Thèse de Cagan Arslan, Université de Lille, 2020

Université de Lille
Centre de Recherche en Informatique, Signal et Automatique de Lille
ED SPI Université Lille Nord-de-France

Cagan Arslan

Doing More without More: Data Fusion in Human-Computer Interaction

Faire mieux avec ce que l'on a déjà : La fusion de données en Interaction Homme-Machine

Thesis submitted in fulfillment of the requirements for
the degree of Doctor of Philosophy in Computer Science.

Defended on 28/10/2020

Committee:

Eric Lecolinet, Télécom Paris	Rapporteur
Marcelo M. Wanderley, McGill University	Rapporteur
Laurence Duchien, Université de Lille	Présidente
Sophie Lepreux, Université Polytechnique Hauts-de-France	Examinatrice
Jean Martinet, Université Côte d'Azur	Co-encadrant
Laurent Grisoni, Université de Lille	Directeur

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

ABSTRACT

The increasing variety of tasks which require human-computer interfaces result in the production of new and improved sensing devices and therefore causes the obsolescence of older technologies. In a world of limited resources, the production rate of new interaction devices is unsustainable. Sustainable design calls for re-appropriation of existing materials, so we need to design interfaces that are modular, re-usable, yet that allow new interaction techniques. We believe that combining the strength of different input devices through data fusion can enable powerful interactions while extending the lifespan of electronic materials. As the complexity of sensors increases, their combination presents new challenges and opportunities, notably in terms of computational power and user behavior, which we explore in this document.

We first explain how previous work conducted in different sub-domains of human-computer interaction fit into the data fusion perspective. From this perspective, we take all aspects of input devices into consideration to define the framework to which this thesis belongs. The first step consists of handling input devices to provide meaningful information to be fused, so we demonstrate how to go from a complex data source such as a camera stream, to a small, descriptive bit of information that enables lightweight fusion. Then, we separate the benefits of multi-sensor data fusion for interaction spaces into two categories; enriching the interaction space and extending the interaction space. Our contribution to the enriched spaces mainly focuses on musical interfaces where we propose a movement sonification application on a mobile device and a visual feedback mechanism, all by using a combination sensors. Further, we contribute a virtually extended surface for large display interactions using a hand-held touchscreen and examine the user's appropriation to the new interaction space.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

RESUMÉ

La variété croissante des tâches qui nécessitent des interfaces homme-machine aboutit à la production des nouveaux capteurs améliorés et provoque donc l'obsolescence des technologies plus anciennes. Dans un monde aux ressources limitées, le taux de production de nouveaux appareils homme-machine ne semble pas viable. La conception durable nécessite une réappropriation des matériaux existants, nous devons donc concevoir des interfaces modulaires, réutilisables, mais qui permettent de nouvelles techniques d'interaction. Nous pensons que la combinaison des puissances de différents périphériques grâce à la fusion de données peut permettre des interactions puissantes tout en prolongeant la durée de vie des matériaux électroniques. À mesure que la complexité des capteurs augmente, leur combinaison présente de nouveaux défis et opportunités, notamment en termes de puissance de calcul et de comportement des utilisateurs, que nous explorons dans ce document.

Nous expliquons d'abord comment les travaux antérieurs menés dans différents sous-domaines d'interaction homme-machine s'intègrent dans la perspective de la fusion de données. Dans cette perspective, nous prenons en compte tous les aspects des dispositifs d'entrée pour définir le cadre auquel appartient cette thèse. La première étape consiste à manipuler les périphériques d'entrée pour fournir des informations significatives à fusionner.Nous montrons donc comment passer d'une source de données complexe, telle qu'un flux de caméra, à une simple information descriptive qui permet une fusion légère. Ensuite, nous séparons les avantages de la fusion de données multi-capteurs pour les espaces d'interaction en deux catégories; enrichir l'espace d'interaction et étendre l'espace d'interaction. Notre contribution aux espaces enrichis se concentre principalement sur les interfaces musicales où nous proposons une application de sonification de mouvement sur un appareil mobile et un mécanisme de retour d'information visuelle, le tout en utilisant une combinaison de capteurs. De plus, nous contribuons à une surface virtuellement étendue pour les interactions sur grand écran à l'aide d'un écran tactile portable et examinons l'appropriation de l'utilisateur dans ce nouvel espace d'interaction.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Contents

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

List of Figures

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

One

Introduction

The number of HCI devices that surround us is increasing. Mobile devices such as smartphones, tablets and smart-watches are the "handiest", they are used by a vast number of consumers. These devices contain various types of sensors, cameras, touch-screens, microphones, accelerometers, proximity sensors etc. Every sensor comes with their own interaction space, that can be a surface (touch-screens, buttons) or a volume (cameras, microphones). Stationary devices such as computer vision systems (depth, rgb, infrared cameras), head mounted displays, wearable technologies are often used in public settings for both scientific and artistic applications. With the advances in the appropriation of these technologies in our daily life, we need newer and more powerful devices.

The constant need of newer material is not sustainable. Blevis [27] argues that HCI's focus on sustainability is often anthropocentric, that its requirements are derived from users, rather than the global good. He states that software drives the demand for new hardware, causes premature obsolescence of old materials. I agree that sustainable interaction design (SID), should take into account the finite primary materials when we design the software.

Producing new electronic devices requires precious metals such as gold, silver, copper, platinum and so on [16], but also rare earth elements (REE) such as neodymium, dysprosium and so on [96]. These elements have important properties such as heat resistance and conductivity and difficult to replace. For example, Royal Australian Chemical Institute [159] reports that in the next 20 years, the demand for Neodymium and Dysprosium will grow 700% and 2600% respectively and the only way to meet this demand will be through recycling. To some extend these important elements can be harvested from *urban mines* (e-

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

waste) [16], but given the fact that the recycling rate of REE is less than 1% [34], it would be wiser to keep the value in products to eliminate waste.

Even though the e-waste can be used creatively [94], most of the use is limited to decorative purposes. Re-appropriation of old devices by means of hacking rarely takes the energy consumption into consideration. This is why designing interaction which promotes *renewal* and *reuse* [27] is essential. The ecological limits might require more drastic measures [98] which involves legislation and public policy, thus sustainable human computer interaction (SHCI) will adopt more determined research aims to join forces in the battle against climate change. Yet, in this thesis we try to explore an approach that may be beneficial in achieving longevity of material.

Can we combine existing materials for a new interaction technique instead of waiting on new hardware to obsolete the older techniques? Can we conceive these systems in a way that the hardware can be detached and reused for other tasks? If so, how do the users adapt to such systems?

We believe these questions should be considered on technological and conceptual levels. On the **technological** level, we must figure out how to fuse the data from multiple sources. Sources may refer to sensors or more generally input devices, because composite devices blurred the distinction by employing multiple sensors in the same entity. Therefore, I use the terms interchangeably. Data fusion can be defined as the combination of multiple sources about specific phenomenon to obtain an improved quality of information. Humans combine their senses to increase their perception of the surroundings. Along with the traditional senses (sight, hearing, taste, smell, touch), humans posses other sensory modalities such as thermoception, proprioception, chemoreception to interpret and to react to stimuli. What considered a single sense is in fact a combination of multiple receptors. Our brain combines both eyes to see or both ears to hear, but the taste involves the nose and the tongue. Machine perception is also broadened by multi-sensor systems. Homogeneous sensor groups (analogous to two eyes) increase the reliability of the systems, while heterogeneous sensor groups (analogous to nose and the tongue) are usually employed to enhance awareness of the machines. If sensors observe the same physical phenomena, the fusion can be achieved in low level using raw data. If the sensors focus on different properties of the same observation, the fusion happens on semantic level once each sensor has made a preliminary decision. As the number and the variety of sensors increase in a system, managing the relations between sensors on different levels becomes a challenging task.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Data fusion is often used in human computer interaction. When the information sources provide simple outputs, the fusion goes unnoticed. The keyboard & mouse duo is the simplest yet most commonly used devices that combine user input. Holding down or releasing the control key while clicking sequentially on multiple items, changes the output of the action. Thus, the outcome depends on the fusion of the states of the devices. However, the data fusion becomes more interesting as the complexity of the sources increases. Devices such as cameras or touchscreens, provide rich information that should be processed before being included in decision making. Not only this requires careful analysis of processing resources and tools, it also raises interesting questions which influence user's behavior.

At this point we should discuss the **conceptual** level. Heterogeneous multi-sensor systems permits the creation of *compound interaction techniques* [176] that increase the size of the command vocabulary and offer users more nuanced control. As we gather data from different sources, we should also distribute interaction to different components. Spatiality is the key to the distribution of interaction. In proxemics , spatial relationships play an important role in how we physically interact, communicate, and engage with other people and with objects in our environment. Interfaces and users can react to the position and distance of its entities as either continuous movements, or as movements in and out of discrete proxemic zones. Seamless transition between these spaces is essential to create a consistent interaction experience.

The opportunities offered by the developing technologies can be explored with existing interaction techniques. Yet, new interaction techniques emerge through appropriation of the technology by communities of users [65]. The emergence of techniques is faster when it comes to multi-sensor systems, as the novelty does not stem only from individual technologies, but also from the intersecting interaction spaces. However, in order to appropriate the emerging technologies, we have to make them usable to an extent. How can we enable multi-sensor spaces for users to facilitate the exploration of the new techniques?

A similar research question was asked in a workshop about Blended Interaction Spaces [49]: *how can Blended Interaction Spaces facilitate seamless integration of individual creative sessions (e.g. using iPads and mobile phones) with collaborative ones (e.g. using wall sized displays in combination with iPads), hereby allowing for ideas to travel across platforms and contextual boundaries?"*. Here, "seamless integration" is the keyword that describes our intent. We are interested not only in the integration of individual and collaborative elements, but also in all modalities that open artistic and expressive opportunities.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

1.1 THESIS STATEMENT

In this thesis, I explore the new approaches data fusion brings to HCI by combining strengths of different input devices. My dissertation focuses on how to merge information obtained by multiple HCI devices not only by concentrating on the data but also by taking into account the interaction spaces. To this end, I start by explaining how to prepare input devices to extract meaning for lightweight fusion. Then I propose multi-sensor interfaces in two contexts. First, I consider enriching the interaction space by using the intersection of inputs and provide design spaces. Then I investigate extending the interaction space by concatenating inputs and examine how users adapt to compound interaction spaces.

1.2 THESIS ORGANIZATION AND CONTRIBUTIONS OVERVIEW

This thesis is organized in three contribution chapters which cover work issued across four publications parts of which are distributed in different subsections for consistency. Let us note that in order to allow the reader to understand the position of the contributions with respect to the state-of-the-art, I provide a literature overview in chapter 2.

In chapter 3, I present the first step to a reliable data fusion that is the preparation of sensors. This involves extracting meaning from an input stream of data which is a challenging step when the abstraction level of the sensors increases. First, I illustrate the concept with the extraction of motion information from RGB cameras. Section 3.2.1 presents extracting meaning for continuous gestures and is part of our work in [9]. Section 3.2.2 explains using the motion information for very simple descriptions of discrete gestures and is part of our work in [10]. Then, in order to put individual source descriptors in the same context, I explore a seamless calibration technique for depth cameras which observe a common volume.

In chapter 4, I argue that features can be combined at the intersection of devices to obtain richer interaction techniques. First, I illustrate the fusion of simple motion information with touch input in a mobile context. The use of touchscreen allows easy filtering of the motion information and decreases the computational time while opening new expressive opportunities for artistic expression. The mobile application was presented in [9]. Then in section 4.3, I show how depth

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

sensors can complement other sensors to provide visual feedback and guide the users during interaction. This work was a subsection of our work in [24]. Finally I present a design space for local precision refinement with depth cameras which operate at different proximities.

In chapter 5, instead of limiting the interaction to the intersection of devices, I explore the interaction space which stems from the concatenation of a mobile touchscreen with its surrounding 3D volume. Then I present a user study to test its capacity for commanding a large display using extended continuous gestures that start on the touch surface and end in mid-air. This chapter corresponds entirely to our work in [11]. Finally, I provide guidelines for future work in the conclusion.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Two

Literature Overview

Human-computer interaction communities have been using data fusion to achieve various tasks. It is not a concept which originated from our domain, yet its impacts on our work are visible throughout the literature. To define the context of this document, in this chapter I briefly present the beginnings of data fusion, then I argue about how different sub-domains of HCI regularly use data fusion, consciously or unconsciously. Moreover, I present a taxonomy of input devices to underline the specific elements to consider for the framework of our contributions that is heterogeneous systems. Finally, I provide examples of existing interactive systems that fit into the framework we created.

2.1 Brief history of multi-sensor data fusion

The most referred [40] [73] [189] conceptualization of fusion systems is the JDL model. This definition of data fusion was provided by the Joint Directors of Laboratories (JDL) workshop [181]: *A multi-level process dealing with the association, correlation, combination of data and information from single and multiple sources to achieve refined position, identify estimates and complete and timely assessments of situations, threats and their significance*.

JDL model (Figure 2.1) is conceived for military applications and consists of five levels. Source preprocessing selects the sensors that are most crucial to current situation and allocates the processing sources. In the object refinement level, sensor data is transformed into a consistent set of units and coordinates and the captured object's attributes are refined. Situation refinement focuses on relational information to determine the meaning of a collection of entities.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

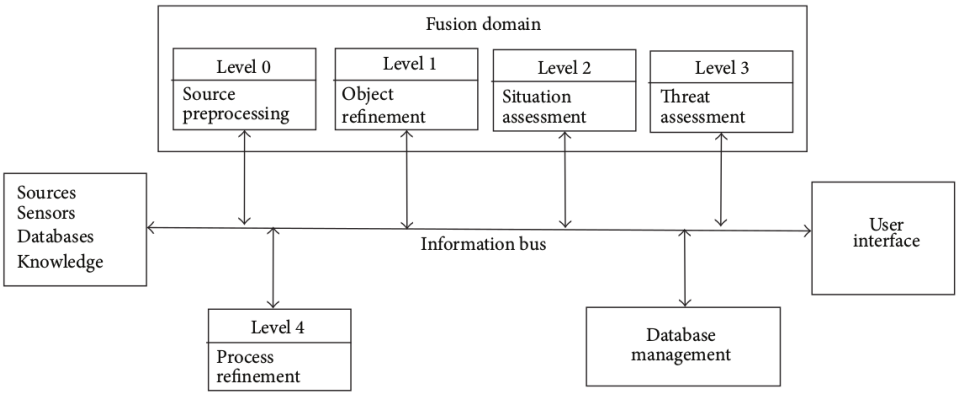


Figure 2.1: JDL fusion model. Extracted from [40]

In other words, it addresses the interpretation of data. Threat refinement level predicts the enemy threats by reflecting the ongoing situation into the future. Process refinement level encompasses the fusion domain because it supervises other processes. It identifies what information is needed to improve the multilevel fusion product.

The right side of the Figure 2.1 shows the human computer interaction function for fusion systems. HCI allows human input such as commands and information requests. In general, HCI incorporates not only multimedia methods for human interaction (graphics, sound, tactile interface, etc.),but also methods to assist humans in direction of attention, and overcoming human cognitive limitations.

However JDL model is considered too specific for military applications. Therefore, others [56] [73] [50] [108] tried to extend the model to respond to general issues of data fusion. Castanedo [40] provided a classification of data fusion techniques to highlight its main steps for broader application areas.

Durrant-Whyte [56] proposed classification of data fusion techniques by the relations between data sources. *Complementary* data fusion occurs when the information provided by the input sources represents different parts of the scene and so that it be used to obtain more complete global information. *Redundant* fusion takes place when two or more input sources provide information about the same target and they can be fused to increase the confidence level of the decisions. *Cooperative* fusion occurs when the provided information is combined into new information that is typically more complex than the original information.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Dasarathy [50] proposed a classification depending on the input-output relations. *Data in-data out*: This type of data fusion processes raw input and outputs raw data to achieve more reliable or accurate data stream. *Data in-feature out*: the data fusion process gathers raw data from the sources to extract features or characteristics that describe an object of interest in the environment. *feature in-feature out*: both the input and output of the data fusion process are features. Thus, the data fusion aims to improve the quality of features, or to obtain new features. *feature in-decision out*: this level requires features as input and outputs a set of decisions . According to Dasarathy, most of the systems that perform a decision based fusion on a sensor fits into this category, however it often requires that the raw data is transformed into a set of features first. *Decision In-Decision Out*: This type of classification is also known as decision fusion. It fuses input decisions to obtain better or new decisions as output.

Luo et al. [110] provided classification based on the abstraction levels. *signal level*: directly addresses the signals that are acquired from the sensors. *Pixel level*: operates at the image level and could be used to improve image processing tasks . *Characteristic level*: uses features that are extracted from the images or signals. *symbol level*: at this level, symbols represent information bits; this level is also known as the decision level. This classification is very similar to Dasarathy's as it is mainly interested in the separation of raw data - features - decisions.

Castenado [40] contributes another type of classification which is based on the node architecture of fusion mechanism. *Centralized architecture:* All the input sources send raw data to the central processor where all the fusion processes are handled. *Decentralized architecture:* in this type of architecture every node has its own processing capability and there is no central processing unit. Information received from adjacent nodes is fused autonomously in the local node and may be transfered to the others. *Distributed architecture:* First, each node processes independently the raw data it receives from the corresponding input and sends the information to the fusion node. The fusion node processes the information it receives from the independent node. *Hierarchical architecture:* it is a combination of decentralized and distributed fusion. Fusion is performed at different levels of a hierarchical tree; some nodes only process and pass the information while others fuse the information received from other nodes.

The aforementioned data fusion classifications apply to multi sensor systems in human-computer interaction. For example, multi-modal interfaces usually achieve data fusion at the feature level, decision level, characteristic level, symbol level. Yet, they do not respond to various aspects of HCI applications such as user behavior around the sensors.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Wald [177] says: *A search for a more suitable definition was launched with the following principles. The definition for data fusion should not be restricted to data output from sensors (signal). It should neither be based on the semantic levels of the information. It should not be restricted to methods and techniques or architectures of systems, since we aim at setting up a conceptual framework for data fusion.*

In the classical sense, the term fusion has not been used to cover the possibilities offered by merging or combining input sources. However as we move towards a ubiquitous future, we need to broaden this definition to surpass just a mathematical function to operate on two sets of arrays. We need to be able to describe a more general approach that is open to novelty and different interpretations in order to accommodate the combination of more complex interaction devices that did not exist when the JDL model was proposed.

Wald identifies two main issues the classical models do not adress. The first one is *topological issues*. These issues include the spatial distribution and placement of sensors, the processing hardware which is capable of decision-making and the communication protocols between the input devices. The second one is the *processing issues* that arise from the problems such as to select fusion methods suitable the project specifications and the dynamic features and measurements required to complete the objectives.

These concerns are also valid when we use multi-sensor interfaces. As we move from military applications to civilian ones, the objectives might change, but topological and processing issues still construct the core of the reflexion process when it comes to interaction design.

2.2 MULTI-SENSOR INTERACTIVE SYSTEMS

Multi-sensor human computer interfaces are not new. This implies that the fusion of the information provided by these sensors has already been done in various forms. Even without a formal definition, the design process involves the identification of the input sources, feature extraction, merging the information and decision making.

