, we assumed that complementing this feedback with haptic feedback would be beneficial for users. Therefore we implemented the tactile dwell cursor the following way.

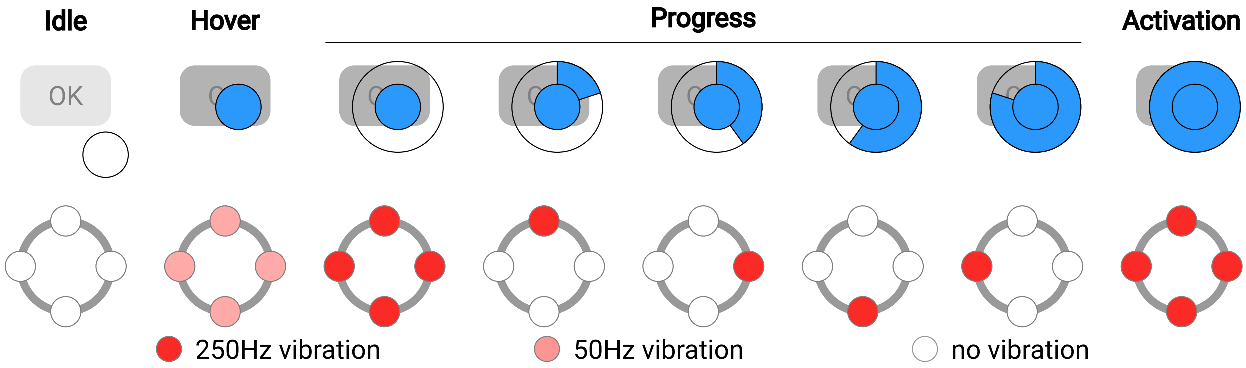


Figure III.3 – Tactile dwell cursor. When the cursor is not over a button, the users receive no feedback. When it is over a button, the users feels a 50Hz vibration all around their wrist. If the they stay on the button, the progress animation starts with all vibrators at 250Hz. Then the four actuators vibrate in a sequence. When the sequence finished all the actuators vibrate and the command is activated.

The cursor has no color and gives no tactile feedback when it is not over a target. When it hovers a target, it is colored and all the four actuators vibrate at 50 Hz. After 1 ms over the button, a ring is drawn around the cursor, and all the four actuators vibrate at 250 Hz during 150 ms, then they stop for another 150 ms. It warns the users that the animation is about to start. The animation vibrates the four actuators in a clockwise sequence at 250 Hz during 200 ms, followed by a 175 ms pause. After this animation all the actuators vibrate at 250 Hz for 200 ms, then the target is activated. Therefore, overall users had to hover a button during 3 s to activate it.

We ran an experiment with the idea to measure an increase in performance in a tactile condition over a visual condition. This was motivated by the fact that when we tried the buttons, tactile feedback seemed to bring some benefit. We did not have a clear idea of *what* was better though. We presented to participants a screen with an array of four-by-four buttons. They had to select a series of buttons indicated with a highlight. The details of the experiment do not matter. We failed at detecting a significant difference between the conditions, either in selection time or error rate. I am convinced now that performance is not the benefit of tactile feedback in this situation.

That being said, several participants reported that tactile feedback was a good addition to visual feedback, as it reduced their visual attention. For example, two participants said: "without tactile feedback, I have to focus more on the visual feedback" and "I found [tactile feedback] more helpful than the visual feedback because I didn't have to focus 100% visually with the tactile redundancy." Users also appreciated that tactile feedback indicated when they hovered a button. For example, one of the participants said: "It felt more like interacting with a physical button when activating the button produced a sharp buzz." Therefore, the benefits of the tactile feedback seemed to be qualitative rather than quantitative. This is why in the next iteration we focused on the qualitative benefits of tactile feedback for gestural interaction.

1.2 Qualitative limitations

The dwell buttons scenario did not seem to be the most appropriate for evaluating qualitative benefits. Another scenario for haptic 3D gestural interaction in this project was an open-source car racing game[6] that I adapted for Kinect (Figure III.4). The users steered the car by moving their arms as if they were holding a steering wheel. The Kinect API computed a skeleton of the user. The steering angle was computed as a function of the relative position of the hands of the skeleton, capped between −90° and 90°. Braking was mapped to 15–30 cm between the hands and the chest, and throttle to 30–50 cm. The spatial location of vibrations indicated the steering angle as shown on Figure III.4. The speed of the car was mapped to a modulation of the 250 Hz signal with a low-frequency signal between 1 Hz and 25 Hz with 50 ms durations. This modulation makes users feel as if equidistant strips covered the road. In addition to this, the bottom actuator vibrated for 200 ms at 100 Hz when the car was braking.

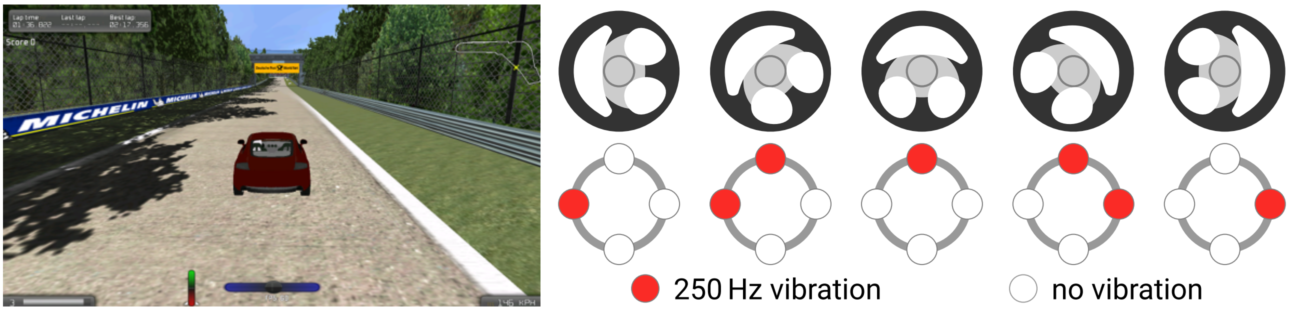


Figure III.4 – Screenshot of the customized version of VDrift, an open-source car racing game. The right side shows the mapping of the vibrotactile feedback depending on the steering angle.

---

[6] https://sourceforge.net/projects/vdrift/

In a first pilot study, we compared performance between the tactile feedback and no tactile feedback conditions. In hindsight, there were few chances haptic sensations increased such metrics. However, participants reported that the tactile sensations provided them benefits in terms of realism and immersion in the game. Therefore we conducted the following experiment, which focused on qualitative benefits. We made the hypotheses that H_1 tactile feedback would increase the sensation of realism of the game because of the increased sensory stimulation, and H_2 tactile feedback would increase the users' sensation of control because it provides them feedback about the way the game interprets their actions.

1.2.1 Methodology

40 participants took part in the experiment (mean age 24.7 years). They were instructed to drive as many laps as possible in 10 minutes. They were advised to avoid going out of the road since it notably reduces the speed. They stood 2.5 m away from the Kinect during the game. The game was displayed on a 17" laptop screen, with a 1600 × 900 resolution, and the same sound volume was used for all subjects. We opted for a between-subjects design: half of the participants received tactile feedback (T), and the other half did not (NT). We explained the mapping of the tactile feedback to the participants of condition T. Tactile feedback was not mentioned to the participants of condition NT.

Before they performed the task, the participants filled the immersion tendency questionnaire (ITQ) [265]. It measures the ability to get involved and focused on tasks, playing habits, and the tendency to get immersed in games. After the experiment, we measured spatial presence with the presence questionnaire (PQ) [32]. It measures the sensation of control, perception of sensations, distraction from the task, and realism. We added three additional questions about tactile feedback adapted from questions about audio feedback: 1) How much did the tactile aspects of the environment involve you? 2) How well could you identify the vibrations? 3) How well could you localize vibrations? We analyzed our results with T-tests, Pearson correlation, and Cronbach's alpha.

1.2.2 Results

Figure III.5 shows the results of the ITQ and PQ questionnaires. The reliability of the ITQ questionnaire is acceptable ($\alpha = 0.75$). Participants had a mean immersion tendency of $116.05/189$ in the T condition and $117.5/189$ in the NT condition. We did not detect any significant effect ($p = 0.72$).

The PQ questionnaire had a good reliability in both T ($\alpha = 0.89$) and NT ($\alpha = 0.88$) conditions. Participants in the T condition had a $159.75/245$ mean overall Presence score whereas participants in NT condition have $142.5/245$. We detected a significant difference between the two conditions ($p < 0.001$). The analysis shows a significant difference between *Sensory* ($p < 0.001$) and *Realism* ($p < 0.001$) factors. However, we do not detect difference in *Control* ($p = 0.55$) and *Distraction* ($p = 0.90$).

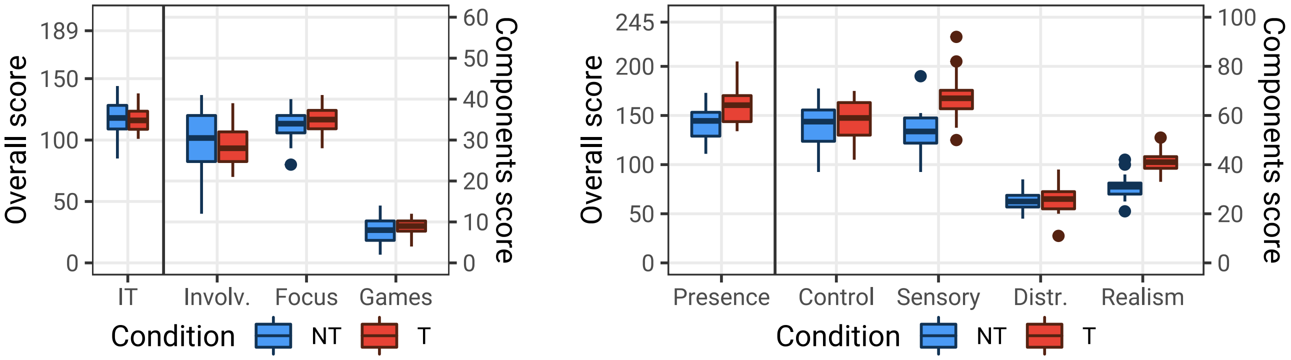


Figure III.5 – Results of the Immersion Tendency Questionnaire and Presence Questionnaire.

1.2.3 *Discussion*

The absence of detected difference in *Immersion Tendency* between the T and NT conditions with a small difference of means and a high p-value tends to indicate that there should not be a strong bias between the participant groups. However, this is not a definitive argument, and the problem of splitting participants into homogeneous groups is still an open question and we are currently working on it.

Regarding our research question, we first hypothesized that tactile feedback would enhance the realism of the game (H_1). The results of the *Realism* items of the *Presence Questionnaire* support this hypothesis. In addition to this, five participants of the NT condition explicitly reported they did not have a sensation of speed while several participants of the T condition spontaneously mentioned feeling the speed.

Our second hypothesis (H_2) predicted that tactile feedback would enhance the user's sensation of *Control*. The results do not support this hypothesis, therefore we cannot make any conclusion regarding this hypothesis. However, participants indeed reported control issues. They had difficulties braking because they had to move their arms very close to their chest, and it was difficult to turn at the same time. Indeed this issue was happening in both conditions and tactile feedback did not help.

1.3 Discussion

The paradox here is that the motivation for 3D gestural interaction was to offer users *natural* ways to interact with systems, without manipulating artificial artifacts. Whether it was relevant or not, without haptic sensations the virtual objects manipulated do not *feel* real to users, and they sometimes struggle to manipulate them because of this.

Restoring haptic feedback on dwell buttons was not sufficient to make them efficient. My hypothesis was that restoring tactile feedback would help users press them. I anticipated that this would increase selection performance. However, with the way I implemented these buttons activation required at least 3 s. This is very long indeed. But it was a replication of buttons in Kinect menus. Should tactile feedback help select buttons faster, the effect size could not be sufficient to make a significant improvement. The average selection time in the NT condition was 3.6 s, and it was 3.5 s in the T. This is the first clue that improving haptic feedback is not sufficient. It requires an efficient input method as well. Otherwise, the eventual benefits of haptics do not compensate for the inefficiency of input. In Section III.2.1 I describe a new paradigm for 3D gestural interaction with haptic feedback, and the systematic design of both the input and output vocabulary to enable direct manipulation on a tactile display.

The qualitative study with the car racing scenario showed a similar issue. Tactile feedback did provide qualitative benefits. However, the low quality of inputs diluted these benefits and made them barely measurable. The haptic feedback provided was questionable as well. I used spatial location around the wrist. However, the task required participants to rotate their wrists. Therefore it might have influenced negatively how participants interpreted the tactile cues. Unfortunately, I did not investigate this issue at the time. Moreover, the haptic cues indicated the steering angle of the wheel, not the car. The rationale is that it provides users immediate feedback on their actions, which is lacking without a physical controller. However, it does not provide haptic feedback about the result of the action. Regrettably, I did not investigate this aspect either. On the positive side, I did observe an effect of tactile feedback on presence. This is an interesting result for the Virtual Reality community, and I regret this result was not published in the end, hence the discussion here. However, later we investigated a related question with the effect of haptic feedback on the sense of embodiment, which we will discuss in Section III.2.2.

2 Contributions

In the previous chapters, I presented contributions to improve output by leveraging the sense of touch, and input by leveraging the motor abilities. In the previous section, we discussed that this approach is not always sufficient to improve interaction. The two examples above illustrate two major problems. The first problem is when we solve a local problem rather than a global problem. Such solutions sometimes patch a couple of issues, but the problem is more general and it is necessary to take a step back and analyze the problem in a holistic way. The second problem is the difficulty to identify and measure the real benefits of proposed solutions.

The contributions below use the orthogonal approach as discussed above to improve interaction by leveraging the sensorimotor loop. The first contribution provide quantitative benefits with two interaction paradigms that leverage gestural interaction and vibrotactile feedback. The second contribution investigates qualitatives benefits of the sensorimotor loop on the sense of embodiment of an avatar in Virtual Reality.

2.1 Haptic interaction paradigms

In Section III.1 we first wanted to measure the quantitative benefits of haptic feedback for gestural interaction. The interaction paradigm we used was so inefficient that haptic feedback could not compensate for its limitations. Here we propose a new interaction paradigm that gets around these limitations. Then, we will discuss a new interaction paradigm that brings direct manipulation to tactile displays. I implemented both paradigms with the device described in Section III.1. These two studies were the continuation of my postdoc work discussed above. I collaborated with Aakar Gupta during his Ph.D. at the University of Toronto under the supervision of Ravin Balakrishnan. He came twice for 6 months internships to work on these projects, during which I co-supervised him with Nicolas Roussel.

2.1.1 Summon interactions

The main limitations of 3D gestural interaction we discussed in Section III.1 are tracking difficulties and the lack of segmentation. The users are tracked without interruption. Therefore gesture segmentation is difficult. Moreover, every gesture the users perform is potentially interpreted. This is called *Midas Touch*, as a reference to the curse of the king that turned everything he touched into gold in the Greek mythology. There are also issues when the user is outside the sensor field of view, either on the edges or when it is occluded. The users need additional feedback for this in order to avoid usability issues. And even though, it makes interaction more complicated.

These issues make it difficult to use standard GUI widgets. We discussed in Section III.1 the simple case of buttons that require a different activation mechanism. 3D gestural interfaces typically use dwell buttons that require users to hold their hand still over a button for a couple of seconds to select it. We proposed to simply add haptic feedback, but we were unable to measure a quantitative benefit. I believe we need deeper changes to improve interaction in this context. Therefore we proposed a different paradigm that does not rely on pointing & selection. This new paradigm relies on summoning & selection [**113**].

This paradigm leverages a combination of semaphoric gestures, continuous gestures, and tactile feedback (Figure III.6). We first defined a segmentation hand posture (an open hand) to summon the GUI elements. Then we defined a different hand posture for different kinds of widgets: buttons, sliders, knobs, switches, spinboxes, and paired buttons. It is of course possible to add other kinds of widgets with other hand postures. When the users perform one of these postures, they can select one of the widgets of this type. They receive a 150 ms/350 Hz vibration pulse to confirm the selection, and the currently selected widget is visually highlighted. If the GUI has several widgets of this type, the users can disambiguate with a continuous movement. Then they perform a gesture for manipulating the widget and receive immediate haptic feedback with continuous vibrations on the thumb and index finger. For example, they can pinch and drag to move a slider knob. They can release the knob by releasing the pinch and release the slider by performing the segmentation gesture again.

We conducted user studies to measure the benefits of this new paradigm. In the first one, we showed that this paradigm avoids the Midas touch issues, and we compared two disambiguation mechanisms. In the second study, we showed that this paradigm has quantitative and qualitative benefits compared to midair pointing. Despite these benefits, this new paradigm has challenges that are still to be addressed. In particular, it relies on semaphoric gestures that users have to know. It contradicts Nielsen's *recognition rather than recall* heuristic [179, 180], which is one of the essential benefits of the point & select paradigm. Therefore, we still have to evaluate the discoverability and learnability of the gestures and improve them if necessary [58]. We can for example encourage learnability and discoverability with feedforward visual cues in the vicinity of the widgets [163].