Various research communities in HCI use the buildings blocks of the JDL model and the aforementioned fusion techniques without explicitly branding it

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

data fusion. Multimodality and ubicomp are probably the first keywords that come to mind while discussing multi-sensor interfaces. It is also possible to find elements of data fusion in AR/VR, collaborative environments, tangibles, prototyping processes and so on. But these topics are not exactly orthogonal; for example, a lot of ubiquitous environments are multimodal or some virtual spaces are collaborative. Nevertheless, in order to put things in perspective, it is beneficial to discuss these different dimensions from a data fusion point of view.

2.2.1 MULTIMODALITY

Multimodality is a term which is widely and generously used in a variety of fields such as psychology, media, linguistics and semiotics. More importantly it has a diverse use in computer science and has different definitions even within the HCI community. In its simplest form, multimodal is *communicating in a variety of ways* [85]. Humans communicate through multiple channels and interpret information by cognitively fusing the channels. Machines also fuse communication channels to interpret data. As we browse through different fields, the definitions become more specific to respond to the research objectives. Detailing the definition, one step forward, a modality corresponds to each acquisition method in multiple experiments or subjects, under different conditions, using any instrument [103]. It is fundamentally a mathematical problem that requires analysis of several datasets which can interact with each other.

Early on, HCI community addressed the multimodal challenges [132] [131] [6]. Nigay and Coutaz [131] define a mode as *a state that determines the way information is interpreted to extract or convey meaning*. In that sense, the difference between multimedia and multimodal is that rather then conveying information on different channels of communication, a multimodal system can interpret abstracted information and it strives for meaning. In a broader sense, interfaces possess different interaction modes. As we move from unimodal to concurrent and multi modal interaction [162], we switched from using only a keyboard to *richer* modes such as voice, gaze, facial expressions, brain activity and so on.

Sharma and Pavlovic [162] distinguish human action modalities and computer sensing modalities. They cite typing, handwriting, pushing and clicking, gloved hand gestures, speaking, body and head movement, free hand gestures, facial expression, eye movement, hand pressure and brain-activity as human-action modalities. On the other hand they separate computer-sensing modalities into position and motion sensing, audio sensing, visual sensing, tactile and force

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

sensing, and neural sensing. This list is non-exhaustive today, but they provide a mapping between the two types of modalities via devices. For example a computer mouse converts pushing and clicking human action modality to position sensing or a camera converts eye movements to video sensing.

Bernsen characterizes the 'Modality Theory' as identifying the input-output modalities which constitute an optimal solution to the representation and exchange of the information that needs to be exchanged between user and system in a specific context. [21]. Here modalities correspond to *representational modalities*, [22] [79] that are ways of representing information to humans or to machines by different media such as graphics or acoustics. To quote Bernsen, "Multimodal modalities are combinations of unimodal modalities", thus using our claim that the definition of data fusion should be extended, multimodality implies data fusion.

Even though the term multimodality is used in various ways in HCI, I believe above definition still holds today. In the rest of this document, I use the term accordingly and precise it when it is used in differently in other works.

Many believe that the origin of multimodality in HCI is traced back to Bolt's *put-that-there* [28] that combined hand gestures, speech and gaze to interact with virtual objects. On a large screen, the user points at an object and voices commands such as "create a blue square *there*". Voice activated keyword *there* signifies the positional modality which uses hand gestures to determine the location on the screen. Since *put-that-there* there has been other works that combined hand pointing, speech and gaze [173] [145] which also address conflict of these modalities using a decision matrix-based multimodal fusion [191].

Speech and gesture have been popular for humanization of communication with devices. Battleview [23] combined hand gestures and speech to control a virtual battlefield. As in the case of Battleview, hand gestures have been detected with the help of cameras. An earlier example is the MUSIIC Architecture [91], which was a system design to help people with disabilities, where speech signals were complemented with pointing gestures.

The reason why hand gesture-speech-gaze are so popular in HCI is because they are relatively easy to track but more importantly they are the three modalities we use as humans in our daily conversations. We should also add facial gesture recognition which is becoming more and more robust [4]. This is why conversational user interfaces (CUI) should be multimodal [156]. Non-verbal signals are intuitive, they are essential to communicate interactively and can be effectively fused with speech signals to increase the reliability of decision making.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Other than the gesture-speech-gaze multimodality, the widened definition of multimodality illustrates the combination of interaction modes which has interesting applications. Multimodal authentication relies on more than one input method to protect the privacy of users. GazeTouchPass [93] uses a combination of eye tracking and tactile input to protect the users from shoulder surfing attacks. Friedman et al. proposed a fusion architecture on biometric sensors to combine keystroke, mouse movement and web browsing for a multimodal active authentication method. Vielhauer et al. [48] also proposed a fusion strategy, but for speech and handwriting modalities.

Haptic modality has also been complemented with speech in multimodal interfaces [193] and found a place in the literature, for various tasks such as referring to objects [104], navigation through image databases [90], education [133] and so on.

Generally, the fusion strategy for multimodal interfaces is to process information separately for each modality and to combine the decisions [160]. As a result, the fusion is usually *complementary* or *redundant* according to Durrant-Whyte's classification. As previously mentioned, this indicates that individual features and decisions should be fused in a context-dependent manner.

2.2.2 Ubiquitous computing

Ubiquitous computing is an area in which multimodal information flowing from the environment are fused together to obtain meaningful data. In an ubiquitous environment, the users can interact with devices of varying size and forms, using different modes, and often simultaneously. In environments that are densely populated with input devices, decisions obtained through different interfaces need to be fused to make sense of the ubiquitousness.

Mark Weiser's vision for the 21th century computer [179] was ubiquitous. He anticipated the replacement of writing surfaces with computationally capable, interconnected visual displays. Even though today ubiquitous environments a variety of devices, especially with smartphones, the initial vision stands. If we imagine an environment which only consists of a hundred small displays that support only the handwriting modality, it is ubiquitous but not multimodal. Thus data fusion can be present in multimodality and ubiquitous computing independently.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

When Salber proposed an HCI agenda in ubiquitous computing [155] , they defined ubicomp as *an attempt to break away from the current paradigm of desktop computing to provide computational services to a user when and where required.* According to Salber, an ubiquitous system includes a series of computing devices and a a series of tasks enabled by the ensemble of the interconnected devices. One of the many strengths of an ubicomp system is its ability to distribute a complex acquisition task which would otherwise be very challenging. Therefore, if the acquisition task is distributed, acquired information should be reintegrated through fusion.

One of the most evident interests of ubicomp is the smart rooms/cities which consist of distributed interfaces. The intelligent room project (1997) claimed that they aim to pull the computer out into the real world of people [32]. To do so, they decorated a room with video cameras and microphones so that the user can interact with other humans or with the room itself without being invasive to the user. As in the previous subsection, gesture and speech modalities are combined together to give the user the opportunity carry out natural tasks.

Another goal in ubicomp is seamlessness i.e. making the computers and capturing devices invisible to users [182]. Of course, this does not mean that the physical devices are invisible; it rather indicates that the transition between capturing devices and modalities do not require the user's consciousness. Consequently, data fusion plays an important role to assure the seamlessness. the illusion of an ubiquitous environment is broken if the user needs to mind the presence of a sensing device even if it is invisible to the naked eye.

LightSpace [182] builds on this idea by providing interactivity across many surfaces in the environment and in the space between surfaces. It allows the user to transfer tasks between screens in an intuitive manner. It also enables interesting interaction methods such as touching two displays at the same time to connect them. Light Widgets [58] combines multiple cameras and processors to transform every-day objects into interactive widgets that allows users to control them with hand gestures. It uses a low-level fusion architecture to triangulate the cameras to detect user's interaction with a widget.

In ubicomp environments, detecting user's intention to control a specific interface is problematic. It is essentially a proxemics problem in a non-verbal environment because the user's distance and orientation with respect to one device defines their intention to use it. Proxemic interaction [17] aims to resolve this problem by regulating implicit and explicit interaction techniques by transitioning between proxemic regions.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Finally, the advances in the hand-held and wearable technologies reshaped the mobility of interaction. As a result, ubiquitous environments are not limited to fixed locations. Smartphones in tomorrow's world -if not today- will be used as universal controllers to interact with omnipresent interfaces [26]. This will require data fusion to be achieved in a decentralized manner, in the server side and client side. Internet of things will facilitate crowdsourcing of information for environmental and humanitarian purposes [168]. However, within the scope of this thesis, I approach the smartphone based fusion from a more interactive point of view in section 2.4.2.

2.2.3 Composite input devices

Composite input devices consist of multiple sensors but appear as a singular apparatus. They can either transmit each sensor data individually, or perform on-board data fusion to transmit one relevant message. Smartphones, smartwatches, smart gloves, weemote are examples of such devices.

An important technology which enables composite devices is the Inertial measurement unit (IMU). IMUs are composed of an accelerometer, a gyroscope and a magnetometer, and are mainly used in devices to measure velocity, orientation, and gravitational force. They are very versatile in measuring an object's state in 3D coordinates.

The three units can be combined to obtain absolute orientation, where gyroscope angle is corrected by accelerometer and magnetic compass, but they are vulnerable to accumulated error [154]. IMUs are used in various applications from navigation to augmented reality systems [3]. Once fused, the totality of measurements allow 9 degrees of freedom (9-D0F) and can be integrated easily in hand-held devices. Therefore data fusion exists on IMU level in countless input devices.

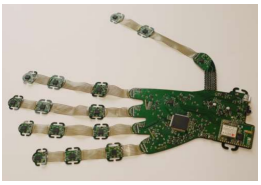


Figure 2.2: The IMU Smart Glove rev 2, extracted from [134]

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Considering IMUs are small and do not require external tracking, they are widely used in wearables. Smart gloves take advantage of multiple IMUs (Fig.2.2) to track finger orientation and hand poses [134] [113]. Even 3D arm postures were successfully tracked with the IMUs inside a smartwatch [163]. These examples show that IMUs are already powerful as a stand-alone technology to rely one.

The strength of IMUs is amplified when they are combined with other devices. Especially depth cameras have proven to form a powerful couple with IMUs. A detailed survey [46] shows that fusing information from depth and inertial sensors leads to more robust recognition. Commercially available depth cameras and wearable inertial sensors are both low-cost, widely available and more importantly they both provide 3D data. As a result it becomes possible to obtain tracking performances comparable to that of an expensive MoCap system. For example, Rodrigues et al. [150] integrate multiple Microsoft Kinects with four Shimmer IMUs and combine quaternions from all acquisitions to create a markerless motion analysis device which can compete with a VICON system.

Other than motion capture, IMUs can complement input devices for other applications, such as a person identification system. [43]. The necessity for accurate position and orientation tracking makes it so that inertial sensors are often used in augmented/virtual reality applications. Shall et al.'s work on multi-sensor fusion for outdoor augmented reality [157] is a perfect example of the combination of visual tracking, IMUs and Global Positioning System (GPS). On a handheld AR device They use the GPS for global outdoor registration, the IMU for the orientation estimates and a camera to compensate the drift of the IMU. In fact, using a camera-IMU couple is now a common practice to correct physical quantities such as the speed, raw and pitch angles and other inertial measurements [118] [122].

2.2.4 TANGIBLES

Tangible user interfaces are designed to give the user the opportunity to physically grasp and manipulate. Some tangibles are used individually, yet there exists interface where more than multiple tangible objects are manipulated in the interaction. Data fusion in tangibles may be separated in two types: Logical and Technical (Sec.. 1). In the case of *passive* tangible interfaces, the fusion is achieved on logical level. The information related to tangibles tracked by markers are combined for decision making. A good example to such a tangible interface is the ReacTable 2.3.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Figure 2.3: ReacTable

ReacTable [86] is a tabletop tangible userface used as a modular synthesizer which supports multiple users. It tracks the identifiers of tangible objects, their position and orientation on the table surface to and uses their topological structure to create complex sound waves. Even though there is one active input device (a camera placed underneath a translucent table), the information related to each tangible is fused according to a simple set of rules. XPaint table [55] also uses RFID-tagged tangible objects to draw on a standard WIMP interface. Output of the recognizers for the tangible objects on the table are fused in a centralized architecture. Tangible Bots [138] are motorized tangibles capable of giving haptic feedback to users. Even though they are described as active tangible objects in the original paper, as the tangibles are tracked by a camera using fiducial markers, from our point of view, they are handled by logical fusion.

In the case of active tangibles subjected to technical fusion, a sensor network provides individual information on the components involved in the interaction. Siftables [120] are compact displays that are capable of sensing other Siftables and basic actions and they can transmit the sensed information to other Siftables or to a computer. Thus, spatial configurations of the tangibles are used to complete different tasks. Remote Furnitures [63] consist of two rocking chairs mounted with tilt sensors and a motor. The chairs interact with each other in both direction.

To conclude, whether tangibles are subject to logical or technical fusion, they are strong tools to give haptic feedback and visualization possibilities and enable collaborative interaction.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

2.2.5 PROTOTYPING

Every concept or product in development is at some point a prototype. It can be used to illustrate a simple idea that emerged during a brainstorming session, or can be the early materialized version of something more complicated. Before the final product emerges, often an uglier, bulkier version is produced with available materials.

Toolkits offer valuable support for rapid prototyping in order to save time and effort. Developers can explore APIs to facilitate access to input sensors and call specific functions to obtaining meaningful information from a stream of raw data. For example, XPaint table [55] I mentioned in the previous subsection uses Hephais toolkit for easy integration of recognizers for creating multimodal interfaces. The toolkit uses a finite state machine paradigm to manage fusion of modalities.

Toolkits are especially important for cross-device interaction. However cross-device interaction requires synchronizing devices and designing the communication protocol is time consuming. Huang and Kong [80] developed a toolkit for combining a horizontal screen and multiple smartphones. Their toolkit respond to issues such as data transfer, authentication in multi-user interaction and interface composition. Conductor [74] is a toolkit which focuses on cross-device interaction between tablets. It distributes tasks across multiple devices and manages sessions through Websockets. MilkyWay toolbox [100] enables collaborative use of mobile phones via Bluetooth and WiFi and facilitates prototyping mobile-based interactions.

Multi-sensor data fusion is also a valuable asset to demonstrate interaction techniques that are ahead of their time in terms of technology. Technology and interaction techniques influence each other as sometimes a new technology allows a new technique and sometimes a new techniques require a new technology to be developed. Often a new technique might require multiple sensors which are not bundled in a single device at the time of its emergence. For example, WatchConnect [78] is a toolkit which has an hardware extension and which uses sensor mappings to demonstrate concepts such as on and above smartwatch interaction and interacting with other surfaces via smartwatches. Proximity Toolkit [116] solves the problem of interpreting proxemic relationships in a ubicomp environment. It enables proxemic interaction between mobile devices, interactive surfaces, ordinary objects and users to facilitate the design of new interaction techniques.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

To sum up, data fusion is an invaluable tool to explore new interaction techniques in areas that are not fully investigated because of technological constraints. Providing toolkits and frameworks is an essential contribution to help researchers develop new concepts which will influence new technologies.

2.2.6 Missing link

The precedent examples show that data fusion is present in different fields inside the HCI community; we combine, we merge, we integrate constantly. For some, it is a mathematical tool, for others it is the connecting node for different modalities. The approaches to use multi-sensor systems intersect, overlap and are sometimes insufficient to provide a framework to respond to cross-disciplinary requirements. Yet, defining a framework which encompasses all the examples is very challenging without arriving to a point where one might claim that everything is a result of fusion.

The common element is the desire to produce interfaces that are more reliable, more accessible and richer in terms of interaction. While we try to unify data in one place to make better decisions, we started distributing acquisition tasks across devices, which in turn resulted in spreading gestures around the physical space.

It is possible to retreat and provide a classification from a wider angle which takes into account different input technologies, modalities, spaces and user behavior. In the next two sections, I try to demonstrate another approach to multi sensor environment by discussing the capabilities of input devices and their implications on interaction techniques.

2.3 Taxonomy of input devices

Our perspective of data fusion in HCI requires both technological and spatial characteristics of input devices to be considered in the design process of multi-sensor systems. Technological aspects include the the properties sensed by the devices, the complexity of the data acquired, while spatial characteristic include the mobility of the devices and the distance at which users interact with them. The interaction spaces and the gestures enabled by the attributes of sensors included in devices have an effect on the best way to combine said devices. As a result,

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

user behavior reshapes around the novel opportunities provided by the combined interaction spaces.

For that reason, before we start discussing the interaction spaces, we should discuss the characteristic sensors which constitute the core of input devices, that are commonly used in HCI applications.

2.3.1 MATERIAL CONSIDERATIONS

As the machines continue to complete more complicated and diverse tasks, they also demand more detailed and varied user input. As a result, the variety of input devices used in human computer interaction is still growing. From the earlier days, researchers tried to classify the input devices to make sense of this variety. The first mention of the physical component in describing the user interface aspects of interactive computer systems appears in Moran's Command Language Grammar [124], however they do not discuss it further. Buxton [35] proposed a taxonomy of continuous hand controlled input devices according to the physical properties and the number of dimensions they sense.

In figure 2.4 primary partitions of the matrix are delimited by solid lines. The rows for position and motion sensing devices are subdivided in order to differentiate between transducers which sense potential via mechanical vs touch-sensitive means. The sub-columns exist to isolate devices whose control motion is roughly similar. However, given the time of its publication, Buxton's taxonomy does not include many advanced input devices such as cameras or GPS.

Card et al. [36] presented a taxonomy that goes beyond Buxton's by including non continuous input devices (Fig 2.5). In this taxonomy, a device is represented in the figure as a shapes connected together. The shapes differ according to their inclusion in other taxonomies, but has no importance at this moment. Each shape represents a transducer in the device and each line indicates a the regrouping of the transducer in one device. The black lines indicate the cross product of two transducers, and called merge composition. For example, the position of the finger on the tablet is the cross product of the positions on the x and y axes. Dotted lines indicate that the transducers are placed at different locations on the device and is called the layout composition. On a radio, selection and volume knobs both rotate on the same axis, but placed separately on the device. Arrowed line indicate that the output of one device enters the input of another and is called connect composition. To illustrate, they give the example of a simple set of radio controls

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

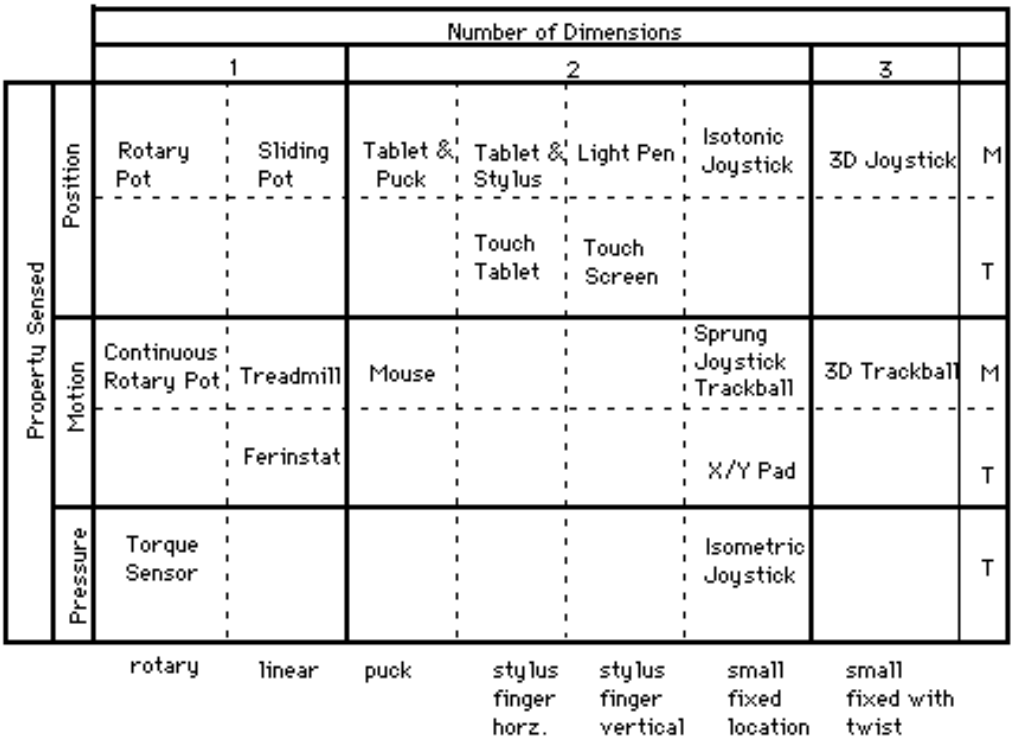


Figure 2.4: Buxton's taxonomy. Extracted from [35]

consisting of volume, selection and station knobs and a slider 2.6. Transducers are placed on the right or left side of a case to indicate their continuousness. For example, the volume knob is placed in the right side of the rZ case because it can turn around the Z axis and can take continuous values. The selection knob is placed on the left side of the same case because it only takes discrete values. The station knob has no determined range and turns infinitely. It controls a slider that moves on the the x-axis, hence called a relative continuous input. As a result, Card's taxonomy regroups input devices according to their composition in terms of transducers they include. However, we are also interested in regrouping devices by not only the intrinsic properties but also their impact on the outside world.

Above taxonomies provide an understanding of elementary input devices, yet do not include more complex devices such as cameras, microphones, inertial measurement units or EMG units. Truillet [171] defines the technologies which compose these devices as physical contextual sensors. A contextual sensor has a capture unit and a calculator unit, which altogether translate physical events into logical variables. A device and a contextual sensor are two different things. A classical device outputs an action performed by a user and the output data can be

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

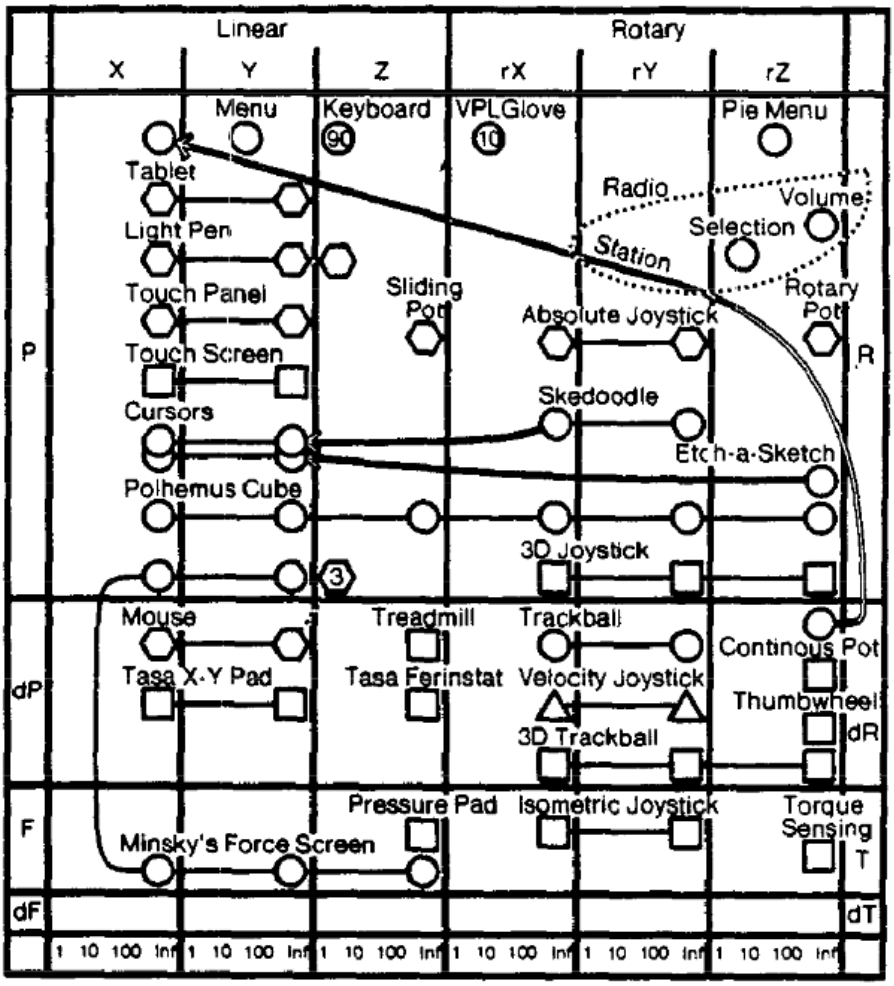


Figure 2.5: Card's taxonomy. Extracted from [37]

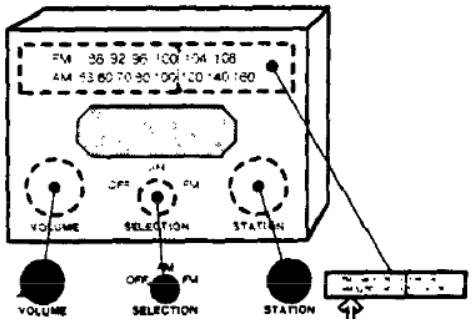


Figure 2.6: Card's radio. Extracted from [37]

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

directly used. However, the output of a contextual sensor must be processed before being interpreted in consonance with the application context. Truillet's taxonomy is divided into two parts; physical context and the human factors (Fig. 2.7). In the human factors branch, sensors can be redundant in the leaves of the tree. This arises from the fact that contextual sensors are heavily context-dependent. A camera stream can be interpreted to obtain physiological clues from facial gestures, or simply processed with geometrical cues to obtain a person's posture. In these two distinct contexts, merging camera stream with other modalities would be quite different.

In order to clarify the concepts used in the rest of this thesis, we provide a taxonomy of components of input devices from the data fusion point of view. Some of the dimensions of our taxonomy (Table 2.1) feature in existing classifications. We combine these dimensions and complete them with additional examples to demonstrate the variety of device characteristics which can be combined.

Properties sensed

One of the commonly used attributes when choosing and input device is the property sensed [35] [36]. The nature of the property that is measured has an impact on the system's ability to support interaction modes. A device can measure

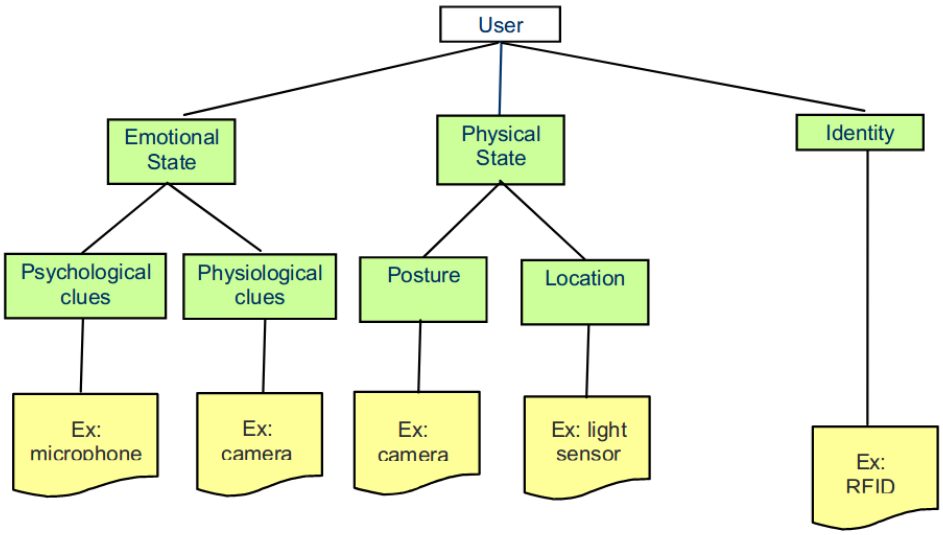


Figure 2.7: Human factors in Truillet's taxonomy. Extracted from [171]

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Dimension	**Values**	**Components**
Property Sensed	Position	Buttons, switches, touchscreen
	Velocity	Mouse, trackball
	Acceleration	Accelerometer
	Intensity	light, temperature, pressure
Dimensions sensed	1D	Buttons, sliders, potentiometers
	2D	Mouse, touchscreen, joysticks, stylus
	3D	3D trackball, RGB+D cameras, IMUs
Abstraction capacity	Low	Single optical sensors: photo-diode, UV-Sensor
	Medium	Motion: Accelerometer, angular sensor
		Location: GPS, active badge systems
		Bio Sensors: Pulse, skin resistance
	High	Touch-screens
		Audio: microphones
		Optical sensor arrays: Cameras
Number of Dimensions (components of ...)	Stationary	touch walls, MoCap
	Mobile	phones, tablets
	Wearable	smartwatches, VR headsets
	Self-Moving	imagery drones, robots
Proximity	Contact	Button, touchscreen
	Near	Leap motion, frontal mobile cameras
	Far	Kinect, VICON ...

Table 2.1: Our taxonomy

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

one or more properties, and properties may be derived from each other. In that case, one can argue that a device can sense more than one property, but there exists a fundamental property a device measures. There are four main properties measured by sensors.

NUMBER OF DIMENSIONS SENSED

Another useful property when selecting an input device is the number of dimensions sensed. It is strongly related to the interaction medium and the tasks to perform. Number of dimensions sensed are different than degrees of freedom in the sense that they are related to user gestures in the physical world. 1D sensors are intuitive to change only one control such as volume or brightness. 2D sensors such as a mouse are ideal for interacting with traditional desktop interfaces, while 3D sensors enable manipulation of physical objects or interacting in virtual reality interfaces.

ABSTRACTION CAPACITY

Complexity of a sensor is defined by the amount of information provided by the sensor and the computational resources needed to process it. Abstraction capacity is the ability of a device to provide a meaning at any level of abstraction. A button has a low abstraction capacity because it informs only about the on/off state. A camera on the other hand, requires a large bandwidth, can provide information at different levels and usually go through heavy processing steps to be fused with other sensors.

In computer applications, context is acquired either explicitly by requiring the user to specify it, or implicitly by monitoring user and computer-based activity. The contextual sensors incorporate either complex information about the surroundings, or information about user's physical or psychological state into the interaction. Schmidt et al. [158] classify the contextual sensors according to the technologies they use (Table 2.1).

At a high level of abstraction, context information depends on more on the situations rather than physical conditions. High level abstractions comprise of features and decisions that are used to determine the context. Commonly desired contextual information includes location, proximity, time, individuals,

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

social interactions, semantics of dialogues, description of the surroundings and so on [186].

DEGREE OF MOBILITY

The way we interact with the devices change with respect to their capacity to follow the user, thus the degree of mobility describes the usage scenarios in a separate dimension of our taxonomy. Rieger et al. [148] argued about mobility of the sensors in their taxonomy for app-enabled devices. They separated the sensors into four classes (Table 2.1): *Stationary* devices are installed in fixed locations, *mobile* devices can be transported freely, *wearable* devices differ from mobile devices in the sense that they are implicitly moved, *self moving* devices are capable of calculate their directions. The same device can be used in different mobility settings, for instance, a microphone can be mounted on a stand or on the user.

PROXIMITY

Proximity dimension of our taxonomy relates to the physical distance of the user to the device which captures their actions. Even though the distance is a continuum, we discretize it into three classes. *Contact:* user is directly in contact with the device, they either touch or handle the sensor. *Near:* the device is in the arm's reach from the user. These devices capture a part of user's body. *Far:* the device is further away from the user and provides a general description of the person or the environment.

2.3.2 GESTURAL CONSIDERATIONS

In the previous subsection, we studied the taxonomy of input devices to choose the necessary material resources. Additionally, we should consider the gestural necessities to identify the elements for fusion. There is a close relationship between *what the user is trying to do?* and *what technologies does the user need to do it?*. According to Karam's categorization of gesture styles [89], users intend communicate with computers by means of 5 gesture classes:

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

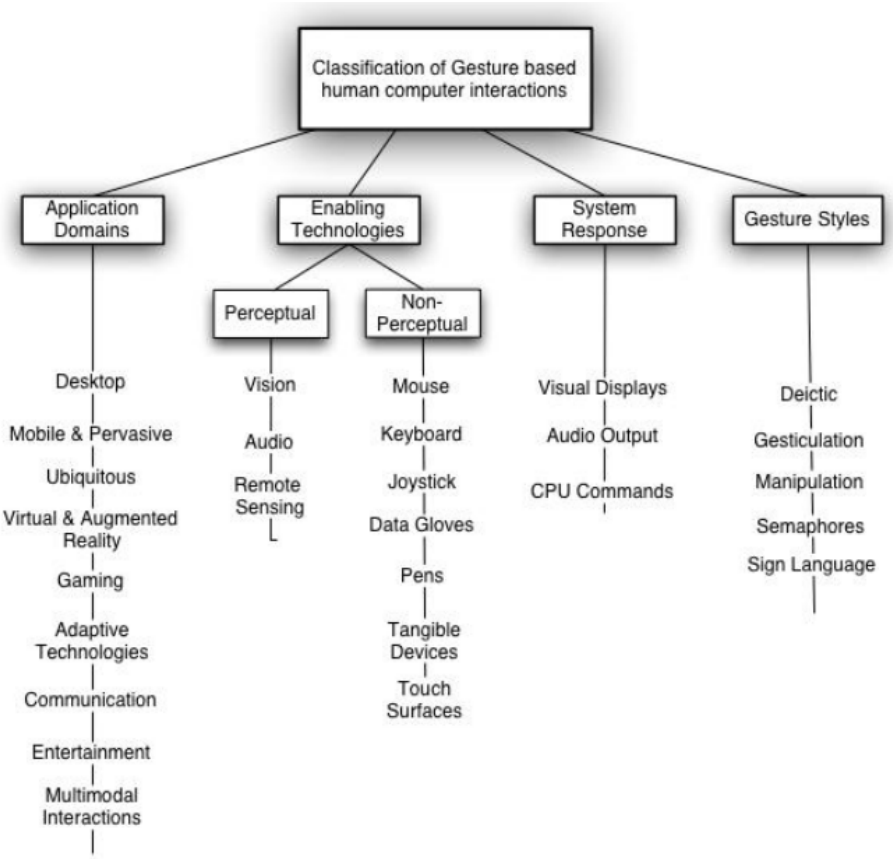


Figure 2.8: Karam's taxonomy. Extracted from [89]

- *Deictic gestures*: Deictic gestures involve pointing to single out an object or to indicate its spatial location. They can be used to obtain something, redirect attention to an object to make a request. Deictic gestures can be performed directly on touch surfaces or from a distance using wearables, cameras.
- *Manipulating gestures*: The aim of manipulating gestures is to replicate the movement of hand/arm gestures on the physical or virtual object being manipulated. There is a tight relation between the number of dimensions of an input device with the type of manipulating gestures it enables.
- *Semaphoric Gestures*: Gestures which can trigger actions defined in a predetermined dictionary constitute semaphoric gestures. For example, mouse or stylus gestures which result in strokes or marks can be mapped to commands on a smart device.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

- *Gesticulation*: Gesticulations do not rely on are not based on pre-recorded gesture mappings as in the case of semaphoric gestures. They usually accompany speech and complements the meaning the person trying to transmit. Howevern they can be used as free-form gestures that are open to interpretation. Wexelblat [180] refers to gesticulations as idiosyncratic, not taught, empty handed gestures. Highly abstractive sensors are very good at contextualizing gesticulations.

- *Language gestures:* They require a series of signs that form grammatical structures. Conversational interfaces work with language gestures. For them to be useful, first semantics should be extracted from the conversation. Consequently they are often fused at the decision level.

In addition to the gesture classification, Karam classifies the enabling technologies for gestural interaction:

- *Non-perceptual input*: Involves the use of devices or objects that are used to input the gesture, and that requires physical contact to transmit location, spatial or temporal information. These technologies include mouse and pen input, touch and pressure input, gloves, sensor-embedded objects and tangible interfaces but also audio input and tracking devices that employ transmitters placed on users.

- *Perceptual input*: They do not require physical contact with an input device. They are not intrusive, allow distant gestures. These technologies include computer vision and remote sensors.

However, we believe the proposed enabling technology classification is too limited to understand the relationship between gestures and devices. The evolution of input devices is parallel to the different gesture classes. Deictic gestures are usually performed by pointing devices, such as the mouse, touchscreen or stylus in contact, or by optical trackers for distant pointing. Manipulating gestures require careful consideration of the mapping of dimensions between the input device and the target to manipulate. Translation and rotation of a graphical 2D object is easy with a mouse, but manipulation of 3D objects resulted in a conquest for specialized input devices and tangible interfaces. Semaphoric gestures, gesticulations and language gestures need different levels of abstraction depending on the support on which user wants to perform them. When an application responds to different types of gestures, it may need different types of input devices that are more

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

convenient to each task. In section 4 we discuss the fusion of deictic gestures with gesticulation by using a touchscreen and camera. In section 3 we talk about the combination of deictic gestures and semaphoric gestures.

2.3.3 Discussion

In this section, I reviewed many things from the classification of simplest sensors to the relationship between contextual sensors and gestures. In the rest of the document, I mainly cover the data fusion of sensors which have a higher level of abstraction. Fusion of sensors which have a lower level of abstraction is less interesting for two reasons. First, because it is either too simple; combining a button with a slider does not require too much reflexion, or it is a mathematical problem that has been studied in the models we discussed in section 2.1. Second, because there is generally no change to the user's behaviour by the combination of said sensors. Whereas the context dependency of more highly abstract sensors forces the users to discover new ways to interact with the multitude of complex devices.

Users discover the potential hidden in the environment without additional steps involving memory or inferences [65]. This is called the affordance of a system and it implies that both the actor and the objects are parts of the attributes of the interaction. Affordances are also independent of perception, they can reveal themselves to a user while being hidden from another.

Affordances are perceived and discovered by all senses; the sight is obviously the principal sense, but tactile and auditory senses are important sources of information. Thus, multi sensor systems, if they feature different sensors, they offer a variety of affordances. I believe that new affordances *emerge* from the combination of input devices. Either these affordances are discovered in the transition between the devices, or are merely the result of the modified input space.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

2.4 HETEROGENEITY IN INTERACTIVE SYSTEMS

Multimodal systems contain various data types. Ubiquitous systems contain various devices. Heterogeneous systems contain various devices and data types, but we define a heterogeneous system as a system that distributes interaction spatially on different devices, in order to combine their physical capacities such as working with different proximities, and the gestural opportunities they offer such as performing deictic gestures and gesticulations at the same time. This definition is not by any means a new description for multi-sensor systems that concurs with multimodality or ubiquitousness, but it provides a clearer characterization of our contributions to multi-sensor interaction.