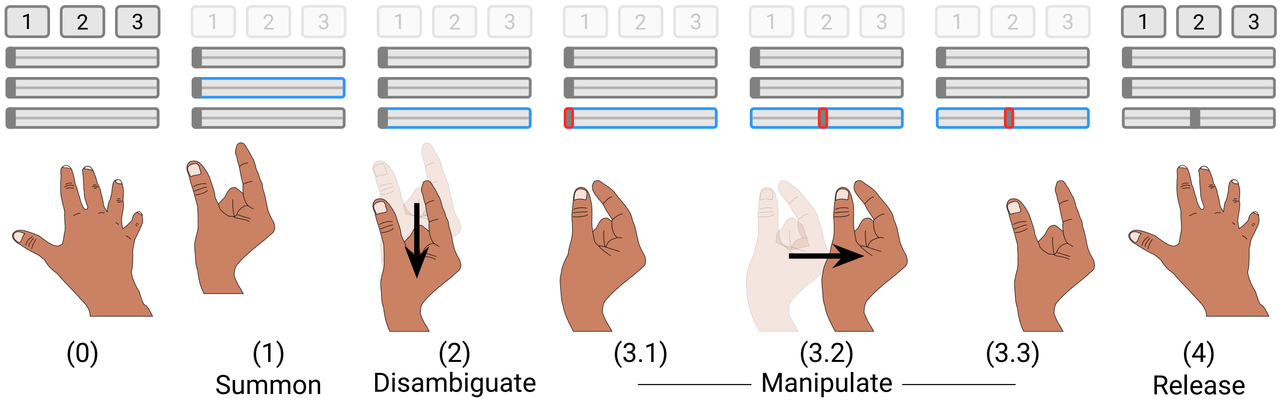


Figure III.6 – Steps of summon & select for the bottom slider. (0) Idle (1) Summoning posture for slider (2) Disambiguating by zoning to the desired slider (blue focus moves to the bottom slider) (3.1-3.3) Manipulation: (3.1) Enter Drag posture to enter dragging mode (red box around the bar) (3.2) Dragging the slider bar (3.3) Exit Drag posture to exit dragging mode (4) Release posture to release the control.

2.1.2 Haptic Direct Manipulation

We discussed the concept of direct manipulation in Section III.3.2 [232]. It is one of the most important concepts of GUIs. Its properties provide valuable usability benefits that highly contributed to the success of GUIs over command line interfaces. Yet, this paradigm was tailored for visual interfaces. The question of whether this concept could be used or adapted to tactile display was open. Therefore we studied the adaptation of direct manipulation to tactile displays [**111**, **112**, **203**].

The most challenging direct manipulation property for tactile display is certainly the one stating that objects of interest have to be visible. Contrary to vision, it is difficult to perceive an overview of the environment at a glance with the sense of touch. Vision is particularly efficient at glancing not only because of the high density and sensitivity of photoreceptor cells in the retina but also because of the high mobility of the eyes and the ability of the brain to process this sensorimotor loop. Therefore, we leveraged the sensorimotor loop with the sense of touch and the motor ability to make objects of interest *touchable* and *explorable*.

In this new paradigm, the users can control a pointer that they can move continuously with gestures and perceive with tactile feedback. When the cursor moves, it feels like a vibration moving continuously on the skin. This property makes the tactile space explorable. Then, the sensation is different when the cursor hovers over a target or moves over the background, which makes objects touchable. Input modifiers such as the number of contact points or the number of contact repetitions are used to switch between the *idle*, *tracking*, and *dragging* stages [45]. With these interactions we can implement fundamental direct manipulation interaction techniques such as *pointing*, *selection*, and *manipulation*.

We implemented this paradigm with a proof of concept 1D 360° tactile display around the wrist (Figure III.7). We used the prototype described in Section III.1, which has four EAI C2 tactors. The continuously moving cursor is implemented with the funelling illusion I described in Chapter I and illustrated on Figure I.2. The display is divided into four quarters in between the actuators. The cursor is a phantom sensation created by interpolating the signal amplitude of the two edge actuators of the corresponding quarter. Targets are represented with a 250 Hz frequency and the background with 100 Hz. Not only it is an easily distinguishable vibration, but the 100 Hz is subtle and avoids or at least reduces numbness. The inputs use a multitouch smartwatch. Up and down swipes move the cursor in either direction. The tracking states trigger with one contact point and the dragging state triggers with two contact points. The cursor is not felt in the idle state, to avoid numbness. The details about feedback and states machines are presented in the paper [**112**].

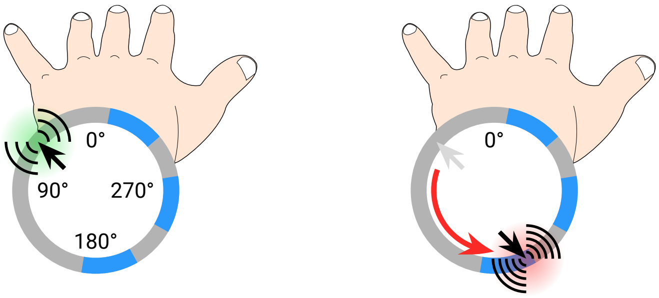


Figure III.7 – A 1D circular 360° tactile display around the wrist. On left, the tactile pointer is over void and the tactile response frequency is 100 Hz. On right, the user navigates the pointer to a target, where the tactile response frequency is 250 Hz.

We validated the concept with user evaluations of the proof of concept prototype. First, we validated that users are able to navigate and distinguish targets with a JND experiment on the maximum number of targets they were able to count. On average, participants were able to count up to 19 targets. Then we evaluated the pointing performance and confirmed it globally follows Fitts' law. We note however participants made faster selections when the targets were exactly over an actuator position compared to in-between. We proposed a refined pointing model that takes into account this observation. Finally, we designed two tactile menus with 4 and 8 items and we showed that participants were fast and accurate.

2.2 The sense of embodiment in Virtual Reality

We discussed in Section III.1.2 that haptic feedback has qualitative benefits, in particular when it restores haptic sensations that are non-existent or limited in gestural or multi-touch interaction. This is especially important in *virtual environments* in which we would like to immerse users. Slater defined *immersion* as "the extent to which the actual system delivers a surrounding environment" [236]. Therefore, this notion refers to technological aspects that contribute to immersing the user in the virtual environment. *Presence* is rather the subjective feeling of being inside a virtual environment [234, 235]. Witmer and Singer proposed a questionnaire to measure users' presence in a virtual environment [265]. They identified four factors that influence the feeling of presence: the ability to *control* objects, *sensory* stimulation, *distraction* from the real world, and *realism*. We used this questionnaire in the study presented in Section III.1.2 and observed that haptic feedback improved presence, in particular sensory and realism factors.

The users are generally represented in virtual environments with an *avatar*. This avatar usually has a visual representation, which is not necessarily realistic or even human [190]. The users explore the virtual environment through this avatar. They can also perform operations that are impossible in the physical world, like telekinesis or teleportation. In fact, the appearance or behavior of the avatar has an influence on the way the users behave in the virtual environment. For example, the *Proteus effect* describes the way the visual representation of an avatar influences the behavior of the users that control it [267]. On the opposite, visuotactile stimulation can lead people to consider a rubber hand as part of their body [34], or that they have a sixth finger on their hand [125]. These effects are examples of extensions of the *sense of embodiment* of a virtual body [135]. Kilteni *et al.* discuss three subcomponents of the sense of embodiment that were extensively studied in the literature [135]. *Self-location* refers to the "volume in space where one feels to be located". *Agency* refers to "the sense of having global motor control" over the virtual body. And *ownership* refers to "one's self-attribution of a body".

We studied this notion of avatar embodiment within the Avatar project funded by Inria. These studies are part of Grégoire Richard's Ph.D. thesis, which I co-supervise with Géry Casiez, Anatole Lécuyer, and Ferran Argelaguet.

2.2.1 Methodologies for measuring the sense of embodiment

There is a number of questionnaires in the literature to measure the sense of embodiment. We discuss some of them in one of our studies [**216**]. There are recent attempts to standardize these questionnaires. For example Roth *et al.* propose a questionnaire with subcomponents: *ownership*, *agency*, and perceived change in the *body schema* [219]. The latter notion is larger than *self-location* as it refers to any difference the users may perceive between their own body and the avatar. Gonzalez Franco and Peck proposed another questionnaire in which they added to Kilteni *et al.*'s subcomponents : *tactile sensations*, *external appearance*, and *response to external stimuli* [101]. They later improved and simplified their questionnaire, and evaluated it with many different tasks [194]. The subcomponents of this new questionnaire are: *appearance*, *response*, *ownership*, and *multi-sensory* [194]. Interestingly, *agency* is not an identified subcomponent but rather distributed among the others, in particular to the *response* subcomponent.

These questionnaires are typically used in controlled experiments after the participants performed a specific task in a virtual environment. We compare the overall embodiment and its subcomponents score in two or more conditions to identify the effects of these conditions. The experimental protocol we can use depends on the task. For example, some studies use a threat like a virtual fire or sharp blade as an objective measurement of embodiment [8, 68]. Subjects are considered embodied if they attempt to avoid the threat despite its virtual nature. The issue is that this kind of metric requires participants to be surprised by the threat. However, this cannot be guaranteed with a *within-subjects design* in which participants perform all the conditions one after the other. In such situations, the experiment must follow a *between-subjects design*, in which separate groups of participants perform a different condition.

User study

In a between-subjects study, participants are assigned to one of the conditions. There is therefore potentially a bias if the groups are not well balanced. We investigated this effect on embodiment studies [**216**]. We experimented a visuomotor task with a synchronous condition and an asynchronous condition with a latency of 300 ms between the inputs and output response. This value is known to have a medium effect on embodiment in the literature [34, 135, 140]. We chose a simple experimental task that requires no special equipment to facilitate replication. Participants were seated on a chair, with their legs on a table, and had to perform gestures with their feet (Figure III.8), similarly to [140]. 92 participants performed this task in a balanced within-subjects design. To study the effect of the sample size and its effect on the statistical analysis we analyzed random data subsets of 10 to 92 participants. To study the effect of the experiment design we simulated between-subjects designs by selecting the first condition every participant made. We considered the analysis of all participants with the within-subjects design as the ground truth. Similarly to the literature this analysis shows that latency reduces the sense of embodiment [34, 135, 140].

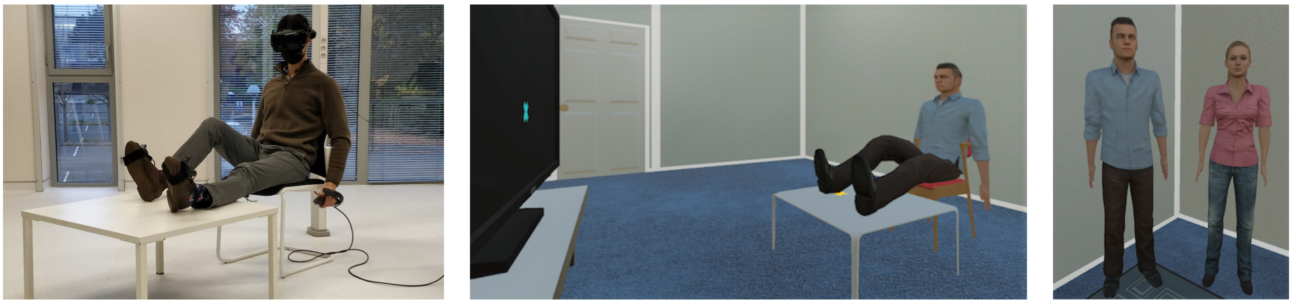


Figure III.8 – The user seated on a chair, performing leg movements, the virtual environment, and the two avatars.

Our results showed that all the random subsets with at least 40 participants with the within-subjects design gave the same result as the ground truth. However, regardless of the number of participants, we did not observe the ground truth effect with the between-subject analyses. Based on the debriefing with participants, our main explanation of this phenomenon is that participants needed a reference to provide a meaningful answer for each question. Therefore they calibrated their answers to the second condition relatively to the first one. Hence, we could not measure the effect with the first condition only. We discuss recommendations and possible mitigation strategies in the paper [**216**]. Interestingly, when we analyzed the second condition as a kind of calibrated between-subjects design we observed the ground truth effect. However, the effect size was about half the effect size of the within-subject analysis. Therefore, we wonder if both designs even measured the same phenomenon. We are still working on this subject, in particular to provide calibration methods and metrics to balance groups for between-subjects design in embodiment studies.

2.2.2 *Haptics and the sense of embodiment*

The study of the causes and effects of the sense of embodiment of an avatar in virtual reality is a hot topic in the Virtual Reality community. Interestingly, all the embodiment questionnaires such as those we discussed before have subcomponents related to the sensorimotor loop. It means that the sensorimotor loop is essential to the sense of embodiment. For example, people have a stronger sense of ownership when they perform actions with a visually realistic hand, and a stronger sense of agency when they embody an abstract-looking virtual hand [8]. Following this idea, we studied the effect of haptics on the sense of embodiment.

We performed a user study to compare embodiment for a drawing task with force feedback, tactile feedback, and a control condition with no haptic feedback [**215**]. The participants were seated on a chair, and they had to paint a mandala in an immersive virtual environment with a Phantom Desktop[7] device (Figure III.9). In the force feedback condition, they felt the surface resistance of hard objects and the viscosity of the paint spheres at the bottom. In the tactile condition, they felt a 250 Hz vibration whose amplitude was proportional to the interpenetration distance to the canvas surface. We attached an EAI C2 tactor to vibrate the Phantom stylus (Figure III.9). In the control condition, the Phantom was only used as an input device, with no force or vibration. We mesured embodiment with Gonzalez Franco and Peck's first standardized questionnaire[8] with the *agency*, *self location*, *ownership*, and *tactile sensations* subcomponents [101].

We observed a stronger embodiment in the force feedback condition compared to the control condition. In particular, participants had a higher sense of ownership. However, we did not observe these differences between the tactile and control conditions. Besides the detailed discussion in the paper, it is important to note that in some ways this task favored the force feedback condition over the tactile condition. Participants certainly expected to feel the stiffness of hard surfaces. Similarly to realistic

---

[7] Today called Touch X by 3D Systems https://www.3dsystems.com/haptics-devices/touch-x

[8] The second one was not published at the time.


Figure III.9 – Haptic device setup, virtual environment and task of the virtual embodiment study.

visual feedback [8], this realistic force feedback aspect reinforced the sense of ownership. On the contrary the vibrotactile feedback was symbolic because participants only received tactile guidance. And we did not observe any improvement in embodiment. It does not necessarily mean that the sense of embodiment requires realistic haptic feedback. For example, non-realistic visual feedback improved the sense of agency [8]. But in our task force feedback *constrained* the stylus tip movement to prevent it from getting through the surface, while vibrotactile feedback only *guided* it. Therefore I believe the force feedback condition helped participants focus on the painting task rather than controlling the stylus to paint the canvas , which reinforced sensorimotor integration. The workload analysis discussed in the paper gives supports this explanation. Further studies should investigate other tasks or a variation of this one in which vibrotactile feedback promotes sensorimotor integration.

3 Computing and the sensorimotor loop

In the previous chapters, we discussed several examples of how haptics as the sense of touch on one side, and haptics as the motor ability on the other side provide useful interactive properties. In the previous section, we discussed the fact that the benefits of one of them cannot necessarily compensate for the issues of the other. This observation is however only the tip of the iceberg. It is one of the many clues showing that inputs and outputs cannot be separated, the same way that our senses and abilities cannot be separated either.

One of the major differences between humans and systems is that humans are as they are, we cannot fundamentally change the way they function. They can learn and get experience, but we cannot give people new senses and abilities. Systems are different: we build them, therefore we can design and build them according to our needs. We have no reason to build a machine that has no purpose for us. We discuss below how the literature modeled the way humans work, and models for designing systems. Interestingly, both work in a similar way, with inputs, processing, and outputs. I will discuss this observation, my vision of how interactive systems work, and the critical role of the sensorimotor loop.

3.1 Human behavior

The understanding of human behavior, and in particular the way humans perceive the world and act on it is the foundation of HCI. Gibson's work on this topic was pivotal to the development of the HCI community. He studied for decades how we, as animals, perceive our environment [92]. His work mainly focused on visual perception, but he also studied other senses like the sense of touch [91]. The idea is that the perception we have of the environment is not just based on a bunch of input streams from our senses. It is the integration of these input streams with the exploratory movement that we made, with our memories and experience. This is because the input streams depend on the exploratory movements, and we make these exploratory movements to seek information and match it with our previous knowledge to shape our perception.