Heterogeneous systems necessarily accommodate multiple gestural modes acquired from different input sources. Gestural modes include, but not limited to, *on the surface*, *above the surface*,*around the surface*,*mid-air*,*distant*,*proximal*,*facial*, *vocal*... The acquisition of multiple gestural modes are facilitated by separate input devices which gives a better representation of the gestures, requires less resources and observes the gestures from different angles. I demonstrate homogeneous and heterogeneous systems by classifying some existing work on multi-sensor interfaces.

2.4.1 HOMOGENEOUS DATA FUSION

Homogeneous multi-sensor systems are not our main interest in the scope of this thesis, but they are useful for elimination of occlusions and the improvement of decision quality. For a given gestural mode, they combine the same type of data from multiple sources in an overlapping interaction volume.

The most evident example of homogeneous data fusion is camera calibration. 3D vision enabling methods such as Zhang's flexible camera calibration [196], or various triangulation techniques that use intersecting rays [75] combine the images from two cameras to obtain depth information.

Today, depth information is obtained through novel technologies such as structured light or time-of-flight. Depth cameras capable of skeletal tracking such as Microsoft Kinect, Leap Motion, Intel RealSense and others are especially popular. Still multiple depth camera streams are fused to avoid occlusion-related problems. The simultaneous use of multiple depth cameras can widen the observed volume

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

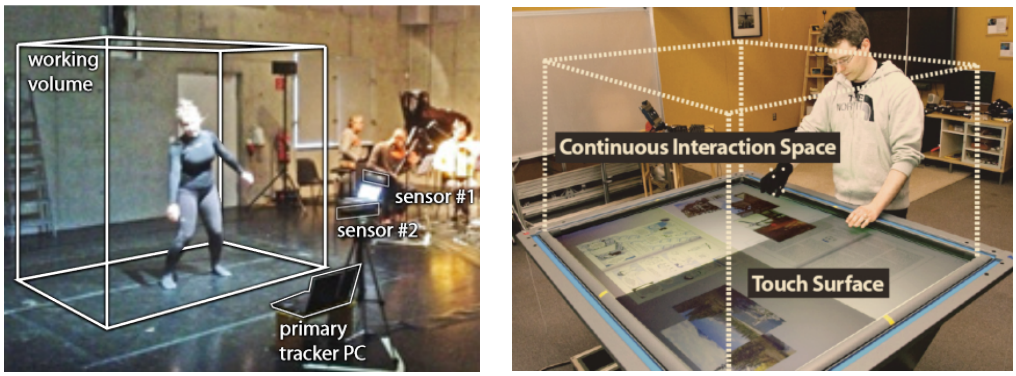


Figure 2.9: (left)Homogeneous Kinect fusion for a dance performance. Extracted From [62] (right) Heterogeneous fusion of a touch surface and the volume above. Extracted from [117]

and provide a better description of the scene. For example, dance analysis and sonification require accurate recognition of body movements. Performers are highly mobile on the scene, therefore multiple depth cameras [62] [97] are used to film the stage from different angles (Fig. 2.9) The common strategy to follow is to calibrate the cameras at a fixed position and fuse the skeletal data continuously to avoid frames with missing joint information.

Different sensors can also be used in a complementary way to increase the accuracy of the systems. Penelle and Debeir [140] use a Kinect with a Leap Motion to improve hand tracking performances. Although their implementation is stable, it needs a processing step to segment the hand and to locate the fingers in Kinect images. However, from our perspective, this work does not constitute heterogeneous fusion. Even though the input devices provide different types of data at different abstraction levels, they are only used for detecting hand poses and do not diversify the gestures the user can perform. In other words, from a material point of view they may seem heterogeneous, but from a gestural point of view, they do not qualify for heterogeneous fusion.

2.4.2 Heterogeneous data fusion

The distinctive characteristic of heterogeneous data fusion is the distribution of gestures across devices. Data fusion between devices are not necessarily continuous, the essential is that multiple devices collaborate to complete one task. While doing so, active devices can activate sequentially or in a parallel manner.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

In a central architecture, a decision making mechanism determines which sensors are actively participating by observing activation and termination patterns.

The WILD room [19] (wall-sized interaction with large datasets) is such a heterogeneous environment. It includes wall sized display composed of mutiple screens, mobile devices, a multi-touch table and miscellaneous input devices. In this setting, multiple users can interact simultaneously with various surfaces combining different input devices in different configurations. Depending on the context, portable devices in the WILD room transform into different instruments. For example, a smart-phone can be used as laser pointer and interact with objects located on other surfaces. Users can select objects on a surface with an instrument, move them with a second instrument, change their color with a third one. In order to fuse the data from the instruments, WILD Room uses a central layer called WILD Input Server which allows users to create different input configurations to aggregate inputs from different sources. For example, a multitouch device tracked by external cameras is seen as a single device that allows both touch gestures and manipulation gestures at the same time, combining two devices and two gestures mode. Hence the overall system qualifies for our heterogeneity definition.

CodeSpace [30] is another heterogeneous ubiquitous system that allows use of multiple personal mobile devices in developer meetings. It combines in-air pointing with touch gestures for fine gestures. In CodeSpace, Kinect sensors enable distant pointing techniques by calculating the ray casted by the user's arm holding the smartphone and the touchscreen of the phone enables finer movements or triggers predetermined actions. In other words, the depth sensor is responsible for ample control, while smartphone is responsible for finer control. It also solves segmentation ambiguity of in-air gestures. Additionally, smartphone's IMU allows the calculation of its orientation, which is then combined with pointing gestures to display or hide personal content on the the shared display.

Similar to CodeSpace's combination of ample control with finer control, MultiFi [67] combines multiple displays with different fidelities for input and output. It uses a head mounted display (HMD) with a smartwatch and smartphone to interact at different scales. The HMD is used to determine the position of the smartwatch while the smartwatch is used for selection. At the same time, both displays provide a visual feedback at different precision levels aligning on and around the user's body. It distinguishes three alignment modes. In device-aligned mode, the interaction and display both happens on the touchscreen of either the smartphone or the smartwatch. In body-aligned mode, the coordinate space is that of the user's body, and the touchscreen activates finer precision zones while being able to move around the user. In side-by-side mode, there is no spatial

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

relationship between devices and the user can interact on both independently. Another example to the same approach is found in [161].

These touch-sensitive surfaces restrict the user interaction to a 2D plane. Yet there is a rich interaction space above touch surfaces that is neglected. Tangibles may use the space above the surfaces, but generally they only exploit the contact surface. In order to enrich the interaction with touch surfaces, adding a third dimension to the system is an approach that have been gathering attention. DepthTouch [20] is an interactive system that enables the user interact with a 3D scene projected on a transparent vertical surface. DepthTouch shows that even if there is only one device to capture both 2D and 3D interaction, there is a fusion of interaction spaces and it should be taken into consideration for scenarios of use. A more concrete example with multiple devices is the Mockup Builder [51]. In Mockup Builder, 3D position of the fingers in the volume above the touch surface observed by two Gametracks. Then, the positions of the fingers are interpreted with the 2D points on the surface to model and manipulate 3D objects on the multitouch table with stereoscopic projector. Their work is based on Marquardt et al. [117]'s Continuous Interaction Space whose aim is to build a system that treats the space on and above the surface as a continuum. Instead of considering on and above the surface as two discrete interaction modes, they investigate the rich interaction space between them. Here, Marquardt et al. fuse the data from a Vicon capture system with the touch surface to interact in a 3D area.

The importance of the Continuous Interaction Space stems from its ability to fuse two interaction spaces. The resulting interaction space enables novel interaction techniques. While it conserves distinct gestures on the touch-surface and in the air, it also enables extended continuous gestures and proximal continuous gestures. Extended continuous gestures can begin on the touch-screen and continue above the surface in a continuous flow. Proximal continuous gestures complement the touch input with hand pose and orientation in the near space above the screen, allowing users to trigger multiple actions. Talaria [146] uses extended continuous gestures to drag-and-drop on a touch wall display that would otherwise require the user to displace too much.

Over the screen interaction is also beneficial for smartphones. Chen et al. demonstrated how in-air gestures can add expressivity to classic mobile touch-screen interaction [47]. The thumb movement before and after touch carries useful information that can be used to detect paths to perform gesture in-between touches. Touch input also helps segmentation of in-air gestures, an approach we demonstrate in the section 4.2. Extending the interaction space around a smartphone does not necessarily require an additional device. [167] uses the

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

built in camera if unmodified mobile devices to recognize static in-air gestures in real-time. They define in-air gestures that can complement touch gestures to switch modes or to modify multiple parameters at the same time using both gestures simultaneously.

Aside from using the finger above small devices, it is possible to use pen devices [12]. Pen input is already supported by modern devices, it prevents screen occlusion and it can compensate ergonomic constraints when combined with the regular multi-touch gestures. Portico [13] uses tangibles around a tablet. It combines the touch input with two cameras placed above the screen. The cameras provide a visual recognizer for the surroundings and tangible objects. This allows tangibles to used both directly on the surface, and at around the frame of the tablet.

Using the space above input devices is not limited to screens. Wacharamanotham et al. made use of the finger above desktop devices such as the keyboard and mouse [175] with the help of VICON system. They determined an appropriate thickness for near-surface interaction layers and proposed a method to dynamically place the layer above the device.

This list of heterogeneous systems is not exhaustive. There exists other multi-sensor systems that use heterogeneous input devices to enrich and extend interaction spaces. However, from the examples we have seen, it is apparent that cameras are very powerful devices to add an additional dimension to conventional interaction devices. As a result, our contributions overwhelmingly feature the external use of both RGB and depth cameras.

2.5 CHAPTER SUMMARY

In this chapter, I presented previous work related to data fusion from an HCI point of view. To put things in perspective, I started with a brief history of data fusion that originated from military applications, and arrived at a very specific case of data fusion that I defined as heterogeneous. In this regard, a discussion about the presence of data fusion in various research domains under the HCI roof, such as multimodality and ubiquitous environments, was necessary. As an intermediate step, a taxonomy of input devices which takes into account both material and gestural considerations was provided. This taxonomy helped us to

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

better understand that heterogeneous systems distribute acquisition of gestures around different input devices.

Heterogeneous fusion examples demonstrate that contextual sensors are very good at enriching and extending interaction spaces. Yet, the taxonomy indicates that contextual sensors such as cameras have a high abstraction level. It would mean that in order to extract context from cameras, some processing steps are necessary to obtain suitable descriptors. However, the larger bandwidth caused by the use of multiple devices require a lighter processing step, especially in computationally constraint scenarios. Consequently in the next chapter, I present our contribution to extracting meaningful interaction data from such sensors.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Three

From signal to meaningful interaction data

3.1 Introduction

Signal, in the broad sense, is an abstract term [42]. Oppenheim and Willsky's widely taught Signals and Systems [137] book states that the signals are functions of one or more independent variables, contain information about the behavior or nature of some phenomenon. A signal should present an observable change over time or space. Signals are captured by sensors specialized for the attributes that are observed (Sec. 2.3.1). They can be analog or digital, continuous or discrete; analog signals measure continuous properties through physical mediums such as electrical or mechanical, digital signals represent a sequence of discrete values and are often obtained by the quantification of analog signals.

The *meaning* relayed by a signal is not always apparent. It is *revealed* through a chain of treatments to retrieve information required for a specific task. The meaning can be simple; if a button sends a signal, and it has only one *meaning*. Moreover, a signal can contain multiple *meanings*; such as a video sequence of a complex environment which can mean that *there are people* and/or *the lights are on* and other things. Therefore, meaning of a signal depends on the context characterized by the interaction between users, applications, and the surrounding environment [53].

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Processing techniques are used to manipulate the pure signal, but not all of them are intended to extract the meaning. For example, even without post-processing, equalization is a processing step; a microphone receives the sound waves, and the amplitude is modified with respect to frequencies to provide a better sound. However, we cannot say that equalization only for cosmatic purposes constitutes a passage from signal to meaning; it is only filtering to achieve a more attractive sound. However, it develops into a part of the search for meaning once it becomes a preliminary step in order to increase the reliability of feature extraction.

In a multi-sensor system, the first step to data fusion is the preparation of sensors. This step involves identification of input devices, calibration and obtaining meaningful data from each source. In the context of heterogeneous sensors, it is coherent to employ either *characteristic level* fusion (cf. Sec. 2.1) where first features are extracted from individual inputs and then all features across inputs are fed into the decision making mechanism, or *symbol level* fusion where the decisions obtained from individual inputs are combined to make the final decision. Thus, the quality of the heterogeneous features extracted from the inputs influences the reliability of the decisions.

Reliability for a human-computer interface does not only depend on the accuracy of the decisions, it also necessitates speed and fluidity. This results in a trade-off between the potency of the processing and the responsiveness. Traditionally, extracting good features consumes more time than extracting mediocre features, but a lagging interface is practically unusable. Considering that one of the aims in designing sustainable interaction techniques requires the ability to reuse older equipment, we find ourselves in a computationally constrained scenario. The question that exactly how heavy should the processing step be to provide a fluid interaction does not have a straightforward answer, and the solution is found in empirical tests. Yet, it is important to explore this duality, especially for devices which operate at a higher abstraction level, such as microphones or cameras.

Processing the incoming video stream is a computer vision problem. However, when applied to human-computer interaction, the practices of computer vision community should be reappropriated. Image processing research relies on offline methods; the main focus is to increase the accuracy with moderate amount of consideration for its applications. This is by no means a criticism; in my opinion, it is similar to developing an interaction technique before the emergence of consumer grade technologies to support it. Nevertheless, the state of the art

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

methods in computer vision are very powerful and can enable usable interactive systems.

In the first section, I explore the gray area between what computer vision is able to provide and what data fusion in HCI applications require. My main contribution [9] [10] is to extract ready-to-use, computationally light features to describe motion information obtained from cameras. Throughout this section, the words *motion* and *movement* are used interchangeably, as there exists no consensus on how to use these them to signify displacement [1].

Another issue in designing sustainable systems arises from the use of reusable, detachable, modular equipment. If we want to be able to disassemble a multi-sensor system to use its pieces in other configurations, we need to be able to assemble the new configuration quickly. For the ease of use, the calibration step should be *light*. In order to ensure the seamlessness we discussed priorly, it should be robust to change of users.

In the second section, I demonstrate a seamless calibration technique for depth cameras in a physically constrained scenario.

3.2 Low-cost motion information from RGB cameras

Tracking moving objects in a video sequence has wide application areas from surveillance to robotics. It is also an important topic in HCI applications, especially to observe moving body parts such as the limbs, head, eyes and so on. Computer vision has numerous methods to track either individual objects or to represent overall movement in a video sequence. The differences between consecutive images indicate a displacement of the included pixels, and regrouping the displaced pixels can give a meaning to the movements of the viewed objects or the capturing device.

Optical flow is the distribution of apparent velocities of moving brightness patterns in an image [77]. The three factors which cause the moving patterns are the changes in the position of the camera and/or the captured objects or the variation of the lighting. As this thesis is predominantly interesting for the HCI community, it is useful to give a simple explanation of the optical flow algorithm.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Given two images I_{t_1} and I_{t_2} at two time instances, the aim is to find the correspondence of the pixel $I_{t_1}(x,y)$ with the pixel $I_{t_2}(x+u,y+v)$. If we assume that the brightness of the pixel remains constant at the two instances:

$$I_{t_1}(x,y) = I_{t_2}(x+u,y+v) \tag{3.1}$$

The second assumption is that the displacements are relatively small between two images (less than one pixel), so, omitting the higher order terms in the taylor series expansion of I_{t_2}:

$$I_{t_2}(x+u,y+v) = I_{t_2}(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v \tag{3.2}$$

Combining the two equations, we have

$$I_{t_2}(x,y) - I_{t_1}(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v = 0 \tag{3.3}$$

as v and u tend to zero, it can concisely be expressed as

$$\frac{\partial I}{\partial t}I + \nabla I \cdot [u\,v] = 0 \tag{3.4}$$

At this point, we have one equation but two unknowns. This is known as the *aperture problem*; it means only the displacement component along the gradient direction is known. In order to get more equations for a pixel, Lucas-Kanade [109] assumes the pixel's neighbors have the same displacement vector, thus using a 5x5 patch, we can obtain 25 equations which can be solved by a minimum least squares solution. There are other solutions to this problem [77] [127] but Lucas-Kanade's method constructs the backbone of the method we later use in our applications.

One of the main problems in this approach is the assumption that the movements are very small. To resolve this problem, Lucas-Kanade algorithm iteratively minimizes the sum of squared errors between I_{t_1} and I_{t_2} by first estimating the velocity at each pixel by solving the above equations, then warping I_{t_2} towards I_{t_1}

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

using the estimations and repeating the estimation and warping until convergence occurs. Warping step is computationally expensive but [15] proposes an inverse method to speed up this process.

For even larger displacements, coarse-to-fine schemes are used to speed up the convergence [7]. Image pyramids consisting of multiple downsampled images are used to estimate the large displacements in the coarser levels and those estimations help warping the image in finer levels.

Optical flow methods can be surveyed according to different research directions [44], for the scope of this thesis, it is beneficial to differentiate *sparse* and *dense* optical flow.

In sparse optical flow techniques, some important feature points such as corners and edges are extracted first and matched between different images. This way descriptor couples between successive images have a displacement vector to characterize the motion for a couple of points. The computational weight of sparse methods stems from the calculation of the descriptors; they require preprocessing of the image to eliminate uninteresting parts of the scene. However, if the content of the image is unknown, pre-filtering becomes another challenge to attack.

Dense optical flow techniques calculate a displacement vector for each pixel in the image. Dense displacement vector fields are more accurate than the sparse ones, but calculating a vector per pixel is generally a computationally time consuming task, even when this problem is alleviated by multi-scale solutions and estimations [59]. Nevertheless they are very useful to solve abstract problems such as action recognition and expression detection [14]. A fast dense optical flow method would be an excellent candidate for HCI applications.

In 2016, Kroeger et al. [101] introduced DIS-Flow, an optical flow estimation method that uses the inverse correspondence method and coarse-to-fine image pyramids we explained in this section. DIS-Flow is one order of magnitude faster than others and produces roughly similar flow quality. Their objective is to trade-off a less accurate flow estimation for large decreases in run-time for time critical tasks. This method provides a dense flow output, in which every pixel is associated to a displacement vector. DIS-Flow was used in the works we present in this section and it also enables the techniques in section 4.2.1.

Visualizing the dense optical flow field of a given gesture would be helpful to understand its usefulness for gesture recognition. The motion at a point can

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

be represented with Δx and Δy, the horizontal and the vertical displacement respectively. Therefore displacement vector can be expressed as $v(\Delta x, \Delta y)$, but also in polar coordinates $v(r, \Theta)$ to represent the amplitude and the direction of the movement for a pixel. Fig. 3.1 shows a color coded representation of the optical flow.

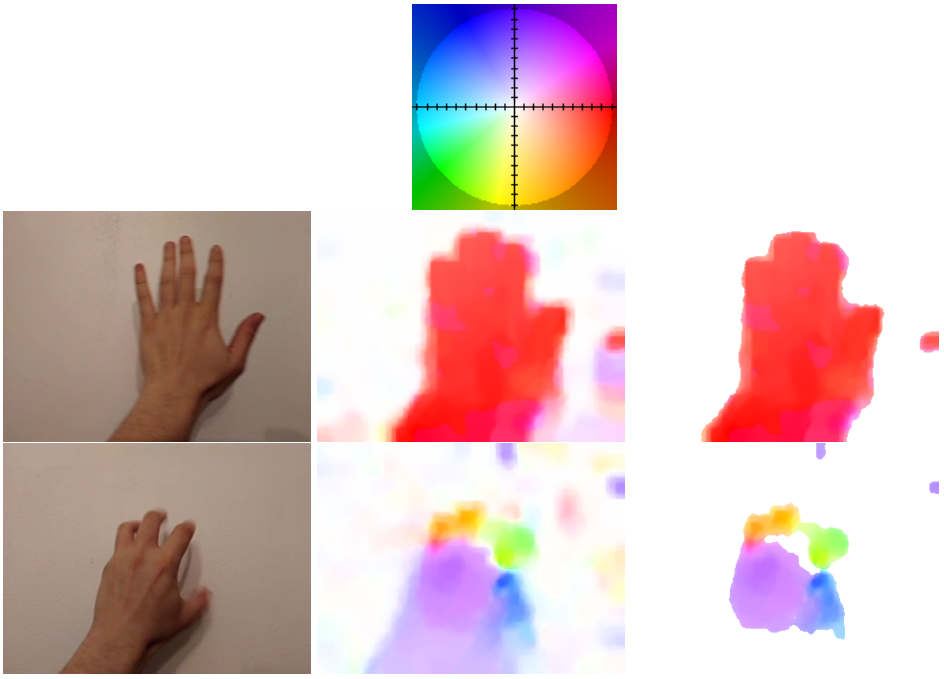


Figure 3.1: Color representation of the optical flow. (Top) Color space: hue indicates direction, saturation indicates amplitude. (Middle) Swipe right gesture, calculated flow, filtered flow. (Bottom) Grasp gesture, estimated flow, filtered flow

In fig. 3.1, the top image shows the color space used to represent the displacement vector for each pixel. Saturation on the wheel indicates the amplitude of the displacement; a pixel which does not move is white colored. As the amplitude increases, the color becomes saturated. The directional component of the displacement is represented by the hue on the wheel. For example, a pixel which moves to the right is red and a pixel that moves to the bottom is yellow.

3.2.1 REPRESENTING CONTINUOUS GESTURES

In section 2.3.2, Karam's taxonomy on gesture styles defined gesticulations as empty handed gestures which do not follow a preexisting definition. Gesticulations usually accompany speech, but are also valuable for creating free flowing

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

movements for artistic expression. A dancer's movements consist of gestures; while a single gesture may not be meaningful in itself, a sequence of continuous gestures tell a story on stage.

Optical flow is a handy tool to represent continuous gestures. Especially when the aim is to obtain simple movement information to be combined with other modalities, it proves to be an effective way of conveying a description of a camera stream. Nevertheless, in order to achieve a good description of the scene, we propose a processing chain.

Optical flow computation is noisy and presents important motion discontinuities as it measures the displacement in the 2D image plane of the real 3D world. Filtering out noise and focusing on the coherent motion information is the key to a successful representation.

Our approach to extracting features from raw flow data is as follows. First of all, for every region of interest, the pixels that have a smaller displacement value than a threshold are discarded to eliminate the noise. In Fig. 3.1, the middle column shows the unprocessed output of the DIS-Flow. Filtering by the magnitude (discarding the unsaturated pixels), we obtain the rightmost column, where the shape of the blob corresponds better to the moving object, in this case the user's hand.

We identified key information to gather from the optical flow output. We collect three global values: the amount of moving pixels, the average direction and the average magnitude of the displacement vectors. The amount of moving pixels corresponds to the ratio of the number of remaining pixels to the image area. The average direction and magnitude are the direction and amplitude of average displacement of the moving pixels.

Other than the global values, it is advantageous to represent the distribution of the movement. To do so, we construct a normalized 8-bin histogram of directions 3.2. By quantizing displacement angles into 8 levels, this histogram shows the amount of pixels moving along 8 different axes. For example, the grasp gesture shown in 3.1 has four main direction components; the hand moves to top-right (purple - direction 2), the thumb moves to top left (blue - direction 4), index and middle fingers move to bottom left (green - direction 6) and ring and pinkie fingers move to bottom (amber - direction 7). Both from the color representation and from the histogram, we can notice that most of the pixels move along the second direction. It is, then easy to distinguish between a swipe right gesture where only the first bin of the histogram would be full, and the grasp gesture.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

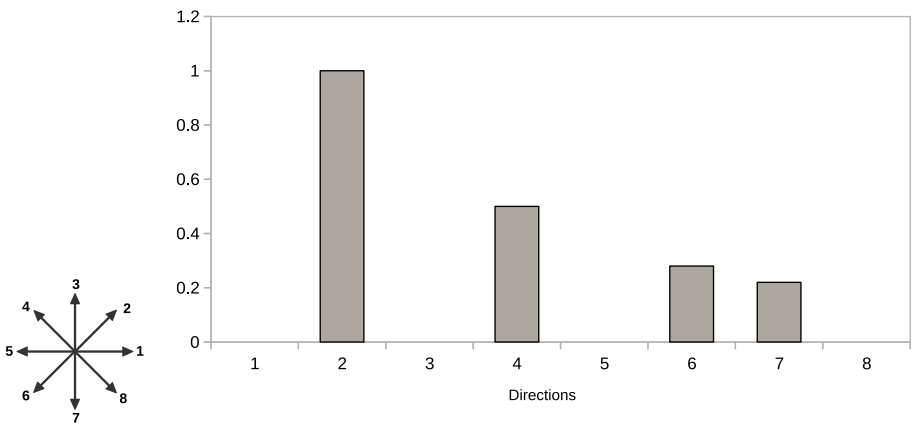


Figure 3.2: (left) 8 directions (right) Histogram of the grasp gesture

Moreover, different functionalities can be assigned to each direction, an approach that I demonstrate in the chapter 4.

Adding the three global values to the histogram information, a feature vector of eleven elements is obtained. This vector is then ready to be combined with the features obtained from other input devices or sensors. For a decision level fusion, we should explore discrete gestures.

3.2.2 REPRESENTING DISCRETE GESTURES

According to Karam's taxonomy (section 2.3.2), semaphoric gestures employ dynamic limb gestures which have a definition in a vocabulary commands. Thus, differing from the previous section, we need to isolate individual gestures performed in a time interval and distinguish one from another.

Video analysis is an important aspect of multimedia data description. A basic task in video analysis is the extraction of optical flow, which helps understanding individual region motion in the video stream. Optical flow analysis enable several applications in gesture based interaction [183]. Discrete human gestures can be inferred from video analysis using optical flow created by the dynamic parts of the body. Gestures can be interpreted at several levels. e.g. full body gestures, facial gestures, or hand gestures. Discrete gestures can trigger actions, such as a left click on the mouse to select an item, and can be combined with other inputs to alternate the action, such as the combination of the control key with the left click

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

to select multiple items. Thus, the aim of this section is to generate action-ready video descriptors for data fusion.

In order to demonstrate the representation of discrete gestures from a video sequence, we address a specific scenario of one-finger gesture in the context of mobile interaction. In this scenario, a user interacts with a mobile phone using the index finger on phone's back camera equipped with a wide angle lens 3.3.

Using the back of devices has been proved to be efficient [184] to resolve the issues [18] such as the fat finger problem, target ambiguity and occlusion that emerges when the user interacts with the touchscreen. Phone cameras have been investigated for back of device interactions. Xiao et al. [188] proposed performing index finger gestures directly on the camera lens. They distinguished partial and full occlusion of the lens and the dynamic swiping with the index finger. A 3D-printed ring [190] was mounted on the camera to turn it to a joystick. Wong et al. [185] widened the interaction space on the back surface with a mirror attached to the camera.

Similarly to above examples, adding a wide angle lens to increase the motor space for the index finger and offer more accurate interaction capabilities. The ability to accurately capture index finger gestures opens interaction opportunities for daily tasks in mobile contexts, such as controlling widgets, sending messages, playing video games and so on. However, this innovative use of the back camera rises a number of issues, such as the non-linear field of view due to the fish-eye lens, and the constant background motion.

Working for such a specific task required the creation of a new dataset. We created a new index finger dataset including 6 gestures together with a baseline classification algorithm dedicated to this dataset. The 6 dynamic gesture classes are swipe left, swipe right, swipe up, swipe down, tap on the lens and a counter-clockwise circle (fig. 3.4). We chose tap and swipe gestures because they are the most commonly used gestures for mobile interaction [170]. A circular gesture is added as the shape of the lens provides a guidance during its execution. Such a dataset is useful to help researchers compare their approaches on the dynamic aspect of the proposed gestures but it can also serve as a base for static fingertip detection. To the best of our knowledge, this is the first index-finger dataset released for the multimedia and interaction communities.

The rest of this section is organized as follows: we first discuss the related work in hand gesture recognition and the available datasets to demonstrate the need for a index finger gesture dataset. Then we introduce our dataset containing 6

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

dynamic gesture classes and the gesture recognition baseline. Finally, we discuss the results of our experiments.

EXISTING DATASETS

Hands are the main body parts used in vision based gestural interaction [187] [147] [143]. We use our hands for non-verbal communication and we are able to perform various gestures ranging from simple hand poses to complex movements. The richness of the shape and motion content in video streams featuring hand gestures has been utilized for image and video recognition methods. However, using only one-finger gestures in front of a camera has not yet been explored for computer vision applications. An overview of hand gesture datasets and gesture recognition methods can be found in [143]. From the overview, one can deduct that the general approach to hand gesture recognition is to extract texture and/or motion features from either an RGB or RGB-D sensor and to classify them in order to obtain class labels for the gestures.

Several hand gesture databases were proposed for testing of hand gesture recognition algorithms. RGB-D datasets [72] [52] are not usable in a mobile context as the phone cameras do not provide depth information. In RGB datasets, a hand gesture can refer several different situations.

1. A static pose [144] [68]: Static images of the hand can be used as an interaction input as the state of multiple fingers allows differentiating various postures, such as the sign language. But the index finger behind the mobile device doesn't allow performing more than a few finger poses.

2. A dynamic gesture corresponding to the whole hand's position relative to the user or to the environment [106] [88]: In these datasets, the performed gestures are very large while the index gestures we propose are of smaller nature.

3. The change of the hand posture by moving the hand and the fingers locally [114] [95]: In these datasets, the hand is the only moving object in the image, but movement of multiple fingers at the same time create more complex gestures than it is possible for the index finger. Moreover, both the use of a fish-eye lens and the back-of-device camera bring some deformation and background motion to the input stream that are specific to this context.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

A variety of approaches were employed to represent the hand movements in the above datasets. Hidden Markov Model based methods [105] are often used on dynamic hand gestures to spot the start and the end of the gestures. Mercel et al. [115] applied an input-output HMM on hand silhouettes. Moreover, Shen et al. [164] used a local descriptor to capture local motion patterns. In the original paper that created the Cambridge Hand Gesture Dataset [95], tensor canonical correlation analysis features are combined with nearest neighbor (NN) classification.

None of the datasets above contain images that focus on individual finger movements that can be useful for back-of-device interaction. We need a set of images that involves index finger, hand tremor (unintentional shaking) and the spherical transformation induced by wide angle cameras to test simple descriptors that can be used in the mobile interaction context. Even thought the processing power of mobile devices increase continuously, methods with a low computational complexity are needed for a fluid interaction.

DATASET

We created an index finger gesture dataset consisting of 746 video sequences of 6 index finger gestures: left, right, up, down, tap, circle (as shown in Fig. 3.4). The gestures are performed several times by 14 users (2 left handed), and captured with a low-cost wide angle (235°) lens (see Fig. 3.3). Each video shows a user performing gestures freely without visual feedback, in order to ensure the naturalness and authenticity of the finger movements. Sequences were taken over different backgrounds with an uncontrolled auto frame rate that varies between 20 and 28 fps depending on the lighting conditions.

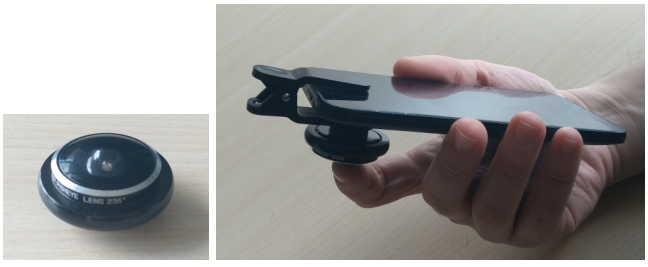


Figure 3.3: Left: Wide-angle lens. Right: Interacting with the lens.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

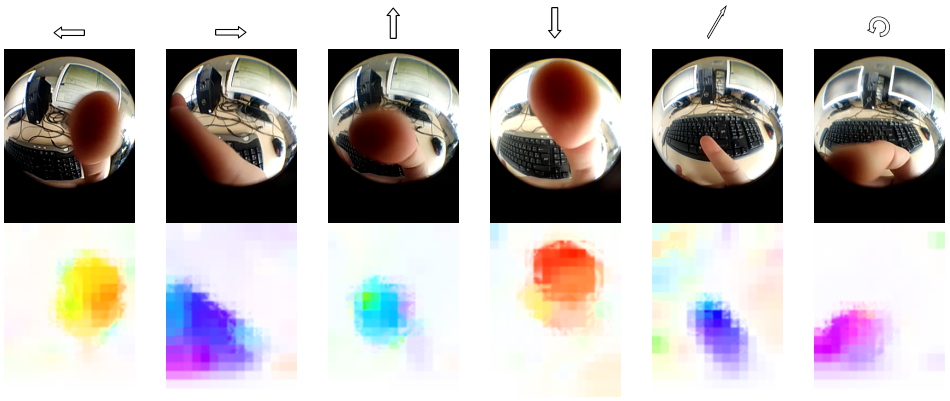


Figure 3.4: 6 gestures in the dataset. (Top row) Example frames from the start of each gesture. (Bottom row) Color representation of the optical flow.

The dataset is representative of main issues that should be taken into account when the user interacts with the device. First of all, the semi-spherical lens does not map seamlessly to the rectangular sensor. This means there is an active circular region where the information is concentrated in the 320x240 image. The wide angle lens introduces shape deformation that is challenging for appearance based methods. Moreover, some cases of lateral illumination result in small lens glare in the videos.

Another challenge is the user's way of holding and interacting with the phone. Some large-handed users are able to hold the phone from the lower part to perform clear index gestures. Users with smaller hands hold the phone closer to the lens, and in that case other fingers may be visible in the image and introduce unwanted movements. Hand size also affects the shape and the size of the fingertip in the image.

Due to the fact that the index finger is attached to the hand holding the phone, moving the finger causes small changes in the orientation of the device (hand tremor). The changes in the orientation of the camera adds a global motion to the image sequence at the time of the gesture. The introduced motion is not uniform because of the shape of the wide-angle lens.

Finally, even though the swipe gestures are labeled as left, right, up and down, the gestures are almost diagonal because of the orientation of the index finger behind the device. As a result, left and down gestures are close to each other, so are up and right gestures. Such an effect is also observed in [188].

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Hence, with regard to the existing state of the art datasets, the newly proposed dataset brings new interactions opportunities as well as new challenges to deal with such as the ability to deal with index finger gesture in mobile context using back camera, the variations in terms of index finger dynamic, unconstrained usage of the mobile dynamic and challenging backgrounds.

BASELINE METHOD

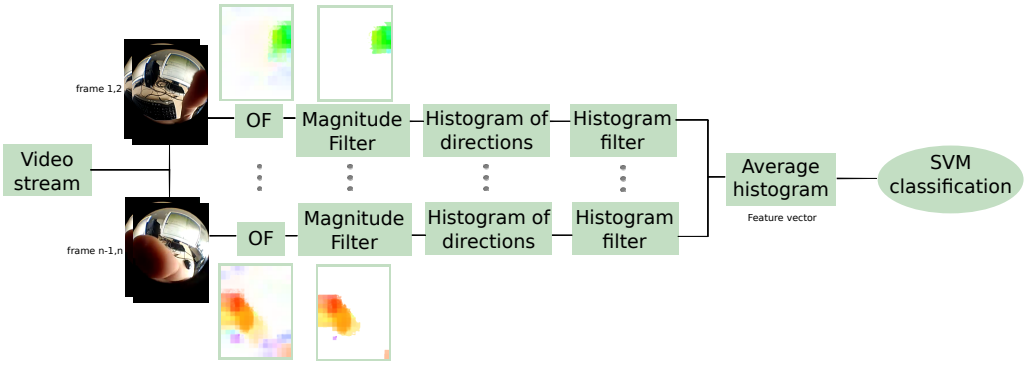


Figure 3.5: Pipeline of our method

In this section, we present our method to serve as a baseline for future works using the proposed datase. The overall scheme of our method is described in Fig. 3.5.

For a given gesture video sequence, a dense optical flow is calculated in every frame. As mentioned previously, DIS-Flow by Kroeger [102], is a less accurate method but, it has a very low complexity to match the computationally constrained scenario of using a smartphone. For the same reason, we keep the rest of our pipeline simplistic.

Unlike in fixed camera setups, use of a mobile camera introduces small movements due to hand tremor. Because of the form of the wide-angle lens, a given movement does not appear with the same magnitude and direction throughout the image. However, the magnitude of the optical flow due to the tremor is smaller than the flow induced by the finger gesture. Thus, in order to filter out the tremor, we discard the flow vectors with a magnitude less than a threshold t_{mag}. Filtering should be applied considering the position of the vector in the image as unintentional flow has higher magnitudes at the center than on the borders. If the ratio of remaining vectors to image size is smaller than a threshold $t_{vectors}$, the frame (f_t) is discarded as it is not sufficiently descriptive.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

$ValidFrames_G$ refers to the number of valid frames in the sequence G corresponding to a gesture:

$$ValidFrames_G = |\{f_t, t \in [0, N_f - 1], \frac{NVP_{f_t}}{H \times W} > t_{vectors}\}| \tag{3.5}$$

In Eq.(3.5), NVP_{f_t} is the number of valid pixels, as obtained with:

$$NVP_{f_t} = |\{m \in OF_t | m > t_{mag}\}| \tag{3.6}$$

where OF_t is the matrix of optical flow for frame t and m is the magnitude of displacement for a pixel.