The nature of the movement itself, that Lederman and Klatzky call exploratory procedures [144, 145], enables people to sense different properties of objects. For example, we can feel the texture of an object with a lateral motion, and we can feel its hardness with a pressure. When we enclose an object with our hands we can sense its volume and global shape, but if we follow its contours with our fingers we can not only sense its global but also exact shape. This intertwined relation of sensations and active exploration in our perception enables outstanding paradigms. For example, sensory substitution consists in translating information that is typically sensed with one sense to another sense [9]. Several sensory substitution devices use the sense of touch with active exploration gestures to replace vision [30, 59, 85, 155]. They typically enable blind people to read printed books, or scan their environment with a haptic white cane. Without active exploration, our brain cannot process such a complex tactile input stream.

Affordance

Gibson's theory involves a close relationship between the animal (humans in our case) and its environment. Gibson makes a difference between the physical world and the environment. The physical world refers to everything from the smallest particles to the biggest possible objects like galaxies. The environment refers to what is reachable for the animal, in particular in terms of size. This study on an animal in its environment is what Gibson calls the *ecological approach of perception*. It makes it possible to study the relations between animals and their environment. In particular, specific combinations of properties of an animal and objects of its environment enable the animal to perform particular actions on the object. Gibson called these relations *affordance* [93]. A handle of a given size and a hand of a similar *affords* grasping. A flat horizontal surface of the size of the pelvis *affords* sitting.

These notions were not initially studied to improve the design of interactive systems. However, HCI researchers and interaction designers applied these principles for the study and design of interactive systems. In particular, Norman's work has a strong influence on the HCI community. His theory of action depicted on Figure III.10 is an operational view of Gibson's ecological approach to perception [187]. It describes seven stages to describe how people interact with objects in their environment, based on their *mental model* how this object works. There is a central stage representing the goal the person would like to reach, and three stages for the execution of actions and the evaluation of the changes in the world. Norman uses this model to explain the many usability issues that can arise when this mental model differs from the *conceptual model* of how this object actually works.

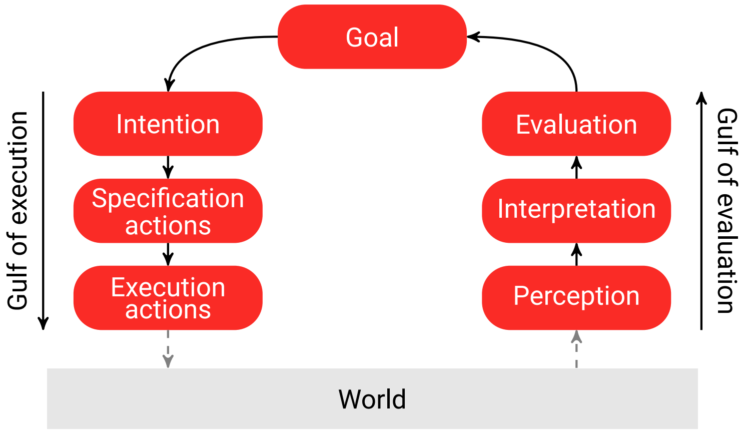


Figure III.10 – Norman's seven stages of action [185]. It describes how people interact with their environment.

Norman also participated in the introduction of the concept of affordance to the HCI community. Contrary to objects found in nature, human-made objects can leverage our knowledge about human characteristics. We can reduce the gaps between the mental model and conceptual model by creating affordances. Norman gives the example of doors designed in a way we know which side to pull and which side to push [187]. The pull side has a handle that affords to grasp and pull whereas the push side has a flat plate that affords pushing. However, we cannot control all the physical properties of all the objects around us. Therefore there can be a difference between the actual affordances and the affordances we perceive. Gaver describes four possibilities [87]. Two of them are desired: a perceived affordance and a true reject (there is no affordance, and no affordance is perceived). He also describes hidden affordances and false affordances. Because of this, Norman now makes a distinction between an affordance, as a property, and a *signifier* which is a perceivable property that advertises the existence of an affordance [185].

Other practical models in HCI literature guide the design of interaction techniques and interactive devices. One of the most universal mathematical models in interaction is certainly *Fitts' law* that measures the difficulty (ID) of a target selection movement $ID = \log_2\left(\frac{2D}{W}\right)$ [79]. The model says that the longer the distance and the smallest the target width, the higher the difficulty. With the original experiment protocol, participants perform a reciprocal left-to-right movement between two targets of width W and distance D (also called movement amplitude). The reciprocal aspect of the task reduces visual search, hence the influence of perception. Therefore this is essentially a measure of motor performance. This is an active research topic with new contributions every year for decades. HCI research usually uses MacKenzie's throughput-based formulation [158], and experimental protocols were adapted to 2D [159], 3D [172], and similar tasks like steering [2]. Other researchers propose alternative interpretations, as a time/error trade-off for example [108].

Besides motor-behavior models, we discussed perceptual empirical evaluations and models in Chapter I. *GOMS* models are examples of human behavior model that takes into account both perception and actions. They model humans with three components: a perceptual system, a motor system, and a cognitive system [47]. These are therefore more generic than models such as Fitts', they model a greater diversity of behavior. In particular, the Keystroke-Level Model (*KLM*) defines several types of operations from mental activities to key presses [46]. Typically it leverages models such as Fitts' to quantify pointing operations. With these models, we can describe interaction techniques as sequences of atomic operations, and predict average performance based on empirically-defined rules.

The models we discussed take into account human behavior, and to some extent the way interaction techniques work, but not necessarily how they are implemented. For example, they do not take into account the transfer function between the pointing device and the cursor. They do not necessarily take into account the integration or separation of degrees of freedom [160] or the type of feedforward [255]. In my opinion, this is a limitation of the generative aspect of these models, and I believe we must include more knowledge about the implementation into interaction models and conversely. This is one of the objectives of the Loki project team.

3.2 System architectures and paradigms

Studying human behavior is useful for the design of machines for several reasons. The first reason is to reproduce the strengths of humans. For example, in the previous section, we discussed the fundamental coupling between humans' perception and action. Systems called *closed-loop systems*[9] also leverage such a mechanism [25]. For example, the non-inverting and inverting amplifiers circuit depicted on Figure III.11 have the output of their operational amplifier connected to one of their input through a resistor. As a result, the output voltage is proportional to the input voltage whose value depends on R_1 and R_2. But most importantly, the feedback loop stabilizes the output voltage to the desired value. This kind of control mechanism is used in many applications such as robots, domestic appliances, or drones. It is also used in haptic devices that leverage information from people with *Human-in-the-loop* models [253]. However, this paradigm focuses on system control. The human only exists as a parameter of the equation. Therefore despite the similarities with the way we describe human behavior, this is not our focus.

The second reason to study human behavior is to improve the interaction between humans and machines. Humans and interactive systems are distinct entities that need each other and must communicate to achieve their objectives. Hence we will discuss below the architecture of interactive systems, the similarities and differences with humans, and how this is critical for improving interactions.

---

[9] https://en.wikipedia.org/wiki/Control_system

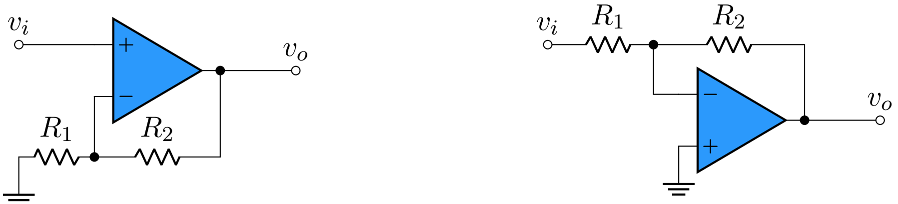


Figure III.11 – Non-inverting amplifier and inverting amplifier. They both use their inputs and their own output to compute their output.

Computation models

Initially, computers and programs were essentially based on theoretical models such as *λ-calculus* [56] or *Turing machines* [252]. These are computing models, and they focus on solving numerical problems rather than helping people with their everyday activities. A Turing machine has an infinite tape with symbols written in advance, and a pre-defined transition table that describes the behavior of the machine. Therefore these machines ignore their environment, in which anything can change at any time. All these models are equivalent (or Turing-equivalent), and the *Church-Turing thesis* says that everything these models can compute can be implemented with an algorithm. Wegner and Goldin explain that both this universality and this limitation are due to their inductive nature [260].

Induction requires structures to be finite, and computation to end. For example, the Listing III.1 shows the inductive definition of a list of numbers and a function that computes the length of a list. A list is built with two constructors: either a `Nil` value for an empty list or a `Cons` function that create a list with a number (the head) and another list (the tail). The `Nil` value ensures that the list is finite. It ensures in turn that the `length` function ends because there is no recursive call on the base case (`Nil`) and every list ends with `Nil`.

```
Inductive list : Set :=
| Nil : list
| Cons : nat → list → list.

Fixpoint length (l: list) : nat :=
  match l with
  | Nil ⇒ 0
  | Cons _ s' ⇒ 1 + (length s')
  end.
```

Listing III.1 – Inductive list and example of inductive function on a list.

This simple example shows the rigidity of algorithmic behavior. All information about the problem must be known in advance, the computing process is precisely defined, and the output is specified by the inputs [139]. Wegner and Goldin describe interaction as a more general model in which the machine is connected to input streams, that provide unpredictable data [100]. They discuss interaction between systems, and Beaudouin-Lafon explains how their concept also applies to HCI [23]. They model interaction with co-induction as in the example in Listing III.2. This example defines a stream of numbers, which is a non-finite structure, and a function that returns another stream in which numbers from the input stream are multiplied by 2. The co-inductive definition of the stream only has one constructor, identical to the second constructor of lists. The absence of a base constructor makes streams infinite structures. Therefore recursive functions on streams potentially run forever. Hence, there is no equivalent of a `length` function because it would never return a value. However, functions such as in this example still make sense. They operate iteratively rather than globally, therefore they can operate on infinite structures.

```
CoInductive stream : Set :=
Cons : nat → stream → stream.

CoFixpoint double (s: stream) : stream :=
match s with
| Cons x s' ⇒ Cons (2 * x) (double s')
end.
```

Listing III.2 – Co-inductive stream and example of co-inductive function on a stream.

We observe here that each iteration of the co-fixpoint can be inductive, as in the example. It shows that interaction is a general process that connects entities in the environment to enable them to exchange information. Algorithms only process information to transform known input into outputs without knowledge of the overall scheme, and no external event can change their behavior during their execution. Interactive systems we use every day react to unpredictable inputs in real-time. Therefore they are not just built with algorithms. They have separated input and output loops at different levels that communicate through streams. For example, on the system level, an input loop gets input streams of input data (*e.g.* mouse displacements) and produces output streams of input events (*e.g.* mouse move event). On the application level, an input loop gets output streams of input events and combines the information they convey with interaction techniques to produce an output stream of actions to be executed. On the application level, a graphics loop gets an input stream of graphic commands and produces an output stream of objects to be displayed. Therefore, applications are what Wegner calls *interaction machines* [259].

Software architectures and interaction paradigms

Software architectures leverage this interaction machine to describe a higher-level structure that connects users to a functional *model*. This model, also called an *abstraction*, defines the objects of the system, their properties, and the operations on them. For example, the original *MVC* architectures distinguish the model with *views* that describe how objects are presented to users and *controllers* that define the way users can manipulate them [211, 212]. *Arch* [19] and *PAC* [62] rather combine inputs and outputs as a *presentation* component, and add a *controler* component that manages transitions between abstract inputs/outputs and domain-specific properties of the model/abstraction. The advantage of these architectures is to separate the objects of interest from the interaction with them. It is therefore easier to display several synchronized representations of the same object and provide multiple ways to manipulate them. These interactive properties contribute to leveraging human capacities and flexibility through *multimodality* [181, 183].

In the previous section, we discussed the critical role of the sensorimotor cycle on human perception. The software architectures above create connections between humans and interactive systems with input and output streams. However, leveraging the full potential of the sensorimotor loop requires an additional layer of interaction paradigm such as *direct manipulation* [232]. The direct manipulation paradigm defines several properties: objects have to be visible and directly manipulable, and actions have to be fast, reversible, and incremental. Implementing these properties into the design of graphical user interfaces (*GUI*) favors their usability. For example it contributes to some of Nielsen's heuristics [179, 180]: *visibility of system status*, *user control and freedom*, *recognition rather than recall*, and *Flexibility and efficiency of use*. The *instrumental interaction* paradigm extends direct manipulation and makes the connection with software achitectures [22]. Similar to software architectures, it defines domain objects. Users can interact with them through *interaction instruments*, which are reifications of commands. When users perform actions on the instruments, they receive immediate feedback and the instrument performs operations on the domain objects. The domain object returns a response to this operation. The combination of the action and both the feedback and response forms a sensorimotor cycle that enables direct manipulation and leverages the users' perceptual skills.

4 Ecological approach to computing

Humans and interactive systems share the same overall schema in the sense that they are both autonomous entities that interact with their environment through input and output streams (Figure III.12). People perceive and act on their environment thanks to their sensorimotor loop. Therefore our perception depends both on our sensorial and motor capabilities. They have limitations that influence the way we perceive our environment. There are *type* limitations. For example, we can perceive light waves but not magnetic fields. There are *range* limitations. For example, we hear a sound wave of 400 Hz, but not 400 kHz. We can reach objects 50 cm away from us, but not 50 m. There is also a *precision* limitation, for example we can distinguish colors of wavelength 100 nm apart, but not 1 nm apart. Precision is actually a range of difference, therefore according to Weber's law, the threshold is proportional to the base stimulus value [76]. Finally, there are *processing* limitations, related to our cognitive abilities to interpret signals resulting from our perceptions and actions. Typically, illusions are distortions of what we could consider as ground truth. Interactive systems have the same kind of limitations. They are limited by the inputs and outputs they receive, and the computing capabilities to interpret them. Here inputs and outputs do not necessarily refer to bit streams, but more generally streams of physical phenomenons of their environment they can sense or produce (*e.g.* movements, sounds, light).

Humans and machines are autonomous entities but they are not independent. On one side, humans use tools, machines, and instruments to extend their limitations discussed above. Extending human

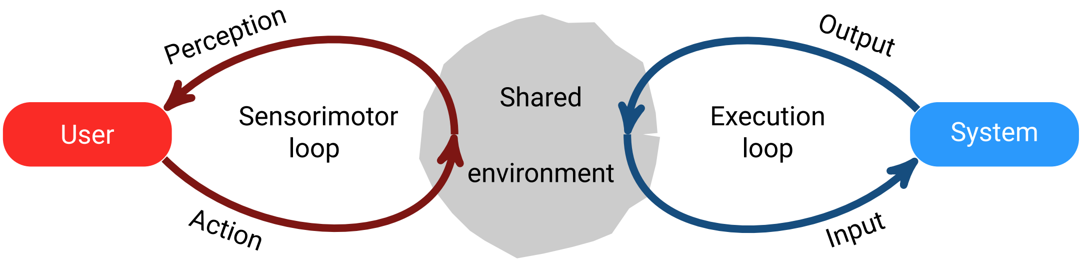


Figure III.12 – The user and the system interact together through a shared environment.

capacities with computers marked the beginning of Human-Computer Interaction, with the pioneer visions of Bush [44] and Engelbart [73]. On the other side, all machines need humans otherwise they have no purpose. They all need instructions and data, and they all modify the environment or produce information. These interactions between entities, whether they are humans or machines, require communication. What I mean by communication is to produce a physical effect on a shared environment that the other entity can perceive. This shared environment has a common space and time, at a similar scale. The characteristics of these communications depend on the skills and capabilities of both entities. For example, the most efficient communication between two humans is certainly speech. This is maybe the reason why there is a push toward vocal interfaces. However, so far technologies struggle with the interpretation of natural languages because of their lack of flexibility compared to a human mind. Vocal interfaces only recognize a limited and pre-defined set of instructions. But in addition to speech, we make gestures with the hands, face, or other body parts that can totally change the meaning. As we discussed in the previous chapter, systems are efficient at sensing gestures, and there is even still room for improvement. Therefore this is today the main communication modality between humans and machines.

4.1 Seven stages of reaction

In Section III.3.1 we discussed how people perceive their environment and in particular interactive systems. We presented how Norman's theory of action (see Figure III.10) explains the difference between the conceptual model of the system, and the perceptual model the users have of it based on their perception. It is known that the implementation of systems depends on ethnographic background of programmers [218], and the system architecture reproduces the organization structure that designed it [61]. Therefore, I suggest that interactive systems follow a similar perceptual scheme than the human perceptual scheme, and their interaction is an interactive loop in itself.

I illustrate this approach with an adaptation of Norman's theory of action, depicted on Figure III.13. The input chain begins with the *sensing* stage. Physical sensors measure physical effects in the environment. Typically they measure the user movements, but it can be various other information such as light, temperature, moisture, or vibrations. All such information is transformed into *input events*. At this stage, we notice that the infinite richness of the world is reduced to a small number of bits. For example, when a user presses a key on a keyboard, the interactive system only senses whether the key is pressed or not. There is usually no information about the finger that pressed it, the speed of the finger, or its trajectory. Input events have to be treated as tokens, or lexical units. They are interpreted as *input phrases* with grammars or finite automatons [5] and form the building blocks of *interaction techniques* [182]. The *software* part is an interaction machine that receives streams of input phrases, interprets them with algorithms, and eventually sends back information through output streams. First of all the systems must *encode* the pieces of information. A visual encoding can be an icon or a text. Audio encodings can be sounds [86] or melodies [39]. Various haptic encodings include vibrations [37], forces [162, **197**, **200**], or textures [**198**, **202**, **208**]. Output devices have driving electronics that require specific *commands* and turn them into *physical effects*. These are typically lights (like screens), sounds, vibrations, and forces. This model extends software models like PAC, MVC, or Arch because it describes the interactive system on three levels. Similarly to these architectures, it describes the application level with the input phrase, the program, and the encoding. But it also describes the system level, with the input event and the command creation. And it also describes the connection with the physical world with the sensing and physical effect stages.