As for the continuous gestures, an 8-bin histogram of directions is constructed to represent distribution of the optical flow in each frame. The tip of the index finger moves rigidly, creating most of the optical flow along the same direction. This behavior is expected to generate dominant directions in the flow as the induced motion share common characteristics along the finger. So, if a direction in the histogram is represented by less than a percentage ($t_{direction}$) of the non-zero pixels, that bin is cleared. This step was previously omitted for the continuous gestures because small changes create a richer free flowing output, but discrete gestures need a stricter frame for gesture recognition.

$$h_T[i] = \begin{cases} h[i] & \text{if } \frac{h[i]}{\sum_{i=1}^{8} h[i]} > t_{direction} \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

Hence, at each time instant, we have a filtered histogram of the optical flow. To represent the dynamic gesture, we construct a feature vector corresponding to the image sequence. As the gestures can start and end anywhere on the lens (especially the circle gesture), and the number of frames is highly variable, we proposed using the average histogram of the sequence as the feature vector.

EXPERIMENTS

For the feature extraction parameters we obtained the best results by choosing $t_{magnitude} = 5$, $t_{vectors} = 0.01$ and $t_{directions} = 0.05$. The average 8-bin histogram vectors is then fed to an SVM with RBF kernel. The average accuracy was obtained by using leave-one-user-out approach. A grid search was used to find

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

the optimal RBF kernel parameters C and γ that give the best average accuracy. In order to demonstrate the need for magnitude filtering, we compared the performance of our proposed method with unfiltered averaged descriptors of histogram of optical flow [172] and of histogram of oriented optical flow [45]. From the results shown in Table 3.1, it is clear that the noise introduced by the motion of the mobile device decreases the performance, and that filtering based on index finger motion coherency improves highly the results.

Table 3.1: Average accuracies on our dataset

	Accuracy
HOOF [45]	%63.8889
HOF [172]	%61
Ours	%73.1

Figure 3.6 shows the confusion matrix of the gestures for our method. The tap gesture appears to be the most challenging gesture to recognize because of the amount of rightward and downward motion in the video stream. This may be explained by the fact that the tap gesture motion involves at some point, vertical or horizontal motions similar to swipe gestures. Circular gesture is also difficult to recognize because the distribution of directions in the video varies according to the completion of the circle.

Table 3.2: Accuracy in the Cambridge Hand Gesture Dataset

	Original (9 classes)	Regrouped (4 classes)
Ours	%70.48	%98.48
TCCA [95]	%82	-

We also tested our method on the Cambridge Hand Gesture Dataset [95] that consists of 9 dynamic hand gestures generated by 3 different poses and motions (Flat/Leftward, Flat/Rightward, Flat/Contract, Spread/Leftward, Spread/Leftward, Spread/Contract, V-Shape/Leftward, V-Shape/Rightward, V-Shape/Contract). The shape of the hand (Flat, Spread, V-Shape) has an effect on the classification which is not the case for index finger gestures. To approximate the gestures to ours, we can regroup the gestures in 4 classes based on their common dynamics: *Left* (Flat/Leftward, Spread/Leftward, V-Shape/Leftward), *Right* (Flat/Rightward, Spread/Rightward, V-Shape/Rightward), *Down* (Flat/Contract) and *Contract* (Spread/Contract, V-Shape/Contract). Table 3.2 shows the average accuracy of our method in the original and regrouped Cambridge Hand Gesture Dataset.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

	left	right	up	down	tap	circular
left	0.75	0.01	0	0.02	0.05	0.17
right	0.01	0.83	0	0	0.16	0.01
up	0	0.03	0.80	0.02	0.09	0.06
down	0	0	0.12	0.73	0.14	0.11
tap	0.03	0.19	0.12	0.06	0.56	0.05
circular	0.07	0	0.10	0.04	0.12	0.67

Figure 3.6: Confusion matrix for our dataset

3.2.3 AFFORDANCES

Being able to represent motion information quickly yet reliably, opens up considerably interesting opportunities. Most importantly, it allows the use of existing -or even older- devices to provide a general description of the moving parts of a video sequence.

For continuous gestures, especially gesticulations, the simple description provided for every frame by our approach enables creative artistic performances. Wearable cameras with a relatively weak processing units (e.g. a raspberry pi) can be placed on a performer, and be combined with other body sensors. Modern smartphones are also ideal candidates for developing multi-sensor applications that incorporate their camera along with other on-board sensors such as global positioning, IMUs, the touchscreen and so on. The affordances of an example system is explained further in the section 4.2.

For discrete gestures, again smartphones are a promising platform to develop interesting multi-sensor applications. Our video dataset which consists of 6 commonly used index finger gestures for mobile interaction can be extended to include a variety of gestures.

3.3 SEAMLESS CALIBRATION FOR DEPTH CAMERAS

Depth-sensors are increasingly used in 3D interaction and VR. They are enabling technologies for contactless, non-instrumented interaction. They are also used for public interaction systems. They may be used to retrieve user movements, so that either virtual representation of human may be reconstructed, or movements

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

may be interpreted to generate commands. Most of these depth sensors are usually based on camera technologies, which are known to have intrinsic technological limits (for instance, vision-based systems are poorly resistant to visual occlusions). Each available sensor has a captation range, visual resolution and acquisition frequency that make it specific. For practical interactive systems, often a trade-off between sensor range, time and space resolution has to be made. We think that a good way to set up a system including a high quality tracking system at reasonable cost, is to follow a data fusion approach, i.e. combine several cameras together so that the whole interaction setup has a stronger captation than what each camera can provide. Such setups classically need a specific calibration phase that has to be achieved each time sensor positions are changed.

Due to their relatively low price and smal structure, Depth cameras capable of skeletal tracking can be used in various environments unlike high-quality Mocap systems. Very common and popular depth sensors systems include *Wide Range Sensors* (WRS in the remainder of this document) such as Microsoft Kinect, and *Short Range Sensors* (SRS in the remainder of this article) such as Leap Motion. Both classes of sensors are used in numerous applications. However different sensors have different limits of usage. With a working distance of 1,5 to 5 meters, Kinect is more suitable for distant interaction while Leap Motion is a better choice for proximal interaction with a working distance of around 25 centimeters.

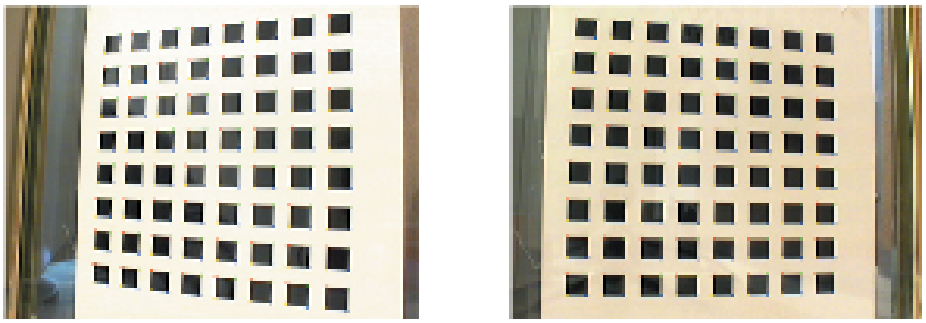


Figure 3.7: Zhang's checkerboard images [196]

Some methods exist in computer vision for calibrating acquisition of multiple view scenes. Data fusion has been used for a long time in computer vision to construct 3D images (e.g. to help a robot combine multiple sensors to achieve a certain goal). In the case of stereo vision, a famous approach of camera calibration is Zhang's flexible technique [196], where a checkerboard is observed at the same time by two cameras (fig. 3.7). Similarly, data fusion approach can be applied in the domain of HCI to take advantage of the sensors whose interaction spaces differ in terms of temporal and spatial resolution and interaction distance.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Obtaining 3D positions from two sensors means that we have two clouds of points in two different coordinate systems. It is possible to treat those points in their own coordinates, but a calibration step is needed to analyze the relations between two points that are not in the same cloud. The conventional approach would be placing the sensors in fixed positions and orientations and obtaining the absolute positions of the observed points in the scene. However these kinds of systems are harder to transport and small changes in sensors' positions decrease the accuracy of tracking. Plus, being able to change the position of the sensors gives the user more freedom.

Our method is based on observing the same line segment in two sensor spaces. If the fields of view of two sensors are overlapped, the intersecting volume is rich in terms of extrinsic parameters. Assuming that it is possible to detect the 3D positions of two fixed points on an observed object situated in the intersecting region by both sensors, we can transfer all other points between sensor spaces given that one common dimension of the sensors stays fixed. To put the idea in simpler terms, using the relative positions of points to a common line segment, they can be transferred between spaces. This stems from the reduction of the dimensions. If both sensors were free on three axes, we would have needed three common points to triangulate.

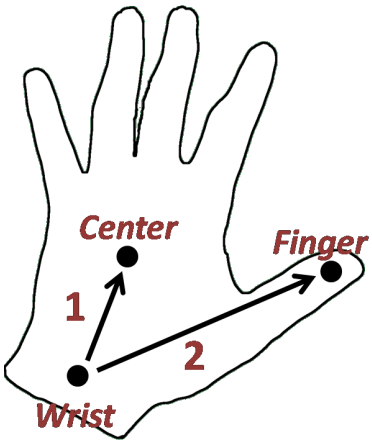


Figure 3.8: The vectors on the hand.

In the interest of materializing the proposition, we employ the method with a SRS and a WRS, both capable of tracking user's hand and wrist joints. In addition to those joints, SRS is capable of tracking the fingers. The objective of using these sensors in this case, is to transfer the finger tips to the coordinates of the WRS. The hand and the wrist form a vector which has 6 DoF. By observing it in

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

the two coordinate systems, we can add the fingers to the hand obtained by the WRS.

Figure 3.8 shows the two vectors on the hand. The vector marked by the number 1 is $\overrightarrow{v}_{hand}$, which is retrievable by both devices. We refer its two instances as $\overrightarrow{v}_{WRS}$ and $\overrightarrow{v}_{SRS}$. When the respective orientations of devices are fixed, their normalized vectors are tied such that

$$\hat{\mathbf{v}}_{WRS} = R.\hat{\mathbf{v}}_{SRS} \tag{3.8}$$

The rotation matrix R is obtained by a version of Rodrigues' equation [149]:

$$R = I + [v_p]_\times + [v_p]_\times^2 \frac{1 - \hat{\mathbf{v}}_{SRS}.\hat{\mathbf{v}}_{WRS}}{(||\overrightarrow{v}_{SRS} \times \overrightarrow{v}_{WRS}||)^2} \tag{3.9}$$

where $[v_p]_\times$ is the antisymmetric matrix of

$$\overrightarrow{v}_p = \overrightarrow{v}_{SRS} \times \overrightarrow{v}_{WRS} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \tag{3.10}$$

With the rotation and scale calculated, it is trivial to move the finger vectors of SRS on the WRS hand center. We define a finger as a vector ($vectiv_{SRSfinger}$) from the hand center to the fingertip with a magnitude of L. In figure 3.8 it is marked with the number 2. The corresponding vector in WRS's coordinates becomes

$$\overrightarrow{v}_{WRSfinger} = L.R.\hat{\mathbf{v}}_{SRSfinger} \tag{3.11}$$

Finally, the position of the fingertip is $P_{WRSfinger} = P_{WRSwrist} + \overrightarrow{v}_{WRSfinger}$. This procedure can be repeated for all or some of the fingers depending on the application.

The method is not limited to the hands and the fingers. Using a tree as analogy, if its trunk is detected by a WRS and a SRS, its branches can be transferred from the SRS coordinates to WRS.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

CONTINUOUS CALIBRATION

If the number of points to transfer is reasonably low, this method is transparent from an interaction point of view. The time difference between two consecutive frames is large enough to perform the entire calibration step continuously. This gives the user to change the sensor placement during the interaction. Wide range sensors have a large field of view is usually sufficient even when the user moves, but she can benefit from moving the short range sensors due to their limited FOV. The only restriction is that the interaction space of SRS to intersect with that of the WRS.

Figure 3.9: SRS can be moved to change target

Figure 3.9 shows a possible scenario for this use. The user can choose the desired object by moving the SRS in front it and start interacting with it immediately.

3.3.1 AFFORDANCES

In standard interaction systems, calibration is achieved as a specific sub-task, usually performed before interaction, or each time sensor relative position is modified. As previously mentioned, the proposed calibration method does not require any specific user sub-task, and can be applied on the fly. This opens the way to interaction technique in which user may move sensors during interaction session, and this can be part of the interaction technique or concepts themselves. The proposed calibration method serves as an enabling technology for rapid prototyping of gestural interaction systems based on sensor fusion.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

In section 4.4, I demonstrate the affordances of the combination of an SRS and an WRS using a design space. We see how it provides a platform for not only for distant and proximal interaction, but also supports interspace interaction and extended gestures.

3.4 Conclusion

In this chapter, we studied the sensor preparation step for data fusion. This step consists of sculpting the raw data; input streams are regrouped, unnecessary parts are discarded, important features are obtained, all to give the data a meaning.

Even though we concentrated on cameras in this chapter, passing from the signal to meaning is necessary for all sensors. In the most simplistic form, if the input provides a sinusoidal wave, extracting its amplitude and frequency describes the nature of the sensed property. Yet, as the abstraction level increases such as in the case of a camera, obtaining powerful descriptors becomes a more interesting and challenging task.

In the first part of this section, we have provided a modest way to describe continuous movements in the camera space. Then we applied a similar approach to discrete gestures in the context of a smartphone camera equipped with a wide-angle lens. The dataset we created carries distinctive properties such as a non linear field of view and hand tremor that are ready to be exploited. We described a lightweight recognition method using optical flow features to provide a baseline that achieves 73% overall accuracy.

Now, we can explore the richer interaction spaces enabled by the combination of the action-ready sensors. In the following chapter, I examine the interaction techniques that can be exploited by the fusion of heterogenous sensors that share a common interaction volume.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Four

Enriching the interaction space with data fusion

4.1 Introduction

In the previous chapter I presented the first step to heterogeneous data fusion that is extracting meaningful interaction data from individual sensors. I focused on cameras because they are contextual sensors that are capable of providing interesting descriptions of the environment. More specifically, I showed that motion information can be extracted inexpensively using a fast optical flow. In this chapter, I first demonstrate how these cheap motion descriptors can be used in complement with a mobile touchscreen, resulting in simultaneous gestures on both sides of the mobile device. The design space we proposed illustrates how such combination of mobile camera and mobile touch-screen creates a richer interaction space in the context of a novel mobile musical instrument.

Also in the previous chapter we have seen a lightweight calibration method for combining two depth cameras which operate at different distances with different precision levels. Later in this chapter, first, I show how a depth camera which has a wide view of the environment can add visual feedback to existing interaction devices through virtual objects. Then I discuss the combination of the large range depth camera with a short range depth camera in the context of a richer and continuous interaction space.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

4.2 FILTERING IMAGE STREAMS WITH A TOUCHSCREEN

Mobile cameras can provide rich contextual input as they can capture a wide variety of user gestures or environment dynamics. However, this raw camera input only provides continuous parameters and requires expensive computation. We propose combining motion/gesture input with the touch input, in order to filter movement information both temporally and spatially, thus increasing expressiveness while reducing computation time. Spatial filtering allows decreasing the number of pixels to process by selecting a region of interest. Temporal filtering serves to identify the start and end of actions. We present a design space which demonstrates the diversity of interactions that our technique enables. We also report the results of a user study in which we observe how musicians adapt themselves to the interaction space given the mobile instrument.

Many sensing capabilities of mobile devices have been explored for musical interaction [57]. Most of the mobile devices today includes various sensors such as a microphone, motion sensors, a touch screen, and multiple cameras. Hence, numerous musical interaction devices were designed around smartphones and tablets [66] [8] [178], in particular using the discrete and continuous input capabilities of the touchscreen. Among these sensors however, the use of built-in camera of mobile devices has been little explored. The simplest approach is to use the camera as a tone-hole. Ananya et al. [8] use the average gray-scale value of the camera input image to detect if the lens is covered. Covering or uncovering of the lens modifies the pitch of the sound similarly to a tone-hole. Keefe and Essl [92] used low-level visual features of the input image such as the edginess to create mobile music performances with visual contributions. Camera image has also been used to track visual reference points to gather position and motion information that are mapped to MIDI controls [152] [151].

Optical flow representation of movements has been used in various musical interfaces in the past. ParticleTecture [81] uses Horn-Schunck's optical flow method to determine the apparent movement in the image. The flow output is interpreted as a Game of Life, where living cells correspond to pixels in motion. The cells are tied to sonic grains that emit sound when activated. However, the only direction information they use is the horizontal component, which is used to trigger grains into proliferation or stasis. Thus, while their work uses optical flow for granular synthesis, they did not fully take advantage directional richness of flow properties. In Sonified Motion Fields [139] Pelletier uses FAST corner detector [153] to extract feature points and estimates the optical flow by using a pyramidal Lucas-Kanade method [194] to track the sparsely identified points.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

He also discusses the potential mappings of the flow field to sound parameters. However he concludes that the flow fields' temporal resolution is poor and the feature detection is not robust. *CaMuS*^{2} [151] estimates the optical flow from a camera phone. A 176x144 pixels image is divided into 22x18 blocks and cross-correlations between successive pairs of block images are calculated to obtain the optical flow. Because only 4 points are sampled in each block, the algorithm performs quickly. The simplicity of the method allows obtaining the global translation and rotation at 15fps, but the system is unable to provide rich motion information from the moving objects in the image.

However, to our knowledge, built-in cameras of mobile devices have not been used for sonification of moving elements in the camera image. Controlling sound through movement has always been of interest and can be traced back to Kurenniemi's DIMI-O [136] and Rokeby's Very Nervous System [2]. More recent examples such as [139] [81] [83] [64] use difference between images and optical flow to represent the moving parts of the image, but the methods have been either too simple to extract rich information or too heavy to be run on mobile devices. To overcome the too heavy - not rich enough trade off, we propose combining visual movement detection with the touchscreen of mobile devices.

4.2.1 THE PHONE WITH THE FLOW

(a) A person, a computer screen, a fan, objects passing by the window are movement sources.

(b) The user chooses a movement source

(c) The user selects a region on the screen

Figure 4.1: From scene to the the touchscreen.

In this section, we first present the main approach behind our system. It is currently implemented as an Android App (`www.caganarslan.info/pwf.html`). Optical flow features, extracted as described in section , are then mapped to sound synthesis parameters. The sound synthesis can be done either directly on the mobile device, with restrictions on the complexity of the synthesis due

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

to limited computing capabilities, or the features can be sent to external musical software via OpenSoundControl messages.

FROM THE SCENE TO THE TOUCHSCREEN

Our system relies on real-time spatial and temporal filtering of rich motion information provided by the built-in cameras of mobile devices. Fig. 4.1 depicts an example scenario of use of *The Phone with the Flow*.

Movement-rich Environment: The physical scene captured by the built-in camera offers a large range of movements, with various periodicity, directions and sources. The sources can be artificial (displays, mechanisms), natural (weather, animals, plants), or originate from people (user's body, other musicians, spectators, by-standers).

Choosing the Interaction Space: Mobile cameras have a limited field of view but their portability enables exploration of the surroundings by a point-and-shoot approach. Unlike fixed installations, the mobile camera's field of view can be changed without effort by simply changing its position and orientation. When the camera aims at a part of the movement-rich environment, objects in the field of view are captured, providing the user with a visual feedback of their movement sources. The user is then free to interact in the combined interaction volume [117] of the camera and the touchscreen.

Filtering the Movements: Once the movement sources are displayed on the touchscreen, the user focuses on a region of interest by touching it. The touch selects a region of the image to be analyzed further for detailed movement information. The user can change the position of the region by dragging their finger on the screen, alter its size, combine multiple regions and switch between them. The touch input thus enables filtering of the movements.

4.2.2 DESIGN SPACE

We present a four dimensional design space to explore the musical expression possibilities presented by *Phone with the Flow*. The dimensions can be combined to provide a large set of interaction techniques.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Movement Source

A smartphone camera enables the user to freely change the subject of the motion. We distinguish two values for this dimension:

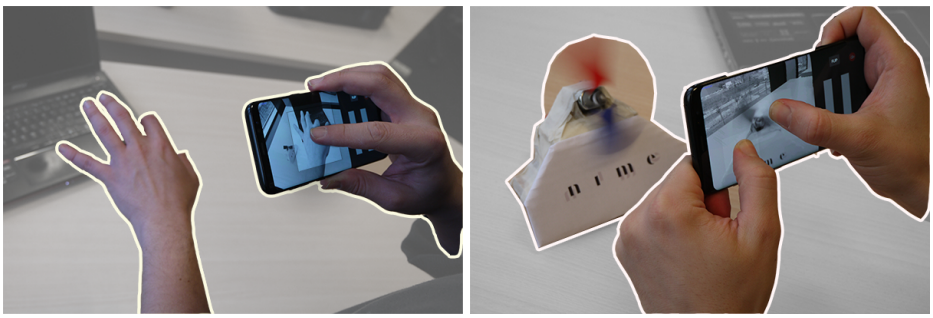


Figure 4.2: Movement sources. (Left) Self-motion, (Right) World-motion.

Self-motion: The holder of the camera is also the creator of the movements. The user can use its body parts to perform dynamic gestures. In the case where the user holds the camera with one hand, the other hand is free to create optical flow. Therefore, self-motion encourages bi-manual interaction on and behind the mobile device. The fingers permit creating movement in multiple directions simultaneously. Feet, legs, head and even the torso may also be the source of the movement. It is also possible to perform gestures by moving the device, as its relative motion with respect to the environment also creates an optical flow.

World-motion: The camera aims at an external source of motion. These sources can act intentionally to create controlled movements, such as a dancer, or be unaware of how they contribute to the process, such as spectators and other musicians on the stage. Objects that are continuously in motion may also be used to create sound, for instance, a busy highway, a waterfall, an assembly line and so on. Additionally, visual sources such as projections and displays may serve as sources. In a rich environment where there are multiple sources, the mobility of the device allows transition between them.

Camera Movement

This dimension is about how we adapt camera movement to complement the movement search.

Fixed: The camera is held at a fixed position. All of the optical flow is created by the moving objects in the static field of view. In practice, a hand-held device

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

is rarely in a fixed position because of hand tremor. However, the optical flow image is filtered to discard small displacements before obtaining the feature vector. Therefore hand tremor does not have an effect in this scenario.

Independent: The camera moves independently from the content of the environment. The optical flow is created by the motion of the camera with respect to the environment. Horizontal and vertical movements of the camera produce an output similar to that of an accelerometer, but the movement in the depth results in an optical flow in every direction as it produces a zoom-in-zoom-out effect. The device can also be rotated around three axis causing circular displacements of pixels.

Adaptive: The camera can track a moving object to strip it from either its translation or rotation. Or, if the object in motion is not rigid, its deformation can be stripped by adapting the camera movement to its trajectory. However, if the object does not fill the camera's field of view, the independent motion mentioned above distorts the optical flow. Hence, adaptive movement requires careful control.

Touch

The touchscreen offers valuable possibilities for filtering the image to discard the movements that are not desirable for the performer. The touch dimension has three values.

None: The touchscreen is not used along with the camera. All the movements in the camera image are processed continuously. This is ideal in controlled environments where no intervention is required. This attribute is best combined with a fixed camera to avoid continuous sound output of a moving device.

Activation: The touchscreen is used for temporal-only selection of movement. All movements captured in the camera image are therefore processed, but only while there is a finger touching the screen. This allows for discrete activation/deactivation of otherwise mostly continuous changes in the captured movements.

Isolation: Fingers are used to select a region of the camera image to focus on a movement source and discard the rest of the movements in the view. When there is no contact between the finger and the screen, nothing is processed. This enables both spatial and temporal filtering, making the system more robust to noise. Another advantage of isolation is decreasing the number of pixels to

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

process in order to lower the latency. Multiple regions may be activated at the same time.

Mapping

The mapping dimension describes the mappings between the finger position on the screen and sound parameters.

Independent from spatial position: Sound parameters are influenced only by the optical flow. Regions that encounter the same movement sound the same independently from their position.

Linked to spatial position: X and Y axes of the touchscreen are mapped to the sound parameters. The position of the finger determines the values along these axes. Consequently, the same movement results in different sounds in different regions of the camera image.

4.2.3 Evaluation

We evaluate what techniques are used by musicians when exploring the instrument. The goal of the evaluation is to determine how the users utilize the interaction space created by the combination of the camera and the touch screen.

10 participants (2 left-handed) volunteered to take part in the experiment. Their ages varied between 23-52 (mean 35, sd 9). All participants were involved in musical activities and were regular users of smartphones. The experiment was conducted on a 5.8" Galaxy Edge S8 smartphone running Android. The built-in camera provided images of 320x240 pixels at 30 fps. The feature vectors were sent via OSC messages to a standard PC running our Pure Data patches.

Protocol

We presented a mobile instrument which demonstrated our design space to the participants. In this instrument, the sound is produced by granular synthesis with 3 voices (one per finger). The mapping is *linked to spatial position* such that the horizontal and vertical touch position on the screen respectively changes the initial position in a sample and the initial frequency of the grains. While the

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Questions	Mean	SD
I made a lot of physical effort	2.65	1.05
I needed to concentrate	5.25	0.60
I felt frustrated/annoyed	2.6	0.99
I enjoyed interacting with the instrument	5.75	0.84
I was able to create a wide variety of sounds	4.35	1.18
I found my own techniques to play	4.9	1.14

Table 4.1: Likert Scale results

finger is touching, the grains that make up each voice then move around the sound (in position and frequency, as if browsing a spectrogram) according to the optical flow. The flow histogram gives the amplitude of displacement in 8 directions for the grains. The average flow amplitude changes the duration of the grains and the amount of moving pixels controls the volume. Participants were also able to change the size of the region that is activated by a touch.

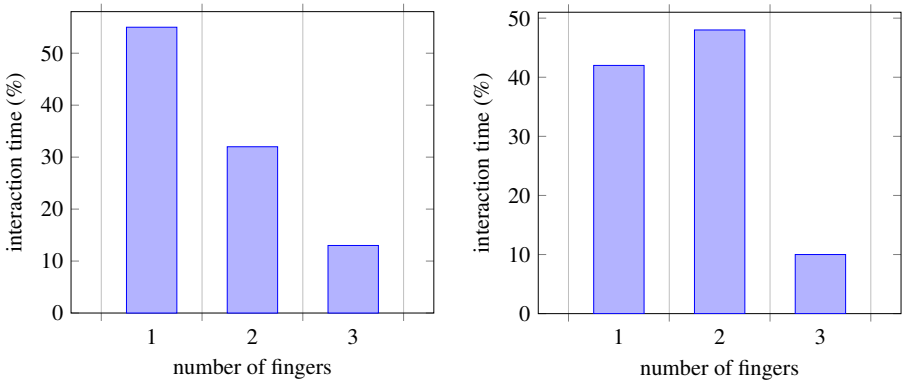
The experiment was conducted in two stages. In the first, the participants were provided with a display that could generate animations and a small fan as world movement sources. In the second they were instructed to use their bodies to create movement. For each stage they were given 5-10 minutes to explore and to prepare a 5 minute performance. During the performance, all parameter variations were logged in the Pure Data patch. The sound samples were altered between the steps. Both the order of steps and the samples were counterbalanced across participants. After the open exploration and the performances, participants responded to 7-point Likert-scale statements for which the results can be seen in table 4.1.

RESULTS

Without any instruction, all of the participants used their dominant hand to hold the smartphone and to use the touch input. They stated that orienting the smartphone and using the touchscreen at the same time was a difficult task for the non-dominant hand.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr



(a) self-motion (b) world-motion

Figure 4.3: Simultaneous finger use

We measured the frequency of simultaneous fingers used on the screen (Fig. 4.3). While performing hand gestures in the self-motion condition, the participants mostly used only one finger (55% 1 finger, 32% 2 fingers, 13% 3 fingers) to isolate the regions. However, when participants moved the camera with two hands, they were able to use multiple fingers. In world-motion, they often used both thumbs (42% 1 finger, 48% 2 fingers, 10% 3 fingers). Figure 4.4 also confirms the frequent use of both thumbs in world-motion.

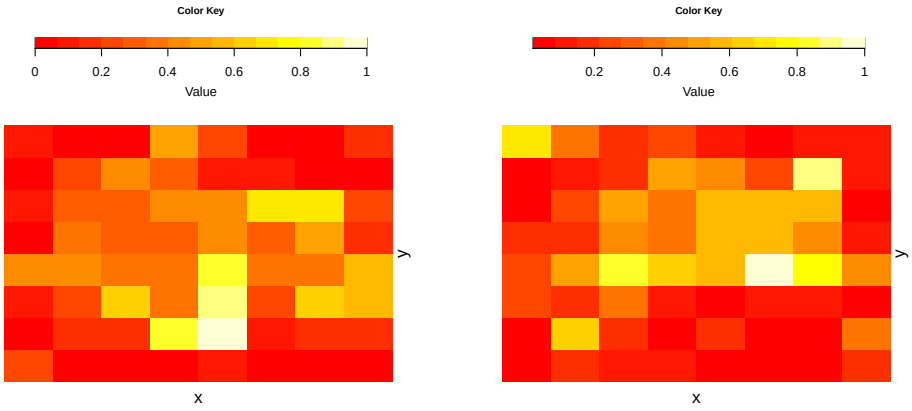


Figure 4.4: The screen use during user experiences. (Left) Self-motion, (Right) World-motion.

Figure 4.5 shows the average time it takes to calculate optical flow and to produce the feature vector for regions of different size. As the direction histogram and averaged optical flow values are calculated only from the moving pixels, computation time shows variation according to the captured movement. In

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

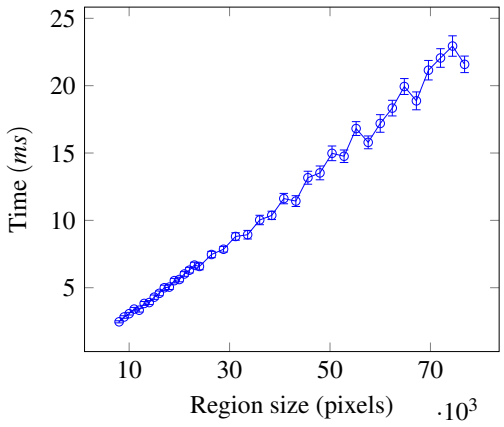


Figure 4.5: Computation time of the features vs region size

smaller regions, the amount of moving pixels do not vary significantly because the movements generally fill the whole region. Thus, the standard deviation of the computation time increases with the region size. We calculated the average region size used by the participants to be 16800 (140x120) pixels. This shows that using image regions instead of the entire image (76800 pixels) significantly decreases the computational time from 23ms to 5ms.

70% of the participants preferred using world objects instead of gestures. While the participants who preferred world objects argued that the instrument required concentration when performing hand gestures and using the touch input at the same time. The opponents all stated that gestural interaction allowed a finer control on the created movement.

DISCUSSION

We presented a novel approach to sonify visual movements by combining the touchscreen and the built-in cameras of mobile devices. In this context, the touchscreen proves to be a valuable tool to filter the visual input temporally and spatially. We also discussed the different interaction techniques enabled by our design space.

An interesting future work would be to combine the use of optical flow, i.e. motion features, with the other features in the images captured by the built-in camera, bringing objects textures, colors and shapes to the sonification process. Thereby, the camera would be working in two different contexts and our process

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

would require a logical fusion step. In addition, to make the interaction more heterogeneous, future work should incorporate the smartphone's IMU into the system. IMU could both be used to compensate the optical flow created by the smartphone's movement and add a new dimension to interaction by taking smartphone's orientation in 3D.

The usability of the Phone with the Flow arises out of the overlap of the display and the touch surface. This gives the user the opportunity of directly selecting the moving objects, therefore it does not need an additional feedback. However, other novel gestural instruments often lack visual feedback to be perfectly usable. In the next section, I propose adding a wide range depth camera to gestural instruments for providing visual feedback.

4.3 Adding visual feedback to sensors

Like RGB cameras, depth cameras are contextual sensors which have a high abstraction level. Their ability to sense the physical space in 3D makes them more popular then RGB cameras for spatial and embodied interaction. However, the opportunities offered by spatial interaction also brings challenges, particularly if 3D virtual objects are included. Interacting with virtual objects are easy with the new generation of VR systems where user's hands are also visible in virtual environment. Yet, in an AR setting, it is more challenging. Virtual objects are displayed using projectors, but they need a real object on which they are projected. The projections are then used as a visual feedback that guides the user and guides their movements.

In this section, I present our implementation of simplified versions of three existing gestural instruments, and demonstrate how they can be augmented using the depth camera space. It is important to point out the difference between a simple feedback mechanism and what was achieved with our contribution that was part of the Revgest project [24]. A simple feedback mechanism such as in the case of a cursor which shows the motion of a computer mouse does not constitute data fusion. The cursor on the screen moves only according to the mouse's physical movement. However in our contribution, the visual feedback reflected on the environment require additional information about the position and the state of the agents involved in the performance.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

4.3.1 REVEALING GESTURES

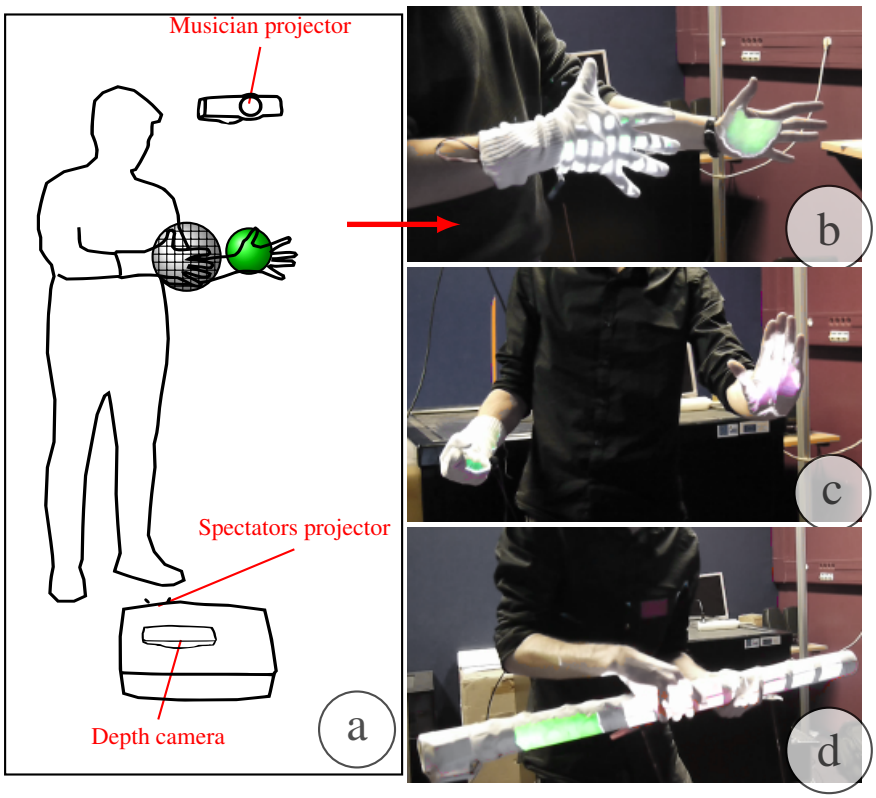


Figure 4.6: (a) Revgest in a setup with two projectors and one depth camera for public performance. The top projector is placed behind the musician and allows for feedback visible only to them. A virtual sphere is attached to the musician's right hand and provides feedback on finger movements sensed by a glove, another sphere in green controls a delay effect. (b) The resulting augmented gestural instrument. (c) Another glove based gestural instrument. (d) Augmentation of a handheld instrument.

Gestural interfaces, which make use of physiological signals, hand / body postures or movements, have become widespread for musical expression. While they may increase the transparency and expressiveness of instruments, they may also result in limited agency, for musicians as well as for spectators. This problem becomes especially true when the implemented mappings between gesture and music are subtle or complex. These instruments may also restrict the appropriation possibilities of controls, by comparison to physical interfaces. Most existing solutions to these issues are based on distant and/or limited visual feedback (LEDs, small screens)

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

From a hardware point of view, various signals from the body may be used, through devices that allow to use instrumented (or not) hand movements [112], finger movements and hand poses [33], or muscular activity [54]. Research on gestural instruments now provide knowledge about how to increase the transparency of Digital Musical Instruments through metaphors of physical world actions [60] or gestural sonic affordances [5]. They also create new opportunities for expression as they are more closely linked to the musician's body and as they remove some physical constraints of object-based instruments, e.g. on the amplitude of gestures.

In the context of computer-based instruments the lack, or limited use, of interaction with physical objects may also restrict feedback, both for the musician and the audience, and also restrict the appropriation [71] possibilities offered by the instrument. Visual feedback (with limited resolution) can be obtained by adding LEDs on the interfaces, for example using handheld devices as described by Hattwick et al. [76] or to glove interfaces such as the mi.mu gloves [121]. Mulder and Fels [125] used the deformations carried on virtual objects to perform and visualize sounds synthesis. Richer visual feedback is usually displayed distant from the interface, for example on a screen on stage in front of the musician, as done for some of the T-stick performances [112]. However, as shown by Berthaut and Jones on control surfaces [25], the visual feedback designed by musicians, if relevant, often requires higher resolution as well as co-location with the gestures. Outside the musical field, Sodhi et al. [166]] describe how 3D guides can help learn gestures when projected on the hand to indicate directions.

In order to display the virtual objects, our pipeline first needs to sense the physical space, by which the objects will be revealed and in which they will appear. The first possibility is to scan the physical space using a depth camera, with either structured infrared light or time-of-flight technology. From the image, a 3D mesh is created and transformed to world coordinates. The second possibility is to track revealing physical objects and assign their transformation to 3D models with matching shapes. Each physical element, depth camera and objects, is also assigned a unique identifier that can later be used to determine by which each virtual object was revealed.

4.3.2 Feedback and extended control

Originally Revgest is capable of providing virtual feedback in many configurations between the performer, spectators and objects, but not every configuration

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

requires heterogeneous data fusion. There exists, however, some configurations in which the performer has to take actions to stimulate both the gestural instrument and the depth sensor, thus creating a projection as a result fusion of two gestural modes. First, let us briefly resume the Revgest's design space.