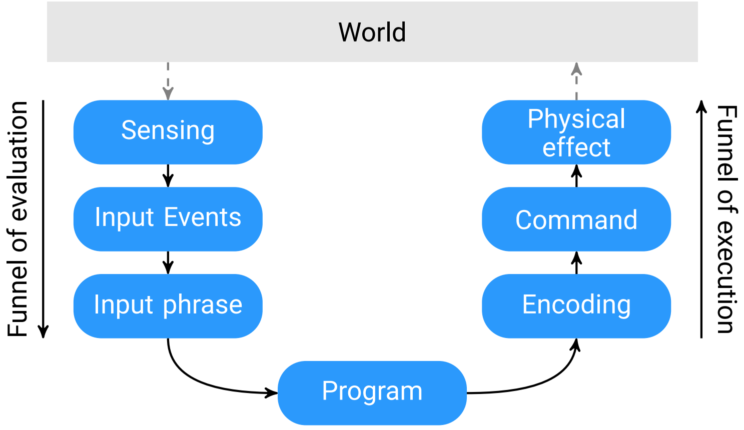


Figure III.13 – The seven stages of reaction, adapted from Norman's seven stages of action. It describes how interactive systems interact with their environment.

Funnel of evaluation

The funnel of evaluation designates the fact that the input stages reduce the complexity of the system's environment into a few bits. A good design of the input chain senses the right phenomenons, at an appropriate amplitude, with a sufficient spatial and temporal resolution, and with little distortions. This information must be combined correctly to form a meaningful sequence of actions that the users must perform. For example, the *Midas touch* problem is a usual issue with 3D gestural interaction. Since the sensors observe the users' movements without activation or interruption, there is no obvious segmentation. The system has no way to know if the users move their hand to interact with the system, or for scratching their nose for example. *Occlusion* has the opposite issue. The sensor cannot get position information for objects outside of its field of view. In these situations, we can grow the funnel of evaluation respectively by adding segmentation gestures (see Section III.2.1.1 below), and using multiple cameras.

Funnel of execution

The funnel of execution is symmetrical to the funnel of evaluation. The way the objects of the internal model are shown to the users can have a huge impact on how they interact with it [272]. Therefore, the encoding part is crucial, and it is the first filter for reducing the internal complexity of the system. The specification of the output device is a second filter. There is a limitation of physical effect (*e.g.* force, color, brightness, frequency) each device can produce in practice. There is also a limit of precision, that depends on electronics and mechanics. Last, the physical effect can be inconsistent for the same command. Some haptic devices can behave differently depending on ambient conditions (*e.g.* temperature, finger moisture, cleanliness).

4.2 The Human-System loop

Norman's theory of action is typically used to describe differences between the user's perceptual model and the system's conceptual model. The adaptation of this theory to systems describes a similar difference, but the conceptual and perceptual models are inverted. The systems' conceptual model of the user is based on the user's perceptual model of it that describes the way the users interact with the system. This is what Norman describes with his seven stages of action (Figure III.10). The systems' perceptual model of the user is based on its own conceptual model that describes its ability to interact with humans. This is what I describe with the seven stages of reaction above (Figure III.13). Usability issues occur when these behaviors cannot connect together. Norman's theory of action and the seven stages of reaction must not be two separate concepts, but a unified process that creates real-time loops between the human and the system. One of the interesting properties of such connections is their ability to connect and disconnect. Some of them are planned, when the users know the affordance, or perceive a signifier of this affordance. Others are serendipitous as the users explore the interactive system.

There is a fundamental difference between human and system behavior though. We design the system behavior thanks to our engineering skills and scientific knowledge. Therefore we control it and we can adapt it to our needs. We can add sensors and actuators, tune their sensitivity and operating range, and change their software. On the opposite, human behavior depends on cultural and social background, and experiences. We can only shape it through training and education. Moreover, we cannot change the human body to add another sense or adjust its sensitivity and range for example. This is why we need tools and instruments to extend our capacities beyond human capacities.

The Figure III.14 summarizes the difference between the two approaches we discussed in this manuscript. The outer circle shows the connection between the output path discussed in Chapter I detailed on Figure I.1, and the input path described in Chapter II detailed on Figure II.1. It is similar to Abowd and Beale's interaction framework [1]. The advantage of this approach is that it describes better the connections between the user and the system and takes both into account. The inner loop shows the connection between the human and the system through Norman's seven stages of action (Figure III.10), and the seven stages of reaction I described above (Figure III.13). The advantage of this approach is that it leverages the sensorimotor and execution loops through both perception (resp. input) and action (resp. output).

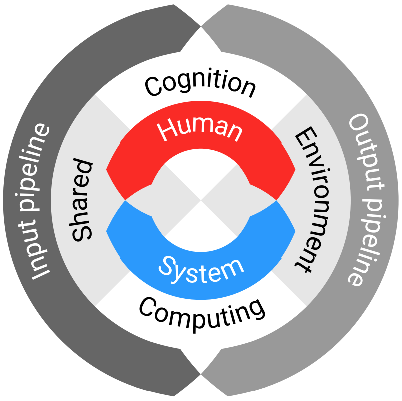


Figure III.14 – The Human-System loop. Both the human model and the system model are connected in a shared environment of the physical world.

We notice that the way I describe the system on Figure III.13, it reacts to the user's initiative, and Norman's model on Figure III.10 starts with the goal at the top. When we connect both, the cycle starts from the user, goes through the systems, and returns to the user. This is an example of Beaudouin-Lafon's *computer-as-a-tool* paradigm, which is the usual paradigm in HCI. The *computer-as-a-partner* paradigm would look like a symmetric model in which the system has an objective, and the user would react to the system's requests. This is basically the *human-in-the-loop* approach I mentioned above, favored in AI and automatics.

4.3 Discussion

The idea here is that interaction is not separate loops on the user and system sides. Once connected, these loops form a Human-System loop that cycles between the user and the system. Therefore, smooth interaction between a user and an interactive system requires efficient connections between them. Here we focus on computer-as-a-tool systems in which the users have the initiative. Hence, the system must not be a limiting factor to the user's sensorimotor loop, it must react in *real-time*. But the notion of real-time is context-dependent. The visual system requires a 100 Hz loop. The touch system requires a 1000 Hz loop. The audio system requires a 40 000 Hz loop. We know that breaking this loop affects perception. For example, a disruption of the human visual input stream prevents people from seeing changes in their visual field [214]. On the opposite, as we discussed in Chapter I, force-feedback devices use force models computed over 1000 Hz, but the force model to be applied can be updated at a lower frequency without affecting the perception of a shape. Hence, the fine characterization of the bottlenecks in this Human-System loop is necessary to avoid undesirable effects. But we can also leverage the limits of our sensorimotor loop to create illusions to extend interaction. More generally, combining the user and the system at the same level with complementary roles facilitates the holistic approach of my research.

The similarity of the user and the system in this model can give the wrong impression that humans and systems have the same capacities. Indeed, this is not the case. Humans are good at adapting to many situations, they have intuition, and they have high perceptual and motor skills. On the opposite, systems are good with repetitive tasks, precise calculations, and automation in general. In the Artificial Intelligence community, this is known as Moravec's paradox which states that contrary to general assumption, sensorimotor behavior is much more complex to implement than reasoning and computation [169]. This statement should not be surprising after reading this manuscript. My analysis of this paradox is that first, the assumption was a clear underestimation of the complexity of perception. Second, the inductive nature of systems and the co-inductive nature of humans are the core of this difference. Systems are made to execute algorithms with high efficiency. On the opposite, the human sensorimotor loop enables humans to explore, try, analyze, and therefore understand their environment. Research started to apply this principle of *curiosity* as a building block of a robot behavior[142]. Thanks to a kind of sensorimotor loop, a robotic arm can manipulate objects without explicit instructions on how to actually do that. Maybe if we manage to build systems that can adapt their behavior to the users, we will stop considering that humans must use their adaptation capacity to compensate for system failures.

5 Conclusion

In this chapter, I discussed the sensorimotor loop, how it is connected to many research domains, and aspects of the design, implementation, and evaluation of interactive systems. I discussed two examples of rejected submissions in which the work had the common fundamental flaw that we overlooked the sensorimotor loop. The reason why it happened is probably that my previous research that leveraged the sense of touch barely required inputs because they were essentially notification mechanisms. Later, I worked on the motor ability and visual feedback was sufficient. The interesting fact with these studies is that people who tried my 3D gestural interaction prototypes with haptic feedback appreciated this feedback, even if we did not measure quantitative benefits, and barely observed qualitative benefits.

The issue with the initial quantitative study was basically that the gestural interaction we used for target selection was too inefficient. The dwell time to activate the button was 3 s while the mean selection time was 3.5 s. This means that 85 % of the selection time is inevitable. The 3 s correspond to replications of selection in Kinect applications. There is certainly a slight progression margin, but nothing close to fast selection like with a mouse or a touchpad. Therefore, we proposed a new interaction paradigm for 3D gestural interaction that replaces point & select by summon & select. This new paradigm avoids the typical issues of gestural interaction: lack of segmentation, Midas touch, and tracking in general. However, these advantages are balanced with new potential issues. This paradigm relies on semaphoric postures and gestures that users have to know beforehand. Therefore there are potentially guessability and learnability issues. We did not study this aspect, but it has to be addressed or at least considered before implementation in an actual interactive system.

The initial qualitative study showed an increase of presence thanks to tactile feedback, in particular for the *sensory* and *realism* factors. Some of the controls remained difficult though. The task consisted in driving a racing car with the hands in the air as if participants were holding a steering wheel. However, a physical steering wheel has many advantages in addition to precise and reliable steering. In particular, users can rest their arms on the steering wheel. But the difficulty with the input mapping is that both the steering and acceleration/braking degrees of freedom were integrated. It made it difficult to brake and steer at the same time, which is a common task when driving. Our next studies on the qualitative aspect of haptic feedback focused on the sense of embodiment in immersive virtual reality. On one hand, we showed that participant answered embodiment questionnaires relatively, which mean they either need calibration or we must use a within-subjects design. On the other hand, we showed that force feedback has an effect on embodiment for a painting task. We did not observe any effect of tactile feedback on embodiment. We explained this difference by the fact that force feedback prevented interpenetration, therefore participants could focus on the task. We must investigate other tasks in which tactile feedback contributes to sensorimotor integration.

The interaction models I described mostly correspond to current good practices. Therefore I tend to claim they have descriptive properties. But at this stage, I am still uncertain of their generative properties. These are the results of years of thinking about what started as an attempt to make a bridge between HCI and theoretical computing. Theoretical computing, and in particular logic was a large part of my educational background. It significantly shaped my vision of science at the beginning of my career. When I started thinking about this document years ago I struggled with avoiding crossing the boundaries of my expertise in these topics. There is therefore much more to say on the connections between interaction and computing, but with less confidence about relevance. This is also the reason why I needed to make the connection between interaction and design models, and shape my own vision of HCI research that I put into practice in my research.

Conclusion

> There are no happy endings in history, only crisis points that pass.
>
> Isaac Asimov

This document presents the evolution of my vision of Human-Computer Interaction as I practice it in my research. The first two chapters describe my early vision that either focuses on output and the sense of touch, or input and the motor ability. The third chapter describes the limits of this approach, and how the sensorimotor loop combines these two approaches to form a whole that worth more than its parts. However, each chapter presents useful lessons learned from years of research.

Chapter I was the connection with my Ph.D., which essentially focused on haptic feedback. I used haptics essentially to encode information with Tactons that users mostly received as notifications. My work focused on the encoding for the system side and interpretation on the human side. In this chapter, I presented work in which I studied other parts of the pipeline, in particular the device level on the system side, and psychophysics on the human side. My vision of this work evolved thanks to collaborations with researchers of other domains, like cognitive sciences, electronics, and material sciences.

After working almost exclusively on haptic output, I got interested in input techniques as well. The Chapter II describes the symmetry between input and output systems. The difference is who between the user and the system stimulates the other. Interestingly, while systems have many ways to stimulate human senses (*e.g.* visual, audio, or haptic cues), humans essentially stimulate systems with gestures. However, the richness of the gestures we can perform and their properties we can measure is an endless source of input solutions.

In Chapter III, I wanted to describe how thinking with the sensorimotor loop in mind changed the way I approach HCI research. I described the foundations of the field and their connections to many other research domains. Of course, each of these notions was individually studied for decades since their discovery, and they are still studied and extended today. But instead of focusing on a precise notion, I rather wanted to take this opportunity to look at the whole picture. It felt like a giant mind puzzle because all these notions are connected. It reflects the systemic and interdisciplinary approach in my research and life in general.

Interdisciplinarity is certainly the most exciting aspect of Human-Computer Interaction to me. The design, implementation, and evaluation of interactive systems require knowledge of the users, their biology, their cognition, their behavior, and knowledge about technologies, their mechanics, electronics, and software. Many researchers are highly specialized in one of these aspects. I rather prefer a systematic approach that consists in studying all the angles of a research problem. It is a difficult practice because each research community has its own literature, methodologies, and practices. But it is also an infinite source of ideas to push my research in new directions.

Future work

In Chapter III I described interaction with three components: users and their sensorimotor loop, systems and their execution loop, and a shared environment. Each of these components is a source of research questions, as they are potential bottlenecks of the Human-System loop. This integrated view helps me to look at interaction as the whole, with complementarity between users and systems, between inputs and outputs, and between the physical and software worlds. My research follows principles that I will refine in the next years.

First I like to propose simple yet expressive solutions. The simplicity can wear different forms. It can be simple from an engineering point of view, like the vibrotactile devices discussed in Section III.1. It can be simple interactions that increase the input or output vocabulary, like Métamorphe in Section I.2.4. This principle follows Kay's famous quote: *"Simple things should be simple, complex things should be possible"*[10].

Second, the digital world is an occasion to perform actions that are impossible in the physical world. Therefore I refrain as much as possible from the necessity to reproduce the physical world in the digital world. By this, I do not mean to ignore Nielsen's "Match between system and the real world" heuristic [180]. I mean that simulating the physical world is difficult, and it is difficult to validate that the replicate has the same properties as the original one. Moreover, slight discrepancies can trigger an uncanny valley effect [170]. On the opposite, artificial effects and objects encourage my simplicity principle above and are a source of expressivity. Graphical widgets on our interactive devices are artificial, they are not meant to be realistic. And we are fine with it. Conversely, the physical world enables interactions that are more difficult to implement and perform in the digital world. This is especially true for object manipulation because we can leverage all the degrees of freedom of our hands and fingers. We leverage the full capacities of our sense of touch. Therefore, I am interested in developing more work on tangible interaction [128].

Third, my background is in Computer Science and I have studied theoretical Computer Science topics like logic and formal systems for years. I always believed I could make connections between these approaches and Human-Computer Interaction. The Human-System loop is an early vision that positions them. In my vision, computation is one of the components of the loop, and it is at times one

---

[10] https://tiny.one/kayquote

of the bottlenecks we should study to improve interaction. The rigidity of the algorithmic approach is today compensated by the promising flexibility of machine learning mechanisms. The idea here is not to replace humans, or even perform activities on their behalf as in the computer-as-a-partner paradigm. The idea here is to leverage mechanisms such as reinforcement learning to break the rigidity of traditional computer programs. I believe it will enable them to probe their environment, try, evolve to avoid making repeatedly the same mistakes, or even avoid Gödel's incompleteness phenomenon by developing new features and capabilities by themselves [95]. I believe we can leverage a proper combination of rational and empirical methods to improve interaction.

So far, my research did not focus on any category of users, technology, or interaction context. I rather had a horizontal approach consisting in searching for appropriate solutions to any problem for any user. In my medium-term future research, I would like to focus on particular contexts, in order to favor incremental research and facilitate transfer of research results to the industry as I will continue to shape my vision of HCI.

Interaction in immersive virtual environments

There is an increased interest in immersive virtual environments over the past few years. The idea of metaverses was already there decades ago in popular culture [57, 94, 242]. The technology also emerged decades ago [244, 245]. However, like every technology, it required time and a lot of research to become mainstream. The availability of consumer-grade VR headsets over the last few years fostered research on interaction in immersive virtual environments. I described the contributions we made on this topic in Chapter II and Chapter III. It is however still unclear what metaverses will be like [110]. Some see it as a shopping mall, others as a social network, a gaming platform, or a classroom. Whatever it will be, immersive virtual environments challenge our senses and abilities. Without the constraints of the physical world, there are many new possibilities to leverage the sensorimotor loop.

I will continue the work on the sense of embodiment. On the methodological side, there is still work to improve measures of embodiment, whether with qualitative metrics like in Section III.2.2 or quantitative metrics with sensors. In addition to user evaluations, quantitative measures of embodiment could be used as inputs. It would be interesting to study the kind of sensorimotor loop we can create with this. More generally, I am interested in any method that leverages the sensorimotor loop. For example, I am part of a consortium that aims at leveraging microgestures in particular for virtual and augmented reality. In this project, I am interested in the combination of microgestures and haptic sensations to interact in immersive virtual environments.

Interaction in Space

Humans will return to the Moon in the upcoming years with the ambition to keep a sustained presence at the end of the decade [237]. The objective is to prepare for manned missions to Mars. This ambition requires developing space activities to a scale never seen before. It will involve more humans and robots in space and mission control on Earth. Space challenges many aspects we take for granted on Earth. Entities in space live in a completely different physical environment, for instance with low gravity. And the interaction between entities on earth and in space suffers from communication constraints, like latency, limited bandwidth, and connection availability. There are therefore many open research questions related to the effects of these conditions on the sensorimotor loop. Yet, space exploration has remained a niche topic in HCI research, gaining little attention and staying the exclusive domain of spacial agencies.

My objective is therefore to measure the effects of space conditions on human activities to test the limits of the sensorimotor loop and propose mitigation strategies. For example, I coordinate a consortium of experts in HCI, communications, robotics, and dependability. Together, we aim at improving the teleoperation of lunar rovers. To achieve this, we will first work on the effects of space communications conditions on the operator's ability to perform actions, and compare different communications architectures and protocols. Then we will study the human control/automation trade-off with new rover controls and environment sensing. Finally, we will integrate our solutions and ensure they could operate in a fault-tolerant way. Later I will extend this work to robot fleets, controlled either from Earth or locally by astronauts. I am also interested in the communications between astronauts and Earth with seconds- or minutes-long latencies. It will be critical during operations, but it will also affect personal communications between astronauts and their relatives. The question of how we could interact between Earth and space is an important challenge for space exploration.

Index

Bibliography

[1] Gregory D. Abowd and Russell Beale. Users, systems and interfaces: A unifying framework for interaction. In: *Proceedings of HCI '91 – People and Computers VI*. Vol. 91. Edinburgh, UK: Cambridge University Press, 1991, pp. 73–87.

[2] Johnny Accot and Shumin Zhai. Beyond Fitts' Law: Models for Trajectory-based HCI Tasks. In: *Proceedings of CHI'97*. Atlanta, United States: ACM Press, Mar. 1997, pp. 295–302.

[3] David S. Alles. Information Transmission by Phantom Sensations. In: *IEEE Transactions on Man-Machine Systems*, 11(1), 1970, pp. 85–91.

[4] Michel Amberg, Frédéric Giraud, Betty Lemaire-Semail, Paolo Olivo, Géry Casiez, and Nicolas Roussel. STIMTAC: A Tactile Input Device with Programmable Friction. In: *Extended Abstracts of UIST'11*. Santa Barbara, USA: ACM Press, Oct. 2011, pp. 7–8.

[5] Caroline Appert and Michel Beaudouin-Lafon. SwingStates: Adding State Machines to the Swing Toolkit. In: *Proceedings of UIST'06*. Montreux, Switzerland: ACM Press, Oct. 2006, pp. 319–322.

[6] Toshifumi Arai, Kimiyoshi Machii, Soshiro Kuzunuki, and Hiroshi Shojima. InteractiveDESK: A Computer-Augmented Desk Which Responds to Operations on Real Objects. In: *Extended Abstracts of CHI'95*. Denver, United States: ACM Press, May 1995, pp. 141–142.

[7] Ferran Argelaguet and Carlos Andujar. A survey of 3D object selection techniques for virtual environments. In: *Computers and Graphics*, 37(3), 2013, pp. 121–136.

[8] Ferran Argelaguet, Ludovic Hoyet, Michaël Trico, and Anatole Lecuyer. The role of interaction in virtual embodiment: Effects of the virtual hand representation. In: *Proceedings of VR 2016*. Greenville, USA: IEEE Computer Society Press, 2016, pp. 3–10.

[9] Paul Bach-y-Rita. Brain Mechanisms in Sensory Substitution. Academic Press, 1972.

[10] Gilles Bailly, Alexandre Demeure, Eric Lecolinet, and Laurence Nigay. MultiTouch menu (MTM). In: *Proceedings of IHM 2008*. Metz, France: ACM Press, Sept. 2008, pp. 165–168.

[11] Gilles Bailly, Eric Lecolinet, and Yves Guiard. Finger-Count & Radial-Stroke Shortcuts: 2 Techniques for Augmenting Linear Menus on Multi-Touch Surfaces. In: *Proceedings of CHI'10*. Atlanta, GA, USA: ACM Press, Apr. 2010, pp. 591–594.

[12] Gilles Bailly, **Thomas Pietrzak**, Jonathan Deber, and Daniel Wigdor. Métamorphe: Augmenting Hotkey Usage with Actuated Keys. In: *Proceedings of CHI'13*. Paris, France: ACM Press, May 2013, pp. 563–572.

[13] Gilles Bailly, Sidarth Sahdev, Sylvain Malacria, and **Thomas Pietrzak**. LivingDesktop: Augmenting Desktop Workstation with Actuated Devices. In: *Proceedings of CHI'16*. San Jose, USA: ACM Press, May 2016, pp. 5298–5310.

[14] Marc Baloup, Veïs Oudjail, **Thomas Pietrzak**, and Géry Casiez. Techniques de Pointage pour Cibles Distantes en Réalité Virtuelle. In: *Proceedings of IHM 2018*. Brest, France: ACM Press, Oct. 2018, pp. 100–107.

[15] Marc Baloup, **Thomas Pietrzak**, and Géry Casiez. Amélioration du Raycasting par utilisation de la sélection par proximité et du filtrage. In: *Journal d'Interaction Personne-Système*, 8(1), 2019, pp. 61–83.

[16] Marc Baloup, **Thomas Pietrzak**, and Géry Casiez. RayCursor: a 3D Pointing Facilitation Technique based on Raycasting. In: *Proceedings of CHI'19*. Glasgow, UK: ACM Press, May 2019, 101:1–101:12.

[17] Marc Baloup, **Thomas Pietrzak**, Martin Hachet, and Géry Casiez. Non-isomorphic Interaction Techniques for Controlling Avatar Facial Expressions in VR. In: *Proceedings of VRST 2021*. Osaka, Japan, 2021, pp. 1–10.

[18] Gareth Barnaby and Anne Roudaut. Mantis: A Scalable, Lightweight and Accessible Architecture to Build Multiform Force Feedback Systems. In: *Proceedings of UIST'19*. New Orleans, USA: ACM Press, Oct. 2019, pp. 937–948.

[19] Len Bass, Ross Faneuf, Reed Little, Niels Mayer, Bob Pellegrino, Scott Reed, Robert Seacord, Sylvia Sheppard, and Martha R. Szczur. A metamodel for the runtime architecture of an interactive system: the UIMS tool developers workshop. In: *SIGCHI Bulletin*, 24(1), 1992, pp. 32–37.

[20] Olivier Bau, Ivan Poupyrev, Ali Israr, and Chris Harrison. TeslaTouch: Electrovibration for Touch Surfaces. In: *Proceedings of UIST'10*. New York, USA: ACM Press, Oct. 2010, pp. 283–292.

[21] Michel Beaudouin-Lafon. Designing Interaction, Not Interfaces. In: *Proc. AVI '04*. Gallipoli, Italy: ACM Press, 2004, pp. 15–22.

[22] Michel Beaudouin-Lafon. Instrumental Interaction: An Interaction Model for Designing post-WIMP User Interfaces. In: *Proceedings of CHI'00*. The Hague, The Netherlands: ACM Press, Apr. 2000, pp. 446–453.

[23] Michel Beaudouin-Lafon. Interaction is the Future of Computing. In: *HCI Remixed, Reflections on Works That Have Influenced the HCI Community*. MIT Press, 2008, pp. 263–266.

[24] Georg von Békésy. Funneling in the nervous system and its role in loudness and sensation intensity on the skin. In: *The Journal of the Acoustical Society of America*, 30, 1958, pp. 399–412.

[25] Wael Ben Messaoud, Betty Lemaire-Semail, Marie-Ange Bueno, Michel Amberg, and Frédéric Giraud. Closed-Loop Control for Squeeze Film Effect in Tactile Stimulator. In: *Proc. International Conference and exhibition on new actuators and drives (Actuator 2014)*. Bremen, Germany, June 2014, pp. 1–4.

[26] Xiaojun Bi, Tovi Grossman, Justin Matejka, and George Fitzmaurice. Magic desk: bringing multi-touch surfaces into desktop work. In: *Proceedings of CHI'11*. Vancouver, Canada: ACM Press, May 2011, pp. 2511–2520.

[27] Xiaojun Bi, Tomer Moscovich, Gonzalo Ramos, Ravin Balakrishnan, and Ken Hinckley. An Exploration of Pen Rolling for Pen-Based Interaction. In: *Proceedings of UIST'08*. Monterey, USA: ACM Press, Oct. 2008, pp. 191–200.

[28] Melissande Biet, Géry Casiez, Frédéric Giraud, and Betty Lemaire-Semail. Discrimination of Virtual Square Gratings by Dynamic Touch on Friction Based Tactile Displays. In: *Proceedings of HAPTICS 2008*. Reno, NV, USA: IEEE Computer Society Press, Mar. 2008, pp. 41–48.

[29] Melissande Biet, Frédéric Giraud, and Betty Lemaire-Semail. Squeeze film effect for the design of an ultrasonic tactile plate. In: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 54(12), 2007, pp. 2678–2688.

[30] James C. Bliss, Michael H. Katcher, Charles H. Rodgers, and Raymond P. Shepard. Optical-to-Tactile Image Conversion for the Blind. In: *IEEE Transactions on Man-Machine Systems*, 11(1), 1970, pp. 58–65.

[31] Florian Block, Hans W. Gellersen, and Nicolas Villar. Touch-Display Keyboards: Transforming Keyboards into Interactive Surfaces. In: *Proceedings of CHI'10*. Atlanta, GA, USA: ACM Press, Apr. 2010, pp. 1145–1154.

[32] Stanley J. Bolanowski, George A. Gescheider, Ronald T. Verillo, and Christine M. Checkosky. Four channels mediate the mechanical aspect of touch. In: *Journal of the Acoustical Society of America*, 84(5), 1988, pp. 1680–1694.

[33] Richard Bolt. Put-That-There: Voice and Gesture at the Graphics Interface. In: *Proceedings of ACM SIGGRAPH 1980*. Seattle, USA: ACM Press, 1980, pp. 262–270.

[34] Matthew Botvinick and Jonathan Cohen. Rubber hands 'feel' touch that eyes see. In: *Nature*, 391(6669), 1998, p. 756.

[35] Elodie Bouzbib, Gilles Bailly, Sinan Haliyo, and Pascal Frey. "Can I Touch This?": Survey of Virtual Reality Interactions via Haptic Solutions. In: *Proceedings of IHM 2020-21*. Metz, France: ACM Press, Apr. 2021.

[36] Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. 3D User Interfaces: Theory and Practice. Redwood City, CA, USA: Addison Wesley, 2004.

[37] Stephen A. Brewster and Lorna M. Brown. Tactons: structured tactile messages for non-visual information display. In: *AUIC '04: Proceedings of the fifth conference on Australasian user interface*. Dunedin, New Zealand: Australian Computer Society, Inc., 2004, pp. 15–23.

[38] Stephen A. Brewster, Faraz Chohan, and Lorna M. Brown. Tactile feedback for mobile interactions. In: *Proceedings of CHI'07*. San Jose, California, USA: ACM Press, Apr. 2007, pp. 159–162.

[39] Stephen A. Brewster, Peter C. Wright, and Alistair Edwards. An evaluation of earcons for use in auditory human-computer interfaces. In: *Proceedings of CHI'93*. Amsterdam, The Netherlands: ACM Press, Apr. 1993, pp. 222–227.

[40] Lorna M. Brown. Tactons: Structured Vibrotactile Messages for Non-Visual Information Display. PhD thesis. Glasgow, Scotland: University of Glasgow, 2007.

[41] Lorna M. Brown, Stephen A. Brewster, and Helen C. Purchaise. Multidimensional tactons for non-visual information presentation in mobile devices. In: *Proceedings of MobileHCI '06*. Helsinki, Finland: ACM Press, 2006, pp. 231–238.

[42] Leah Buechley and Michael Eisenberg. The LilyPad Arduino: Toward Wearable Engineering for Everyone. In: *IEEE Pervasive Computing*, 7(2), 2008, pp. 12–15.

[43] Ivica Ico Bukvic, Jonathan Wilkes, and Albert Gräf. Latest developments with Pd-L2Ork and its Development Branch Purr-Data. In: *Proc. 5th International Pure Data Convention*. 2016.

[44] Vannevar Bush. As We May Think. In: *Atlantic Monthly*, 176, July 1945, pp. 101–108.

[45] William A. S. Buxton. A three state model of graphical input. In: *Proceedings of INTERACT 1990*. Cambridge, UK, 1990, pp. 449–456.

[46] Stuart K. Card, Thomas P. Moran, and Allen Newell. The Keystroke-level Model for User Performance Time with Interactive Systems. In: *Commun. ACM*, 23(7), July 1980, pp. 396–410.

[47] Stuart K. Card, Thomas P. Moran, and Allen Newell. The Psychology of Human-Computer Interaction. CRC Press, 1983.

[48] Tom Carter, Sue Ann Seah, Benjamin Long, Bruce Drinkwater, and Sriram Subramanian. UltraHaptics: Multi-Point Mid-Air Haptic Feedback for Touch Surfaces. In: *Proceedings of UIST'13*. St. Andrews, UK: ACM Press, Oct. 2013, pp. 505–514.

[49] Géry Casiez, Stéphane Conversy, Matthieu Falce, Stéphane Huot, and Nicolas Roussel. Looking through the Eye of the Mouse: A Simple Method for Measuring End-to-End Latency Using an Optical Mouse. In: *Proceedings of UIST'15*. Charlotte, USA: ACM Press, Oct. 2015, pp. 629–636.

[50] Géry Casiez, **Thomas Pietrzak**, Damien Marchal, Sébastien Poulmane, Matthieu Falce, and Nicolas Roussel. Characterizing Latency in Touch and Button-Equipped Interactive Systems. In: *Proceedings of UIST'17*. Berlin, Germany: ACM Press, Oct. 2017, pp. 29–39.

[51] Géry Casiez and Nicolas Roussel. No More Bricolage!: Methods and Tools to Characterize, Replicate and Compare Pointing Transfer Functions. In: *Proceedings of UIST'11*. Santa Barbara, USA: ACM Press, Oct. 2011, pp. 603–614.

[52] Géry Casiez, Nicolas Roussel, Romuald Vanbelleghem, and Frédéric Giraud. Surfpad: Riding Towards Targets on a Squeeze Film Effect. In: *Proceedings of CHI'11*. Vancouver, Canada: ACM Press, May 2011, pp. 2491–2500.

[53] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 1€Filter: A Simple Speed-based Low-pass Filter for Noisy Input in Interactive Systems. In: *Proceedings of CHI'12*. Austin, TX, USA: ACM Press, May 2012, pp. 2527–2530.

[54] Elie Cattan, Amélie Rochet-Capellan, Pascal Perrier, and François Bérard. Reducing Latency with a Continuous Prediction: Effects on Users' Performance in Direct-Touch Target Acquisitions. In: *Proceedings of ITS 2015*. Madeira, Portugal, 2015, pp. 205–214.

[55] Jessica R. Cauchard, Janette L. Cheng, **Thomas Pietrzak**, and James A. Landay. ActiVibe: Design and Evaluation of Vibrations for Progress Monitoring. In: *Proceedings of CHI'16*. San Jose, USA: ACM Press, May 2016, pp. 3261–3271.

[56] Alonzo Church. A Set of Postulates for the Foundation of Logic. In: *Annals of Mathematics*, 33(2), 1932, pp. 346–366.

[57] Ernest Cline. Ready Player One. Ballantine Books, 2011.

[58] Andy Cockburn, Carl Gutwin, Joey Scar, and Sylvain Malacria. Supporting Novice to Expert Transitions in User Interfaces. In: *ACM Comput. Surv.*, 47(2), Nov. 2014, 31:1–31:36.

[59] Carter Compton Collins and Paul Bach-y-Rita. Transmission of pictorial information through the skin. In: *Advances in Biological Medecine and physiology*, 14, 1973, pp. 285–315.

[60] Sunny Consolvo, David W. McDonald, Tammy Toscos, Mike Y. Chen, Jon Froehlich, Beverly Harrison, Predrag Klasnja, Anthony LaMarca, Louis LeGrand, Ryan Libby, Ian Smith, and James A. Landay. Activity Sensing in the Wild: A Field Trial of Ubifit Garden. In: *Proceedings of CHI'08*. Florence, Italy: ACM Press, Apr. 2008, pp. 1797–1806.

[61] Melvin Edward Conway. How do Committees invent? In: *Datamation*, 14(4), Apr. 1968, pp. 28–31.

[62] Joëlle Coutaz. PAC, an Oriented Object Model for Dialog Design. In: *Proceedings of INTERACT 1987*. Stuttgart, Germany, 1987, pp. 431–436.

[63] Andrew Crossan. The Design and Evaluation of a Haptic Veterinary Training Simulator. PhD thesis. Glasgow, Scotland: University of Glasgow, 2003.

[64] Jonathan Deber, Bruno Araújo, Ricardo Jota, Clifton Forlines, Darren Leigh, Steven Sanders, and Daniel Wigdor. Hammer Time! A Low-Cost, High Precision, High Accuracy Tool to Measure the Latency of Touchscreen Devices. In: *Proceedings of CHI'16*. San Jose, USA: ACM Press, May 2016, pp. 2857–2868.

[65] Jonathan Deber, Ricardo Jota, Clifton Forlines, and Daniel Wigdor. How Much Faster is Fast Enough? User Perception of Latency & Latency Improvements in Direct and Indirect Touch. In: *Proceedings of CHI'15*. Seoul, Korea: ACM Press, May 2015, pp. 1827–1836.

[66] William Delamare, Céline Coutrix, and Laurence Nigay. Mobile Pointing Task in the Physical World: Balancing Focus and Performance While Disambiguating. In: *Proceedings of MobileHCI '13*. Munich, Germany: ACM Press, 2013, pp. 89–98.

[67] Marc Demers, Pascal E. Fortin, Antoine Weill, Yongjae Yoo, and Jeremy R. Cooperstock. Active Sampling for Efficient Subjective Evaluation of Tactons at Scale. In: *Proceedings of WorldHaptics 2021*. Montreal, Canada: IEEE Computer Society Press, Mar. 2021, pp. 1–6.

[68] Diane Dewez, Rebecca Fribourg, Ferran Argelaguet, Ludovic Hoyet, Daniel Mestre, Mel Slater, and Anatole Lecuyer. Influence of Personality Traits and Body Awareness on the Sense of Embodiment in Virtual Reality. In: *Proceedings of ISMAR 2019*. IEEE Computer Society Press, 2019, pp. 123–134.

[69] Paul H. Dietz, Benjamin Eidelson, Jonathan Westhues, and Steven Bathiche. A Practical Pressure Sensitive Computer Keyboard. In: *Proceedings of UIST'09*. Victoria, Canada: ACM Press, Oct. 2009, pp. 55–58.

[70] Paul H. Dietz and Darren Leigh. DiamondTouch: A Multi-User Touch Technology. In: pp. 219–226.

[71] Paul Ekman and Wallace V. Friesen. Facial action coding systems. Consulting Psychologists Press, 1978.

[72] George ElKoura and Karan Singh. Handrix: Animating the Human Hand. In: *Proc. of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '03)*. Goslar, Germany, 2003, pp. 110–119.

[73] Douglas C. Engelbart and William K. English. A Research Center for Augmenting Human Intellect. In: *Proc. AFIPS '68 (Fall, part I)*. San Francisco, California: ACM Press, 1968, pp. 395–410.

[74] Mario Enriquez, Karon E. MacLean, and Christian Chita. Haptic Phonemes: Basic Building Blocks of Haptic Communication. In: *Proceedings of ICMI '06*. Banff, Canada: ACM Press, Nov. 2006, pp. 302–309.

[75] Philipp Ewerling, Alexander Kulik, and Bernd Froehlich. Finger and Hand Detection for Multi-Touch Interfaces Based on Maximally Stable Extremal Regions. In: *Proceedings of ITS 2012*. Cambridge, USA: ACM Press, 2012, pp. 173–182.

[76] Gustav T. Fechner. Elemente der psychophysik. Leipzig : Breitkopf und Härtel, 1860.

[77] Nicholas Fellion, Alexander Keith Eady, and Audrey Girouard. FlexStylus: A Deformable Stylus for Digital Art. In: *Extended Abstracts of CHI'16*. San Jose, USA: ACM Press, May 2016, pp. 2482–2489.

[78] Nicholas Fellion, **Thomas Pietrzak**, and Audrey Girouard. FlexStylus: Leveraging Bend Input for Pen Interaction. In: *Proceedings of UIST'17*. Berlin, Germany: ACM Press, Oct. 2017, pp. 2482–2489.

[79] Paul M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. In: *Journal of Experimental Psychology*, 47(6), June 1954, pp. 381–391.

[80] Sean Follmer, Daniel Leithinger, Alex Olwal, Akimitsu Hogge, and Hiroshi Ishii. InFORM: Dynamic Physical Affordances and Constraints through Shape and Object Actuation. In: *Proceedings of UIST'13*. St. Andrews, UK: ACM Press, Oct. 2013, pp. 417–426.

[81] Christian Frisson, Julien Decaudin, **Thomas Pietrzak**, Alexander Ng, Pauline Poncet, Fabrice Casset, Antoine Latour, and Stephen A. Brewster. Designing Vibrotactile Widgets with Printed Actuators and Sensors. In: *Extended Abstracts of UIST'17*. Québec, Canada: ACM Press, Oct. 2017, pp. 11–13.

[82] Christian Frisson, Julien Decaudin, Mario Sanz-Lopez, and **Thomas Pietrzak**. Printgets: an Open-Source Toolbox for Designing Vibrotactile Widgets with Industrial-Grade Printed Actuators and Sensors. In: *Proc. HAID 2020*. Montreal, Canada, May 2020.

[83] Christian Frisson, **Thomas Pietrzak**, Siyan Zhao, Zachary Schwemier, and Ali Israr. WebAudioHaptics: Tutorial on Haptics with Web Audio. In: *Web Audio Haptics WAC Tutorial 2016*. Atlanta, USA, Apr. 2016.

[84] Masaaki Fukumoto and Toshiaki Sugimura. Active click: tactile feedback for touch panels. In: *Extended Abstracts of CHI'01*. Seattle, Washington: ACM Press, Apr. 2001, pp. 121–122.

[85] Olivier Gapenne, Amal Ali Ammar, Charles Lenay, and Katia Rovira. Tactos: Special computer interface for the reading and writing of 2D forms in blind people. In: *Proc. HCI International 2003*. Creete, Greece, 2003, pp. 1270–1274.

[86] William W. Gaver. Synthesizing Auditory Icons. In: *Proceedings of CHI'93*. Amsterdam, The Netherlands: ACM Press, Apr. 1993, pp. 228–235.

[87] William W. Gaver. Technology Affordances. In: *Proceedings of CHI'91*. New Orleans, United States: ACM Press, Apr. 1991, pp. 79–84.

[88] Frank A. Geldard and Carl E. Jr. Sherrick. The cutaneous "rabbit": a perceptual illusion. In: *Science*, 178, 1972, pp. 178–179.

[89] George A. Gescheider, Stanley J. Bolanowski, Tyler C. Greenfield, and Katyarina E. Brunette. Perception of the tactile texture of raised-dot patterns: A multidimensional analysis. In: *Somatosensory & Motor Research*, 22(3), 2005, pp. 127–140.

[90] Emilien Ghomi, Stéphane Huot, Olivier Bau, Michel Beaudouin-Lafon, and Wendy. E. Mackay. ArpèGe: Learning Multitouch Chord Gestures Vocabularies. In: *Proceedings of ITS 2013*. St. Andrews, United Kingdom: ACM Press, 2013, pp. 209–218.

[91] James J. Gibson. Observations on Active Touch. In: *Psychological Review*, 69(6), 1962, pp. 4771–491.

[92] James J. Gibson. The Ecological Approach to Visual Perception. Boston: Houghton Mifflin, 1979.

[93] James J. Gibson. The theory of affordances. In: *Perceiving, Acting, and Knowing*. Lawrence Erlbaum Associates, 1977, pp. 67–82.

[94] William Gibson. Neuromancer. Ace, 1984.

[95] Kurt Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. In: *Monatshefte für Mathematik und Physik*, 38(1), Dec. 1931, pp. 173–198.

[96] Genevieve D. Goff. Differential Discrimination of Frequency of Cutaneous Mechanical Vibration. In: *Journal of Experimental Psychology*, 74(2), 1967, pp. 294–299.

[97] Alix Goguey, Géry Casiez, **Thomas Pietrzak**, Daniel Vogel, and Nicolas Roussel. Adoiraccourcix: Sélection de Commandes sur Écrans Tactiles Multi-Points par Identification des Doigts. In: *Proceedings of IHM 2014*. Lille, France: ACM Press, Oct. 2014, pp. 28–37.

[98] Alix Goguey, Géry Casiez, Daniel Vogel, Fanny Chevalier, **Thomas Pietrzak**, and Nicolas Roussel. A three-step interaction pattern for improving discoverability in finger identification techniques. In: *Extended Abstracts of UIST'14*. Honolulu, USA: ACM Press, Oct. 2014.

[99] Alix Goguey, Daniel Vogel, Fanny Chevalier, **Thomas Pietrzak**, Nicolas Roussel, and Géry Casiez. Leveraging finger identification to integrate multi-touch command selection and parameter manipulation. In: *International Journal of Human-Computer Studies*, 99(2), 2017, pp. 21–36.

[100] Dina Goldin and Peter Wegner. The Interactive Nature of Computing: Refuting the Strong Church–Turing Thesis. In: *Minds and Machines*, 18(2), Mar. 2008, pp. 17–38.

[101] Mar Gonzalez-Franco and Tabitha C. Peck. Avatar embodiment. towards a standardized questionnaire. In: *Frontiers in Robotics and AI*, 5, June 2018, pp. 1–9.

[102] Louis D. Goodfellow. An Empirical Comparison of Audition, Vision, and touch in the Discrimination of Short Intervals of Time. In: *The American Journal of Psychology*, 46(2), Apr. 1934, pp. 243–258.

[103] Simon Grondin and Devin McAuley. Duration discrimination in crossmodal sequences. In: *Perception*, 38(10), 2009, pp. 1542–1559.

[104] Tovi Grossman and Ravin Balakrishnan. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In: *Proceedings of CHI'05*. Portland, USA: ACM Press, 2005, pp. 281–290.

[105] Tovi Grossman and Ravin Balakrishnan. The Design and Evaluation of Selection Techniques for 3D Volumetric Displays. In: *Proceedings of UIST'06*. Montreux, Switzerland: ACM Press, Oct. 2006, pp. 3–12.

[106] Tovi Grossman, George Fitzmaurice, and Ramtin Attar. A Survey of Software Learnability: Metrics, Methodologies and Guidelines. In: *Proceedings of CHI'09*. Boston, MA, USA: ACM Press, Apr. 200, pp. 649–658.

[107] Alfrancis Guerrero, **Thomas Pietrzak**, and Audrey Girouard. HyperBrush: Exploring the Influence of Flexural Stiffness on the Performance and Preference for Bendable Stylus Interfaces. In: *Proceedings of INTERACT 2021*. Bari, Italy: Springer-Verlag, 2021.

[108] Yves Guiard, Halla B. Olafsdottir, and Simon Perrault. Fitt's Law as an Explicit Time/Error Trade-Off. In: *Proceedings of CHI'11*. Vancouver, Canada: ACM Press, May 2011, pp. 1619–1628.

[109] Maxime Guillon, François Leitner, and Laurence Nigay. Investigating Visual Feedforward for Target Expansion Techniques. In: *Proceedings of CHI'15*. Seoul, Korea: ACM Press, May 2015, pp. 2777–2786.

[110] Pascal Guitton and Nicolas Roussel. Le métavers, quels métavers ? In: *Binaire*, Feb. 2022.

[111] Aakar Gupta and **Thomas Pietrzak**. A New Haptic Interaction Paradigm. In: *CHI 2016 Workshop on Mid-Air Haptics and Displays: Systems for Un-instrumented Mid-Air Interactions*. San Jose, USA, May 2016, pp. 193–200.

[112] Aakar Gupta, **Thomas Pietrzak**, Nicolas Roussel, and Ravin Balakrishnan. Direct Manipulation in Tactile Displays. In: *Proceedings of CHI'16*. San Jose, USA: ACM Press, May 2016, pp. 3683–3693.

[113] Aakar Gupta, **Thomas Pietrzak**, Cleon Yau, Nicolas Roussel, and Ravin Balakrishnan. Summon and Select: Rapid Interaction with Interface Controls in Mid-air. In: *Proc. ISS'17*. Brighton, UK, Oct. 2017, pp. 52–61.

[114] Sidhant Gupta, Dan Morris, Shwetak N. Patel, and Desney Tan. AirWave: Non-Contact Haptic Feedback Using Air Vortex Rings. In: *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. Zurich, Switzerland: ACM Press, 2013, pp. 419–428.

[115] Chris Harrison, Julia Schwarz, and Scott E. Hudson. TapSense: Enhancing Finger Interaction on Touch Surfaces. In: *Proceedings of UIST'11*. Santa Barbara, USA: ACM Press, Oct. 2011, pp. 627–636.

[116] Ken Hinckley, Xiang 'Anthony' Chen, and Hrvoje Benko. Motion and Context Sensing Techniques for Pen Computing. In: *Proceedings of Graphics Interface 2013*. Regina, Canada: Canadian Information Processing Society, 2013, pp. 71–78.

[117] Ken Hinckley, Koji Yatani, Michel Pahud, Nicole Coddington, Jenny Rodenhouse, Andrew D. Wilson, Hrvoje Benko, and William A. S. Buxton. Pen + Touch = New Tools. In: *Proceedings of UIST'10*. New York, USA: ACM Press, Oct. 2010, pp. 27–36.

[118] Alexander Hoffmann, Daniel Spelmezan, and Jan Borchers. TypeRight: A Keyboard with Tactile Error Prevention. In: *Proceedings of CHI'09*. Boston, MA, USA: ACM Press, Apr. 200, pp. 2265–2268.

[119] Eve Hoggan. Crossmodal Audio and Tactile Interaction with Mobile Touchscreens. PhD thesis. University of Glasgow, UK, 2010.

[120] Eve Hoggan and Stephen A. Brewster. New parameters for tacton design. In: *Extended Abstracts of CHI'07*. San Jose, California, USA: ACM Press, Apr. 2007, pp. 2417–2422.

[121] Eve Hoggan, Stephen A. Brewster, and Jody Johnston. Investigating the effectiveness of tactile feedback for mobile touchscreens. In: *Proceedings of CHI'08*. Florence, Italy: ACM Press, Apr. 2008, pp. 1573–1582.

[122] Mark Hollins and Ryan Risner. Evidence for the duplex theory of tactile texture perception. In: *Perception & Psychophysics*, 62, 2000, pp. 695–705.

[123] Christian Holz and Patrick Baudisch. Fiberio: A Touchscreen That Senses Fingerprints. In: *Proceedings of UIST'13*. St. Andrews, UK: ACM Press, Oct. 2013, pp. 41–50.

[124] Takayuki Hoshi, Masafumi Takahashi, Takayuki Iwamoto, and Hiroyuki Shinoda. Noncontact Tactile Display Based on Radiation Pressure of Airborne Ultrasound. In: *IEEE Transactions on Haptics*, 3(3), 2010, pp. 155–165.

[125] Ludovic Hoyet, Ferran Argelaguet, Corentin Nicole, and Anatole Lecuyer. "Wow! I Have Six Fingers!": Would You Accept Structural Changes of Your Hand in VR? In: *Frontiers in Robotics and AI*, 3, 2016.

[126] Stéphane Huot. 'Designeering Interaction': A Missing Link in the Evolution of Human-Computer Interaction. Habilitation à Diriger des Recherches (HDR). Université Paris-Sud, 2013.

[127] Yasushi Ikei, Kazufumi Wakamatsu, and Shuichi Fukuda. Texture presentation by vibratory tactile display-image based presentation of a tactile texture. In: *Proceedings of VR 1997*. Albuquerque, USA: IEEE Computer Society Press, 1997, pp. 199–205.

[128] Hiroshi Ishii and Brygg Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In: *Proceedings of CHI'97*. Atlanta, United States: ACM Press, Mar. 1997, pp. 234–241.

[129] Ali Israr and Ivan Poupyrev. Tactile brush: drawing on skin with a tactile grid display. In: *Proceedings of CHI'11*. Vancouver, Canada: ACM Press, May 2011, pp. 2019–2028.

[130] Hiroo Iwata, Hiroaki Yano, Fumitaka Nakaizumi, and Ryo Kawamura. Project FEELEX: Adding Haptic Surface to Graphics. In: *Proceedings of SIGGRAPH '01*. ACM Press, 2001, pp. 469–476.

[131] Rob Jacob, Audrey Girouard, Leanne M. Hirshfield, Michael S. Horn, Orit Shaer, Erin T. Solovey, and Jamie Zigelbaum. Reality-Based Interaction: A Framework for Post-WIMP Interfaces. In: *Proceedings of CHI'08*. CHI '08. Florence, Italy: ACM Press, Apr. 2008, pp. 201–210.

[132] Brett R. Jones, Hrvoje Benko, Eyal Ofek, and Andrew D. Wilson. IllumiRoom: Peripheral Projected Illusions for Interactive Experiences. In: *Proceedings of CHI'13*. Paris, France: ACM Press, May 2013, pp. 869–878.

[133] Ricardo Jota, Albert Ng, Paul H. Dietz, and Daniel Wigdor. How Fast is Fast Enough? A Study of the Effects of Latency in Direct-Touch Pointing Tasks. In: *Proceedings of CHI'13*. Paris, France: ACM Press, May 2013, pp. 2291–2300.

[134] Jun Kato, Daisuke Sakamoto, and Takeo Igarashi. Surfboard: Keyboard with Microphone as a Low-Cost Interactive Surface. In: *Proceedings of UIST'10*. New York, USA: ACM Press, Oct. 2010, pp. 387–388.

[135] Konstantina Kilteni, Raphaela Groten, and Mel Slater. The sense of embodiment in virtual reality. In: *Presence: Teleoperators and Virtual Environments*, 21(4), 2012, pp. 373–387.

[136] Hyunyoung Kim, Céline Coutrix, and Anne Roudaut. KnobSlider: design of a shape-changing device grounded in users' needs. In: *Proceedings of IHM 2016*. Fribourg, Suisse: ACM Press, Oct. 2016, pp. 91–102.

[137] Seoktae Kim, Hyunjung Kim, Boram Lee, Tek-Jin Nam, and Woohun Lee. Inflatable mouse: volume-adjustable mouse with air-pressure-sensitive input and haptic feedback. In: *Proceedings of CHI'08*. Florence, Italy: ACM Press, Apr. 2008, pp. 211–224.

[138] Sunjun Kim and Geehyuk Lee. Haptic Feedback Design for a Virtual Button along Force-Displacement Curves. In: *Proceedings of UIST'13*. St. Andrews, UK: ACM Press, Oct. 2013, pp. 91–96.

[139] Donald Knuth. The Art of Computer Programming. Vol. 1. Addison-Wesley, 1968.

[140] Elena Kokkinara and Mel Slater. Measuring the Effects through Time of the Influence of Visuomotor and Visuotactile Synchronous Stimulation on a Virtual Body Ownership Illusion. In: *Perception*, 43(1), 2014, pp. 43–58.

[141] Regis Kopper, Felipe Bacim, and Doug A. Bowman. Rapid and accurate 3D selection by progressive refinement. In: *Proceedings of 3DUI 2011*. IEEE Computer Society Press, Mar. 2011, pp. 67–74.

[142] Adrien Laversanne-Finot, Alexandre Péré, and Pierre-Yves Oudeyer. Curiosity Driven Exploration of Learned Disentangled Goal Spaces. In: *Proc. CoRL 2018*. Zürich, Switzerland, Oct. 2018.

[143] Mathieu Le Goc, Lawrence Hyunjin Kim, Ali Parsaei, Jean-Daniel Fekete, Pierre Dragicevic, and Sean Follmer. Zooids: Building Blocks for Swarm User Interfaces. In: *Proceedings of UIST'16*. Tokyo, Japan: ACM Press, Oct. 2016, pp. 97–109.

[144] Susan J. Lederman and Roberta L. Klatzky. Action for perception: Manual exploratory movements for haptically processing objects and their features. In: *Hand and Brain: Neurophysiology and Psychology of Hand*. San Diego: Academic, 1996, pp. 431–446.

[145] Susan J. Lederman and Roberta L. Klatzky. Hand movements: A window into haptic object recognition. In: *Cognitive Psychology*, 19(3), 1987, pp. 342–368.

[146] Susan J. Lederman, Jack M. Loomis, and Deborah A Williams. The role of vibration in the tactual perception of roughness. In: *Perception & Psychophysics*, 32(2), 1982, pp. 109–116.

[147] Susan J. Lederman and Martin M Taylor. Fingertip force, surface geometry, and the perception of roughness by active touch. In: *Perception & Psychophysics*, 12(5), 1972, pp. 401–408.

[148] Marjorie R. Leek. Adaptive procedures in psychophysical research. In: *Perception & Psychophysics*, 63(8), 2001, pp. 1279–1292.

[149] Daniel Leithinger and Hiroshi Ishii. Relief: a scalable actuated shape display. In: *Proceedings of TEI'10*. Cambridge, MA, USA: ACM Press, 2010, pp. 221–222.

[150] James Leonard and Claude Cadoz. Physical Modelling Concepts for a Collection of Multisensory Virtual Musical Instruments. In: *Proceedings of New Interfaces for Musical Expression 2015*. May 2015, pp. 150–155.

[151] G. Julian Lepinski, Tovi Grossman, and George Fitzmaurice. The Design and Evaluation of Multitouch Marking Menus. In: *Proceedings of CHI'10*. Atlanta, GA, USA: ACM Press, Apr. 2010, pp. 2233–2242.

[152] Vincent Lévesque, Louise Oram, Karon E. MacLean, Andy Cockburn, Nicholas D. Marchuk, Dan Johnson, J. Edward Colgate, and Michael A. Peshkin. Enhancing physicality in touch interaction with programmable friction. In: *Proceedings of CHI'11*. Vancouver, Canada: ACM Press, May 2011, pp. 2481–2490.

[153] James R. Lewis, Kathleen M. Potiosnak, and Regis L. Magyar. Keys and Keyboards. In: *Handbook of Human-Computer Interaction*. Ed. by Martin G. Helander, Thomas K. Landauer, and Prasad V. Prabhu. Elsevier, 1997, pp. 1285–1311.

[154] Hao Li, Laura Trutoiu, Kyle Olszewski, Lingyu Wei, Tristan Trutna, Pei-Lun Hsieh, Aaron Nicholls, and Chongyang Ma. Facial Performance Sensing Head-Mounted Display. In: *ACM Transactions on Graphics*, 34(4), July 2015.

[155] John Linvill and James C. Bliss. A direct translation reading aid for the blind. In: *Proceedings of the IEEE*, 54(1), 1966, pp. 40–51.

[156] Jean-Luc Lugrin, David Zilch, Daniel Roth, Gary Bente, and Marc Erich Latoschik. FaceBo: Real-time face and body tracking for faithful avatar synthesis. In: *Proceedings of VR 2016*. Greenville, USA: IEEE Computer Society Press, 2016, pp. 225–226.

[157] Jani Lylykangas, Veikko Surakka, Katri Salminen, Jukka Raisamo, Pauli Laitinen, Kasper Rönning, and Roope Raisamo. Designing Tactile Feedback for Piezo Buttons. In: *Proceedings of CHI'11*. Vancouver, Canada: ACM Press, May 2011, pp. 3281–3284.

[158] I. Scott MacKenzie. Fitts' Law As a Research and Design Tool in Human-computer Interaction. In: *Hum.-Comput. Interact.*, 7(1), Mar. 1992, pp. 91–139.

[159] I. Scott MacKenzie and William A. S. Buxton. Extending Fitts' law to two-dimensional tasks. In: *Proceedings of CHI'92*. Monterey, United States: ACM Press, May 1992, pp. 219–226.

[160] Jock Mackinlay, Stuart K. Card, and George G. Robertson. A Semantic Analysis of the Design Space of Input Devices. In: *Human–Computer Interaction*, 5(2-3), 1990, pp. 145–190.

[161] Karon E. MacLean. Putting Haptics into the Ambience. In: *IEEE Transactions on Haptics*, 2, 2009, pp. 123–135.

[162] Karon E. MacLean and Mario Enriquez. Perceptual Design of Haptic Icons. In: *Proceedings of Eurohaptics 2003*. Dublin, UK: ACM Press, July 2003, pp. 351–363.

[163] Sylvain Malacria, Gilles Bailly, Joel Harrison, Andy Cockburn, and Carl Gutwin. Promoting Hotkey Use through Rehearsal with ExposeHK. In: *Proceedings of CHI'13*. Paris, France: ACM Press, May 2013, pp. 573–582.

[164] Nicolai Marquardt, Johannes Kiemer, David Ledo, Sebastian Boring, and Saul Greenberg. Designing User-, Hand-, and Handpart-Aware Tabletop Interactions with the TouchID Toolkit. In: *Proceedings of ITS 2011*. Kobe, Japan: ACM Press, 2011, pp. 21–30.

[165] Damien Masson, Alix Goguey, Sylvain Malacria, and Géry Casiez. WhichFingers: Identifying Fingers on Touch Surfaces and Keyboards Using Vibration Sensors. In: *Proceedings of UIST'17*. Berlin, Germany: ACM Press, Oct. 2017, pp. 41–48.

[166] Albert Mehrabian. Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in Temperament. In: *Current Psychology*, 14(4), Dec. 1996, pp. 261–292.

[167] David E. Meyer, Richard A. Abrams, Sylvan Kornblum, Charles E. Wright, and J. E. Keith Smith. Optimality in human motor performance: Ideal control of rapid aimed movements. In: *Psychological Review*, 95(3), 1988, pp. 340–370.

[168] Georg Michelitsch, Jason Williams, Martin Osen, Beatriz Jimenez, and Stefan Rapp. Haptic chameleon: a new concept of shape-changing user interface controls with force feedback. In: *Extended Abstracts of CHI'04*. Vienna, Austria: ACM Press, Apr. 2004, pp. 1305–1308.

[169] Hans Moravec. Mind Children. Harvard University Press, 1988.

[170] Masahiro Mori. The uncanny valley. In: *Energy*, 7(4), 1970, pp. 33–35.

[171] Bruce J. P. Mortimer, Gary A. Zets, and Roger W. Cholewiak. Vibrotactile transduction and transducers. In: *Journal of the Acoustical Society of America*, 121(5), 2007, pp. 2970–2977.

[172] Atsuo Murata and Hirokazu Iwase. Extending Fitts' law to a three-dimensional pointing task. In: *Human Movement Science*, 20(6), Dec. 2001, pp. 791–805.

[173] Sundar Murugappan, Vinayak, Niklas Elmqvist, and Karthik Ramani. Extended Multitouch: Recovering Touch Posture and Differentiating Users Using a Depth Camera. In: *Proceedings of UIST'12*. Boston, USA: ACM Press, Oct. 2012, pp. 487–496.

[174] Mathieu Nancel, Stanislav Aranovskiy, Rosane Ushirobira, Denis Efimov, Sébastien Poulmane, Nicolas Roussel, and Géry Casiez. Next-Point Prediction for Direct Touch Using Finite-Time Derivative Estimation. In: *Proceedings of UIST'18*. Berlin, Germany: ACM Press, Oct. 2018, pp. 793–807.

[175] NASA. Human Integration Design Handbook. Tech. rep. NASA/SP-2010-3407/REV1. NASA, June 2014.

[176] Andrew Nashel and Sharif Razzaque. Tactile virtual buttons for mobile devices. In: *Extended Abstracts of CHI'03*. Ft. Lauderdale, Florida, USA: ACM Press, Apr. 2003, pp. 854–855.

[177] Albert Ng, G. Julian Lepinski, Daniel Wigdor, Steven Sanders, and Paul H. Dietz. Designing for Low-Latency Direct-Touch Input. In: *Proceedings of UIST'12*. Boston, USA: ACM Press, Oct. 2012, pp. 453–464.

[178] Alexander Ng, Stephen A. Brewster, Frank Beruscha, and Wolfgang Krautter. An Evaluation of Input Controls for In-Car Interactions. In: *Proceedings of CHI'17*. Denver, USA: ACM Press, May 2017, pp. 2845–2852.

[179] Jakob Nielsen. Enhancing the Explanatory Power of Usability Heuristics. In: *Proceedings of CHI'94*. Boston, United States: ACM Press, Apr. 1994, pp. 152–158.

[180] Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In: *Proceedings of CHI'90*. Seattle, United States: ACM Press, Apr. 1990, pp. 249–256.

[181] Laurence Nigay. Design space for multimodal interaction. In: *Proceedings of IFIP Congress'2004, Building the Information Society*. Toulouse, France, 2004, pp. 403–408.

[182] Laurence Nigay and Joëlle Coutaz. A Design Space For Multimodal Systems: Concurrent Processing and Data Fusion. In: *Proceedings of INTERCHI'93*. Amsterdam, Netherlands: ACM Press, 1993, pp. 172–178.

[183] Laurence Nigay and Joëlle Coutaz. A Generic Platform for Addressing the Multimodal Challenge. In: *Proceedings of CHI'95*. Denver, United States: ACM Press, May 1995, pp. 98–105.

[184] Donald A. Norman. Natural User Interfaces Are Not Natural. In: *Interactions*, 17(3), May 2010, pp. 6–10.

[185] Donald A. Norman. The Design of Everyday Things. Basic Books Inc., 2002.

[186] Donald A. Norman. The design of everyday things: Revised and expanded edition. Basic Books Inc., 2013.

[187] Donald A. Norman. The Psychology of Everyday Things. Basic Books Inc., 1988.

[188] Ian Oakley. Haptic Augmentation of the Cursor: Transforming Virtual Actions into Physical Actions. PhD thesis. University of Glasgow, UK, 2003.

[189] Ian Oakley, Marilyn Rose McGee, Stephen A. Brewster, and Philip Gray. Putting the feel in look and feel. In: *Proceedings of CHI'00*. The Hague, The Netherlands: ACM Press, Apr. 2000, pp. 415–422.

[190] Nicolas Olivier, Ludovic Hoyet, Fabien Danieau, Ferran Argelaguet, Quentin Avril, Anatole Lecuyer, Philippe Guillotel, and Franck Multon. The Impact of Stylization on Face Recognition. In: *Proceedings of SAP 2020*. Virtual Event: ACM Press, 2020.

[191] Igor S. Pandzic and Forchheimer Robert, eds. MPEG-4 Facial Animation: The Standard, Implementation And Applications. John Wiley & Sons, Inc., 2003.

[192] Sandra Parriaud, Valentine Reynaud, Jean-Baptiste Joatton, sybille Caffiau, and Céline Coutrix. Towards More Tactile Information on Sliders. In: *Proceedings of IHM 2019*. Grenoble, France: ACM Press, Oct. 2019.

[193] Jérôme Pasquero, Scott J. Stobbe, and Noel Stonehouse. A Haptic Wristatch for Eyes-Free Interactions. In: *Proceedings of CHI'11*. Vancouver, Canada: ACM Press, May 2011, pp. 3257–3266.

[194] Tabitha C. Peck and Mar Gonzalez-Franco. Avatar embodiment. A standardized questionnaire. In: *Frontiers in Virtual Reality*, 1, 2021, p. 44.

[195] Delphine Picard, Catherine Dacremont, Dominique Valentin, and Agnes Giboreau. Perceptual dimensions of tactile textures. In: *Acta psychologica*, 114(2), 2003, pp. 165–184.

[196] **Thomas Pietrzak**. Contributions à la dissémination d'informations haptiques dans un environnement multimodal. PhD thesis. France: Université Paul Verlaine – Metz, Dec. 2008.

[197] **Thomas Pietrzak**. PICOB : Post-It Haptiques par Codes-Barres. Master's thesis. France: Université Paul Verlaine – Metz, July 2005.

[198] **Thomas Pietrzak**, Andrew Crossan, Stephen A. Brewster, Benoît Martin, and Isabelle Pecci. Creating Usable Pin Array Tactons for Non-Visual Information. In: *IEEE Transactions on Haptics*, 2(2), 2009, pp. 61–72.

[199] **Thomas Pietrzak**, Sylvain Malacria, and Gilles Bailly. CtrlMouse et TouchCtrl: duplicating mode delimiters on the mouse. In: *Proceedings of IHM 2014*. Lille, France: ACM Press, Oct. 2014, pp. 38–47.

[200] **Thomas Pietrzak**, Benoît Martin, and Isabelle Pecci. Affichage d'informations par des impulsions haptiques. In: *Proceedings of IHM 2005*. Toulouse, France: ACM Press, Sept. 2005, pp. 223–226.

[201] **Thomas Pietrzak**, Benoît Martin, and Isabelle Pecci. Information display by dragged haptic bumps. In: *Proceedings of Enactive /05*. Genova, Italy, 2005.

[202] **Thomas Pietrzak**, Isabelle Pecci, and Benoît Martin. Static and dynamic tactile directional cues experiments with VTPlayer mouse. In: *Proceedings of EuroHaptics 2006*. Paris, France, July 2006, pp. 63–68.

[203] **Thomas Pietrzak**, Nicolas Roussel, Aakar Gupta, and Ravin Balakrishnan. Manipulation Dialogique pour Affichages Tactiles. In: *Proceedings of IHM 2015*. Toulouse, France: ACM Press, Oct. 2015, pp. 1–10.

[204] Robert Plutchik. The Nature of Emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice. In: *American Scientist*, 89(4), 2001, pp. 344–350.

[205] Pauline Poncet, Fabrice Casset, Antoine Latour, Fabrice Domingues Dos Santos, Sébastien Pawlak, Romain Gwoziecki, Arnaud Devos, Patrick Emery, and Stéphane Fanget. Static and Dynamic Studies of Electro-Active Polymer Actuators and Integration in a Demonstrator. In: *Actuators*, 6(2), 2017.

[206] Pauline Poncet, Fabrice Casset, Antoine Latour, Fabrice Domingues Dos Santos, Sébastien Pawlak, Romain Gwoziecki, and Stéphane Fanget. Design and realization of electroactive polymer actuators for transparent and flexible haptic feedback interfaces. In: *Proc. EuroSimE 2016*. Apr. 2016, pp. 1–5.

[207] Nicole Ke Chen Pong and Sylvain Malacria. Awareness, Usage and Discovery of Swipe-Revealed Hidden Widgets in IOS. In: *Proceedings of ISS 2019*. ACM Press, 2019, pp. 193–204.

[208] Ludovic Potier, **Thomas Pietrzak**, Géry Casiez, and Nicolas Roussel. Concevoir des motifs tactiles par friction programmable. In: *Proceedings of IHM 2016*. Fribourg, Suisse: ACM Press, Oct. 2016, pp. 1–7.

[209] Ludovic Potier, **Thomas Pietrzak**, Géry Casiez, and Nicolas Roussel. Méthodologie de conception de textures pour les interfaces tactiles à frottement programmable. In: *Proceedings of IHM 2012*. Biarritz, France: ACM Press, Oct. 2012, pp. 121–128.

[210] Ivan Poupyrev, Shigeaki Maruyama, and Jun Rekimoto. Ambient Touch: Designing Tactile Interfaces for Handheld Devices. In: *Proceedings of UIST'02*. Paris, France: ACM Press, Oct. 2002, pp. 51–60.

[211] Trygve M.H. Reenskaug. MODELS - VIEWS - CONTROLLERS. Tech. rep. Xerox PARC, Dec. 1979.

[212] Trygve M.H. Reenskaug. THING-MODEL-VIEW-EDITOR - an Example from a planningsystem. Tech. rep. Xerox PARC, May 1979.

[213] Jun Rekimoto, Takaaki Ishizawa, Carsten Schwesig, and Haruo Oba. PreSense: interaction techniques for finger sensing input devices. In: *Proceedings of UIST'03*. Vancouver, Canada: ACM Press, Nov. 2003, pp. 203–212.

[214] Ronald A. Rensink, J. Kevin O'Regan, and James J. Clark. To See or not to See: The Need for Attention to Perceive Changes in Scenes. In: *Psychological Science*, 8(5), Sept. 1997, pp. 368–373.

[215] Grégoire Richard, **Thomas Pietrzak**, Ferran Argelaguet, Anatole Lecuyer, and Géry Casiez. Studying the Role of Haptic Feedback on Virtual Embodiment in a Drawing Task. In: *Frontiers in VR*, 1(28), 2020, pp. 1–18.

[216] Grégoire Richard, **Thomas Pietrzak**, Ferran Argelaguet, Anatole Lecuyer, and Géry Casiez. Within or Between? Comparing Experimental Designs for Virtual Embodiment Studies. In: *Proceedings of VR 2022*. Christchurch, New Zealand: IEEE Computer Society Press, 2022.

[217] Hyocheol Ro, Seungho Chae, Inhwan Kim, Junghyun Byun, Yoonsik Yang, Yoonjung Park, and Tackdon Han. A dynamic depth-variable ray-casting interface for object manipulation in ar environments. In: *Proceedings of SMC 2017*. IEEE Computer Society Press, Oct. 2017, pp. 2873–2878.

[218] Jennifer A. Rode, Eleanor F. Toye, and Alan F. Blackwell. The fuzzy felt ethnography—understanding the programming patterns of domestic appliances. In: *Personal Ubiquitous Computing*, 8, 2004, pp. 161–176.

[219] Daniel Roth and Marc Erich Latoschik. Construction of the Virtual Embodiment Questionnaire (VEQ). In: *IEEE TVCG*, 26(12), 2020, pp. 3546–3556.

[220] Anne Roudaut, Abhijit Karnik, Markus Löchtefeld, and Sriram Subramanian. Morphees: Toward High "Shape Resolution" in Self-Actuated Flexible Mobile Devices. In: *Proceedings of CHI'13*. Paris, France: ACM Press, May 2013, pp. 593–602.

[221] Anne Roudaut, Eric Lecolinet, and Yves Guiard. MicroRolls: Expanding Touch-Screen Input Vocabulary by Distinguishing Rolls vs. Slides of the Thumb. In: *Proceedings of CHI'09*. Boston, MA, USA: ACM Press, Apr. 200, pp. 927–936.

[222] Nicolas Roussel, Helen Evans, and Heiko Hansen. MirrorSpace: Using Proximity as an Interface to Video-Mediated Communication. In: *Pervasive Computing*. Springer-Verlag, 2004, pp. 345–350.

[223] Quentin Roy, Yves Guiard, Gilles Bailly, Eric Lecolinet, and Olivier Rioul. Glass+Skin: An Empirical Evaluation of the Added Value of Finger Identification to Basic Single-Touch Interaction on Touch Screens. In: *Proceedings of INTERACT 2015*. Bamberg, Germany: Springer-Verlag, 2015.

[224] Diego C. Ruspini, Krasimir Kolarov, and Oussama Khatib. Haptic Interaction in Virtual Environments. In: *Proceedings of the Internationan Conference on Intelligent Robots and Systems, IROS '97*. Grenoble, France, Sept. 1997, pp. 128–133.

[225] J. Kenneth Salisbury, François Conti, and Federico Barbagli. Haptic rendering: introductory concepts. In: *IEEE Computer Graphics and Applications*, 24(2), 2004, pp. 24–32.

[226] Grégory Savioz, Mirosław Markovic, and Yves Perriard. Towards multi-finger haptic devices: A computer keyboard with adjustable force feedback. In: *Proc. International Conference on Electrical Machines and Systems*. 2011, pp. 1–6.

[227] Roelof Schellingerhout, Ad W., and Gerard P. van Galen. Texture information in tactual space perception. In: *Acta psychologica*, 99(1), 1998, pp. 93–114.

[228] Richard A. Schmidt and Timothy D. Lee. Motor Control and Learning: A Behavioral Emphasis – 5th Edition. Human Kinetics, 2005.

[229] Oliver S. Schneider, Ali Israr, and Karon E. MacLean. Tactile Animation by Direct Manipulation of Grid Displays. In: *Proceedings of UIST'15*. Charlotte, USA: ACM Press, Oct. 2015, pp. 21–30.

[230] Oliver S. Schneider and Karon E. MacLean. Studying design process and example use with Macaron, a web-based vibrotactile effect editor. In: *Proceedings of HAPTICS 2016*. Philadelphia, USA: IEEE Computer Society Press, Apr. 2016, pp. 52–58.

[231] Hasti Seifi, Farimah Fazlollahi, Michael Oppermann, John Andrew Sastrillo, Jessica Ip, Ashutosh Agrawal, Gunhyuk Park, Katherine J. Kuchenbecker, and Karon E. MacLean. Haptipedia: Accelerating Haptic Device Discovery to Support Interaction & Engineering Design. In: *Proceedings of CHI'19*. Glasgow, UK: ACM Press, May 2019, pp. 1–12.

[232] Ben Shneiderman. Direct manipulation: a step beyond programming languages. In: *IEEE Computer*, 16(8), Aug. 1983, pp. 57–69.

[233] Jamie Shotton, Andrew Fitzgibbon, Andrew Blake, Alex Kipman, Mark Finocchio, Bob Moore, and Toby Sharp. Real-time Human Pose Recognition in Parts from Single Depth Images. In: *Proc. CVPR '11*. IEEE Computer Society Press, 2011, pp. 1297–1304.

[234] Mel Slater. Measuring presence: A response to the Witmer and Singer presence questionnaire. In: *Presence*, 8(5), 1999, pp. 560–565.

[235] Mel Slater, Beau Lotto, Maria Marta Arnold, and María Victoria Sánchez-Vives. How we experience immersive virtual environments: the concept of presence and its measurement. In: *Anuario de Psicología*, 40, 2009, pp. 193–210.

[236] Mel Slater and Sylvia Wilbur. A Framework for Immersive Virtual Environments Five: Speculations on the Role of Presence in Virtual Environments. In: *Presence: Teleoper. Virtual Environ.*, 6(6), Dec. 1997, pp. 603–616.

[237] Marshall Smith, Douglas Craig, Nicole Herrmann, Erin Mahoney, Jonathan Krezel, Nate McIntyre, and Kandyce Goodliff. The Artemis Program: an overview of NASA's activities to return humans to the moon. In: *IEEE Aerospace Conference 2020*. 2020, pp. 1–10.

[238] Rajinder Sodhi, Ivan Poupyrev, Matthew Glisson, and Ali Israr. AIREAL: Interactive Tactile Experiences in Free Air. In: *ACM Trans. Graph.*, 32(4), July 2013.

[239] Hyunyoung Song, Hrvoje Benko, François Guimbretière, Sharam Izadi, Xiang Cao, and Ken Hinckley. Grips and Gestures on a Multi-touch Pen. In: *Proceedings of CHI'11*. Vancouver, Canada: ACM Press, May 2011, pp. 1323–1332.

[240] Anthony Steed. A Simple Method for Estimating the Latency of Interactive, Real-Time Graphics Simulations. In: *Proceedings of VRST 2008*. Bordeaux, France: ACM Press, 2008, pp. 123–129.

[241] Jürgen Steimle, Simon Olberding, Max Mühlhäuser, and James D. Hollan. PLink: Paper-Based Links for Cross-Media Information Spaces. In: *Extended Abstracts of CHI'11*. Vancouver, Canada: ACM Press, May 2011, pp. 1969–1974.

[242] Neal Stephenson. Snow Crash. Bantam Books, 1992.

[243] Robert M. Strong and Donald E. Troxel. An Electrotactile Display. In: *IEEE Transactions on Man-Machine Systems*, 11(1), 1970, pp. 72–79.

[244] Ivan E. Sutherland. A Head-Mounted Three Dimensional Display. In: *Proceedings of AFIPS '68, Fall Part I*. San Francisco, USA: ACM Press, Dec. 1968, pp. 757–764.

[245] Ivan E. Sutherland. The Ultimate Display. In: *Proceedings of the IFIP Congress*. 1965, pp. 506–508.

[246] Katsuhiro Suzuki, Fumihiko Nakamura, Jiu Otsuka, Katsutoshi Masai, Yuta Itoh, Yuta Sugiura, and Maki Sugimoto. Facial Expression Mapping inside Head Mounted Display by Embedded Optical Sensors. In: *Proceedings of UIST'16*. Tokyo, Japan: ACM Press, Oct. 2016, pp. 91–92.

[247] Emi Tamaki, Takashi Miyaki, and Jun Rekimoto. PossessedHand: A Hand Gesture Manipulation System Using Electrical Stimuli. In: *Proceedings of the 1st Augmented Human International Conference*. Megève, France: ACM Press, 2010.

[248] Kaoru Tashiro, Yuta Shiokawa, Tomotake Aono, and Takashi Maeno. Realization of button click feeling by use of ultrasonic vibration and force feedback. In: *Proceedings of WorldHaptics 2009*. Salt Lake city, UT, USA: IEEE Computer Society Press, Mar. 2009, pp. 1–6.

[249] Stuart Taylor, Cem Keskin, Otmar Hilliges, Sharam Izadi, and John Helmes. Type-Hover-Swipe in 96 Bytes: A Motion Sensing Mechanical Keyboard. In: *Proceedings of CHI'14*. Toronto, Canada: ACM Press, May 2014, pp. 1695–1704.

[250] Robert J. Teather, Andriy Pavlovych, Wolfgang Stuerzlinger, and I. Scott MacKenzie. Effects of tracking technology, latency, and spatial jitter on object movement. In: *Proceedings of 3DUI 2009*. Lafayette, USA: IEEE Computer Society Press, 2009, pp. 43–50.

[251] Feng Tian, Lishuang Xu, Hongan Wang, Xiaolong Zhang, Yuanyuan Liu, Vidya Setlur, and Guozhong Dai. Tilt Menu: Using the 3D Orientation Information of Pen Devices to Extend the Selection Capability of Pen-Based User Interfaces. In: *Proceedings of CHI'08*. Florence, Italy: ACM Press, Apr. 2008, pp. 1371–1380.

[252] Alan Mathison Turing. Systems of Logic Based on Ordinals. PhD thesis. Princeton, USA: Princeton University, 1938.

[253] Richard Q. van der Linde, Piet Lammertse, Erwin Frederiksen, and B. Ruiter. The HapticMaster, a new high-performance haptic interface. In: *Proceedings of Eurohaptics 2002*. Edinburgh, Scotland: ACM Press, July 2002.

[254] Lode Vanacken, Tovi Grossman, and Karin Coninx. Exploring te Effects of Environment Density and Target Visibility on Object Selection in 3D Virtual Environments. In: *Proceedings of 3DUI 2007*. Charlotte, NC, USA: IEEE Computer Society Press, Mar. 2007, pp. 117–124.

[255] Jo Vermeulen, Kris Luyten, Elise van den Hoven, and Karin Coninx. Crossing the Bridge over Norman's Gulf of Execution: Revealing Feedforward's True Identity. In: *Proceedings of CHI'13*. Paris, France: ACM Press, May 2013, pp. 1931–1940.

[256] Daniel Vogel and Géry Casiez. Conté: Multimodal Input Inspired by an Artist's Crayon. In: *Proceedings of UIST'11*. Santa Barbara, USA: ACM Press, Oct. 2011, pp. 357–366.

[257] Julie Wagner, Eric Lecolinet, and Ted Selker. Multi-Finger Chords for Hand-Held Tablets: Recognizable and Memorable. In: *Proceedings of CHI'14*. Toronto, Canada: ACM Press, May 2014, pp. 2883–2892.

[258] Thomas Waltemate, Irene Senna, Felix Hülsmann, Marieke Rohde, Stefan Kopp, Marc Ernst, and Mario Botsch. The impact of latency on perceptual judgments and motor performance in closed-loop interaction in virtual reality. In: *Proceedings of VRST 2016*. ACM Press, 2016, pp. 27–35.

[259] Peter Wegner. Why Interaction is More Powerful Than Algorithms. In: *Commun. ACM*, 40(5), May 1997, pp. 80–91.

[260] Peter Wegner and Dina Goldin. Interaction, Computability, and Church's Thesis. 1999.

[261] Sidney Weinstein. Intensive and extensive aspects of tactile sensitivity as a function of body part, sex and laterality. In: *The Skin Senses: Proceedings of the First International Symposium on the Skin Senses*. Ed. by Dan R. Kenshalo. Thallahassee, Florida: Charles C. Thomas, 1968, pp. 195–222.

[262] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Realtime Performance-Based Facial Animation. In: *ACM Transactions on Graphics*, 30(4), July 2011.

[263] Pierre Wellner. Interacting with Paper on the DigitalDesk. In: *Commun. ACM*, 36(7), July 1993, pp. 87–96.

[264] Daniel Wigdor and Dennis Wixon. Brave NUI World – Designing Natural User Interfaces for Touch and Gesture. Morgan Kaufmann, 2011.

[265] Bob G. Witmer and Michael J. Singer. Measuring presence in virtual environments: A presence questionnaire. In: *Presence*, 7(3), 1998, pp. 225–240.

[266] Hsin-Yun Yao and Vincent Hayward. Design and analysis of a recoil-type vibrotactile transducer. In: *The Journal of the Acoustical Society of America*, 128(2), 2010, pp. 619–627.

[267] Nick Yee and Jeremy Bailenson. The Proteus Effect: The Effect of Transformed Self-Representation on Behavior. In: *Human Communication Research*, 33(3), 2007, pp. 271–290.

[268] Takashi Yoshioka, Barbara Gibb, Andrew K. Dorsch, Steven S. Hsiao, and Kenneth O. Johnson. Neural coding mechanisms underlying perceived roughness of finely textured surfaces. In: *Journal of neuroscience*, 21(17), 2001, pp. 6905–6916.

[269] Shumin Zhai. Investigation of Feel for 6DOF Inputs: Isometric and Elastic Rate Control for Manipulation in 3D Environments. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 37(4), 1993, pp. 323–327.

[270] Shumin Zhai and Paul Milgram. Human performance evaluation of manipulation schemes in virtual environments. In: *Proceedings of IEEE VRAIS 1993*. IEEE Computer Society Press, 1993, pp. 155–161.

[271] Haimo Zhang and Yang Li. GestKeyboard: Enabling Gesture-Based Interaction on Ordinary Physical Keyboard. In: *Proceedings of CHI'14*. Toronto, Canada: ACM Press, May 2014, pp. 1675–1684.

[272] Jiajie Zhang and Donald A. Norman. Representations in Distributed Cognitive Tasks. In: *Cognitive Science*, 18(1), Jan. 1994, pp. 87–122.