The *attachment* dimension pertains to how virtual objects can be placed in the physical space and their physical relation with the musician's body:
-*attached to world (AW)* : the object is placed in the physical space around the musician, at absolute coordinates.
-*attached to body (AB)* : the object is attached to and follows the hand or other body parts of the musician.
-*attached to object (AO)* : the object is attached to a handheld or static physical object.
The attachments require the user to search for the virtual object. Surely, the user memorizes the location of the virtual object, but they have to consciously enter in the object's volume to activate the effects it produces.

The *feedback* dimension describes the type of information that can be displayed.
-*mappings feedback (FM)* : the object is used to identify the mappings between sensors and sound parameters, with text labels, textures or colours.
-*parameters feedback (FP)* : the object is used for feedback on musical parameters, in order to provide their context (curves, controlled waveform/timeline)
-*content feedback (FC)* : the object is used to provide the status of musical processes with for example VU meters, activation of sequences, and so on.
All types of feedback show information related to the state of the gestural instrument. For example, by changing the brightness of the projection it can show how bent the flex sensors are in a smartglove.

The *control* dimension pertains to how virtual objects are used for additional control of the music, complementing the expression offered by the gestural instrument.
-*no control (CN)* : the object is only used for feedback.
-*discrete control (CD)* : the object is used for discrete control, such as activating sequences or effects, triggering notes and so on. This corresponds to entering and leaving virtual objects, and going from revealing the surface to revealing the inside of objects.
-*continuous control (CC)* : the object is used for continuous control of musical parameters. The control may be a 3D position inside a virtual volume, a position along a path, a revealed color/texture. Example use cases include exploration of parameter spaces, playing through waveforms / sequences, exploration of

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

audiovisual textures, and so on.
Therefore both *CD* and *CC* requires user to perform gestures to interact with the depth sensor, while controlling parameters of the gestural instrument at the same time. The resulting feedback is a fusion of the gestures performed for the virtual objects, and the gestures on the instruments.

The discrete *CD* and continuous *CC* control of the instruments add an additional dimension to the sound creation process by activating filters and effects. We call this the *extended control* of the instruments. Next, I show the benefits of extended control by augmenting three existing instruments.

AUGMENTED INSTRUMENTS

Xth Sense

The first instrument takes inspiration from the xth-sense [54] instrument, which relies on MMG and EMG signals to control the sound. We recreate the system using the same glove described above. An additional pressure sensor on the palm activated when the hand is squeezed, evaluates the strength of the grasp. Movement speed is mapped to the volume and bending is mapped to a different voltage-controlled filter for each finger. Grasp strength is used to control a filter on the overall sound. Finally, the average finger extension is used to control the wet/dry on a reverb placed before the main filter, allowing one to trigger long reverberation by suddenly opening their hand. The extended control version, shown on Figure 4.6.b, adds three objects around the hand and attached to it. By intersecting these objects, the musician activates three separate delay effects. This version creates opportunities for appropriation as the objects can be revealed by any physical elements and move together with the hand. This version can be classified as *FP/FC, CD, AB*.

Soundgrasp

The second instrument is an adaptation of the musical glove presented by Mitchell and Heap [123]. We created a data glove to reproduce the SoundGrasp system. Four flex sensors positioned on the proximal interphalangeal joints except that of the thumb, measure the opening of the hand. The sensor outputs are connected to analogue inputs of a x-OSC board which sends OSC packets to the server. Our simplified version allows one to record their voice by opening the hand,and to loop the recorded phrase by closing it. Two effects can be applied

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

by extending one or two fingers. The extended control, shown on Figure 4.6.c, adds the possibility of creating a 3D soundpath placed in mid-air when the loop is played for the first time. This path is revealed by the musician and visible for both him and the audience. The musician may then activate and deactivate the playing loop by entering and leaving the path. The center and extent the section revealed in the path then controls the length and starting point in the loop. This version is classified the design space with the following values *FM/FP, CD/CC, AB/AW*.

T-Stick

The third instrument draws inspiration from the t-stick developped by Malloch et al. [112]. It consists in a tube equipped with various sensors. Sound parameters can then be controlled with the movements of the tube and by pressing, sliding, tapping the tube. In our version, 10 pressure sensors are placed along the length of the tube. They control the volume of 10 granular synthesizers which play ten positions along the same sound. The speed of the tube controls a global volume. In the extended control version, shown on Figure 4.6.d, three large zones are defined in the physical space. Each zone is associated to an effect, activated when the musician is inside the zone. In addition to the visual feedback given to both musician and audience on which effects are active, these virtual objects open expression opportunities, such as combining effects or playing with the borders of zones for glitches in their activation. This version can be classified as *FP, CD, AO/AW*.

Figure 4.7: T-Stick. (left) Visual feedback for musician (right) for the spectators

4.3.3 DISCUSSION

In this section, I argued that heterogeneous fusion can provide a rich visual feedback mechanism by incorporating the spatiality of gestures through a wide range depth camera. Essentially, the depth camera served to activate discrete

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

zones in the scene, and these active zones allowed users to perform finer gestures on instruments while obtaining a visual representation of the effects of their fine interaction with their instruments.

In the next section, I argue that a similar ample and fine gestures combination can be adopted by using two depth cameras which provide skeletal data at different precision levels.

4.4 A MULTISCALE DEPTH CAMERA

Depth-sensors which are based on camera technologies are perfect tools for non-instrumented interaction. They are versatile in sensor positioning, can be installed undisturbingly in public settings and retrieve user movements without additional guidance to users. Some depth-cameras provide skeletal information which is ready to use through vendor APIs. Therefore, they are practical for prototyping and developing user interaction. However, the dependence of camera technologies bring intrinsic constraints such as acquisition range, frequency and resolution. Combining multiple depth-cameras can provide a solution to the trade-off of these constraints. In section 3.3 I presented a lightweight calibration method for two depth-cameras with different precision levels. In this section, I demonstrate the affordances of that combination.

Our interactive environment consists of Microsoft Kinect 2.0 and Leap Motion sensors, which we consider are example of WRS and SRS, respectively. According to [84],the optimal distance between the object and the camera is between 1.0m and 2.5m. In [70], it is indicated that Leap Motion's range extends from 25 to 600 millimeters above the device. Thus, we place the devices as in fig. 4.8 so that the user's hands stay in the optimal zone for both sensors.

4.4.1 TEMPORAL AND SPATIAL RESOLUTION

We demonstrate an example use of our calibration method with the Kinect - Leap Motion acquisition couple. The main inadequacy of this system is Kinect's lack of finger tracking and Leap Motion's limited FOV. Thus, we propose a method to extend Leap Motion's field of vision to export the fingers to Kinect's world.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Figure 4.8: Sensor placement. 1. Kinect, 2. Leap Motion

According to [84], the measurement precision of the Kinect 2.0 sensor is between 1mm and 3.5mm, operating at 30 fps. On the other hand, [70] indicates that Leap Motion generates around 40 images per second with a precision of 0.5 - 0.01 millimeters. Consequently, Leap Motion has higher spatial and higher temporal resolution than Kinect.

The temporal resolution of the overall system is chosen to match that of Kinect. The image from Leap Motion is only acquired when the Kinect image is ready. This ensures having unique frames from both sensors even though the faster sensor produces multiple images. In the opposing scenario, a null frame is occasionally produced by Kinect between Leap Motion frames, thus the constant frame rate cannot be maintained.

On the other hand, there is no necessity in downgrading the spatial resolution of Leap Motion. The system has a variable spatial resolution: the ROI around the hands has Leap Motion's resolution, while the rest of the scene is as resolute as Kinect.

Skeletal tracking

In Kinect 2.0, there are 25 trackable joints located all around the body but here are only 4 joints on the hand: wrist, hand center, hand tip and thumb. The lack of hand joints makes Kinect difficult to use for proximal interaction as it limits the number of hand poses. In Leap Motion, all the joints of the hand are available but there are no body joints, which forces it to be a proximal interaction sensor.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

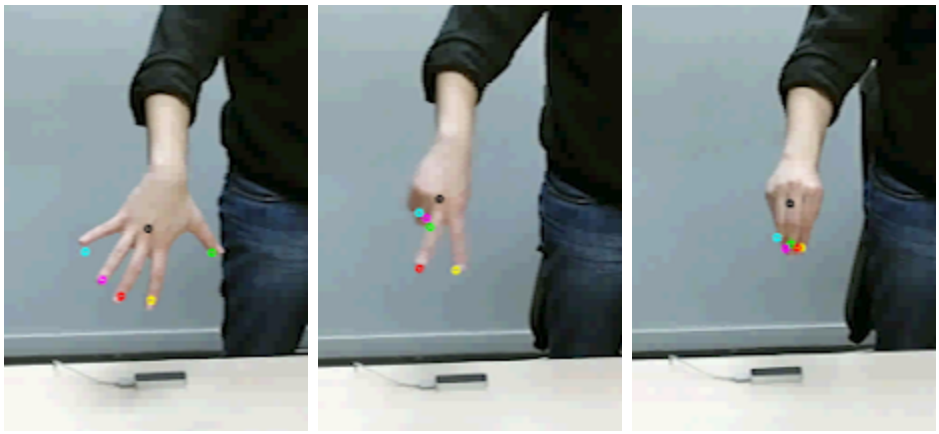


Figure 4.9: Projection of Leap Motion's fingertips to Kinect's coordinates.

Using the common hand center and wrist joints as seen in figure 3.8, We have transferred the fingertips to the Kinect's coordinate system (fig. 4.9). This transfer enables new interaction methods which is discussed in our design space.

4.4.2 Design space

In single device mode, the constraints of the devices force the user to interact from a certain distance. By combining the interaction spaces of both devices, we propose 3 interaction distances with targeted objects: *Distant*, *Proximal* and *Interspace*. But first, the SRS is manipulated tangibly select a target object.

Tangible interaction

Changeability of sensor positions due to continuous calibration permits the user to treat the sensing device as a tangible object. Instead of changing the target's position, user can grasp and move the SRS. However, because our calibration method is constrained to a horizontal surface for the SRS, it can only move horizontally. Figure 3.9 contains two targets; user can move the SRS in front of the selected object to interact with it.

Embodied interaction

The position of the user in the scene is a key parameter for the interaction. Positioning oneself away from the SRS indicates the indifference to the details,

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

while putting the hands on the SRS means that the user *reaches* to a target, showing their interest in interacting with it in a more precise manner.

Distant interaction: This mode is based on the natural usage of the WRS sensor. The gestures to perform are spacious and rough. The user's position and his limb movements are the main elements for the interaction. Distant interaction mode is selected naturally if the user's hands are not on the SRS.

Proximal interaction: This mode is based on the natural usage of the SRS sensor. The gestures to perform are precise and small. The user's fingers and the orientation of her hands are the main elements for the interaction. Proximal interaction mode is selected by placing hands above the SRS sensor.

Interspace interaction: We define interspace interaction as the ensemble of the user commands and gestures that are observed by both sensors. It uses both sensors in a complementary way to exceed their individual limits. In [117], extended hand gestures can be considered as an example of interspace interaction. Otherwise in [51], interspace also exists between the gametrak and the touch screen.

4.4.3 CONTINUITY OF INTERACTION SPACES

Tangibility of the SRS transports the local refinement of the interaction space to different zones in the environment. Interspace interaction allows bending the frame of the refined interaction volume. This can be achieved through registering a hand pose and combining it with arm movements.

Registration pose

In their taxonomy of multi-touch and whole-hand surface gestures [61], Freeman et al. described the registration pose as the state of the hand at the initial touch. Similarly, our registration pose is the state of the hand that triggers interspace interaction. A registration pose is defined by the state of the fingers (open/closed) and the angle between them. Figure 4.10.a shows the L-shaped registration pose where the only open fingers are the index and the thumb and the angle between them is larger than 60 degrees. There is no physical meaning of this pose, it is chosen purely for demonstration purposes.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Combining hand postures with large arm movements

In order to unify the gestures performed in two sensor spaces, user benefits from a sequential approach. Once the hand posture is validated on the SRS to select a virtual tool, the effects of the tool are applied to the environment by large arm movements. The whole arm can be used as a pointing device to apply the effect of the hand posture on a specific target, as well as using the continuous arm movements to apply the result of the hand gesture continuously on the environment.

Figure 4.10: An extended gesture through hand pose transfer.

Hand pose transfer

With registration pose performed on the SRS, we can transport the finger information to the global scene observed by the WRS assuming the user keeps the pose.

In a gesture performed in a multi sensor space, deciding on its start and end can be a problem. In our system, an interspace gesture is triggered by the registration pose, on the SRS, but it can end in either sensor's space.The gesture starts with the acceleration of the hand observed by the WRS, and ends when the hand returns to rest.

Between the start and the end points, the wrist and hand center joints are available. However, finger tip positions are lost once the hand is not observable to SRS. These lost points can be estimated by a similar approach to what we used for the lightweight calibration. At the beginning of the gesture, the length of the open fingers are saved along with their rotation with respect to the vector between the wrist and the center of the hand. When needed, the fingertip estimated on the WRS skeleton can be used outside. In figure 4.10.c, the trails of the fingers are

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

marked. The change in colors signify that the hand is no longer observable by the leap motion and that the points are estimated.

DISCUSSION

The combination of a WRS and a SRS allows us to dynamically adjust the local precision of captured hand gestures. At the same time it conserves classical interaction techniques offered by each sensor. Regardless of the lack of quantitative analysis of the reliability of the system we proposed, the concepts we have conceived exhibit the benefits of a heterogeneous system. It allows a richer interaction locally, and also provides a continuum between two devices. I go into more detail on the continuum to extend the interaction space, in the next chapter.

4.5 CONCLUSION

In this chapter, I proposed enriching the interaction space through heterogeneous data fusion. Thanks to the rich contextual information obtained by both RGB and depth cameras, we were able to incorporate the description of larger volumes into finer interaction. I presented three contributions. Combination of a mobile touchscreen and camera enabled us to filter motion information to obtain a faster and more expressive instrument. Then, combination of depth sensor and gestural instruments provided visual feedback and augmented existing instruments. Finally, to depth sensors with different ranges allowed refinement of skeletal data.

While demonstrating the spatial variation the local precision refinement, hand pose transfer from the short range sensor space to the wide range sensor space overflowed from the premises of enriching the interaction space. It created a continuum between two interaction spaces and allowed us to start a gesture with one sensor and finish it with the second one. In the next chapter, I elaborate on how we can use this type of continuum to extend the interaction space.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Five

Extending the interaction space with data fusion

5.1 Introduction

In the previous chapter, we demonstrated enabling of richer interaction techniques in the intersection of two interaction spaces. However, at the end, we briefly talked about the possibility of transitioning from one interaction space to another in a continuous manner. Next, I propose using this continuum to extend the interaction space of small devices. My aim is to demonstrate that the form factor of input devices can be changed by data fusion to accommodate new gestures.

The physical form of devices evolves in time as a result of years of development, testing and user feedback. Moreover, the tasks people accomplish on devices changes with time. New technologies in batteries, processors, sensors, displays also cause changes on devices; it is a constant development cycle. In the beginning of our millennium, tech companies were trying to make smaller phones. With the introduction of color displays and cameras, phones have gotten bigger. Now, with touchscreens and internet on our phones, they are much bigger because there is a lot more information to show on the screen. The increasing consumption of multimedia on our mobile devices also indicates that they will not be getting smaller any time soon.

As long as we use touch sensitive mobile devices for their design purposes, they are easy to manipulate. However, when we use them for new purposes as we often do with all devices in HCI, they have a limited interaction space. In

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

this chapter, I propose [11] extending the touchscreen of a hand-held device with an imaginary touch surface, fusing the interaction spaces of a smartphone and external cameras.

5.2 EXTENDING A MOBILE TOUCHSCREEN

Relative pointing through a tablet (PAD) on a large display is a viable technique which permits accurate pointing, but the nature of small screen of the PAD and the large screen of the display require a careful calculation of the cursor transfer function. Additionally, such systems requires clutching to move the cursor which degrades the performance. We present EXTENDED-PAD an indirect relative pointing system composed of a small touch surface (tablet) and the space around. A user can perform continuous relative pointing starting on the pad then continuing in the free space around the pad and within the arm's reach.

Large displays are increasingly popular and well adopted for public settings. Ray casting methods [41,87,126,135,141] and tablet based gestures [119] are mainly used to interact with those displays. However, although these techniques perform well for targeting and tracing [87], they can lead to major issues. For instance, using only ray casting methods can lead to hand jitter, fatigue and the lack of supporting surface decreases user performance [126]. On the other side, interacting with large displays differs from desktop and mobile use in pointer movements. In this context, either using absolute mapping or relative one can be problematic. For instance, absolute mapping suffers from the screen sizes difference [128]. In opposite, while relative pointing is a viable technique enabling accurate pointing, the nature of small screen requires a careful calculation of the cursor transfer function. In addition, the small size of touchpad requires clutching to move the cursor which degrades the performance [39].

To overcome such limitations, we propose combining touchscreen and the space around it to perform *extended continuous relative pointing gestures* which enable transitioning from the mobile touchscreen to mid-air. These gestures permit the creation of the *E-Pad*(short for Extended-Pad), that has its unified interaction space, and a modified control to display function to perform large cursor movements while being less tiring and faster than the aforementioned techniques.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Our contributions are as follows: (1) we introduce *extended continuous relative pointing gestures* and conduct a preliminary study to determine how the hand diverges upwards from the smartphone's plane as the motion continues around it,(2) we propose *E-Pad*, a novel technique for relative pointing on large displays, (3) we conduct an experiment to compare the performance of *E-Pad* with *Pad*, and (4) we derive three guidelines for pointing on large displays. We hope that *E-Pad* and our results will prove useful to designers and practitioners interested in indirect pointing on large display designs.

Related work on interacting with large displays

Ray casting methods are largely used to interact with the large displays. *Laser pointing* [41,135] and *image-plane pointing* [141] are the principal applications of ray casting methods. Although these techniques perform well for targeting and tracing [87], hand jitter, fatigue and lack of supporting surface decrease their performance [126]. Kopper et al. [99] also state that the performance of distal interaction systems depends largely on angular movements and sizes. Thereby, if the target and the user are at opposite ends of the display, the target is very difficult to acquire. In order to attack the precision problems, Vogel and Balakrishnan investigated gestural pointing techniques in air and concluded that ray casting may be faster, but less accurate than relative pointing [174].

In parallel, other researchers advocated for using tablet devices as a touchpad when interacting with large displays. This fact has led to a number of orthogonal design. For example, hand-held touch screens based on absolute mapping of the touch screen to the display have been employed to interact with large displays [111, 142]. However an absolute mapping is problematic due to the difference in screen size. Other researchers proposed using relative pointing [128] or combining both absolute and relative mapping by tapping to jump to the corresponding location on the screen, and then invoking relative motion with any finger movement [119]. Siddhpuria et al. conducted an experiment on distal pointing with everyday smart devices [165] and concluded that using smartphone with tho hands as a relative trackpad in landscape orientation gives the best results. Of course, relative pointing on a touchpad is a viable technique which permits accurate pointing, but the nature of small screen requires a careful calculation of the cursor transfer function. Pointer acceleration (PA) functions dynamically adjust the control-display gain, but they are implemented for desktop use in major operating systems [38]. Also, increasing the gain may deteriorate the performance due to hand tremor and quantification errors [82]. Additionally, according to Casiez

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

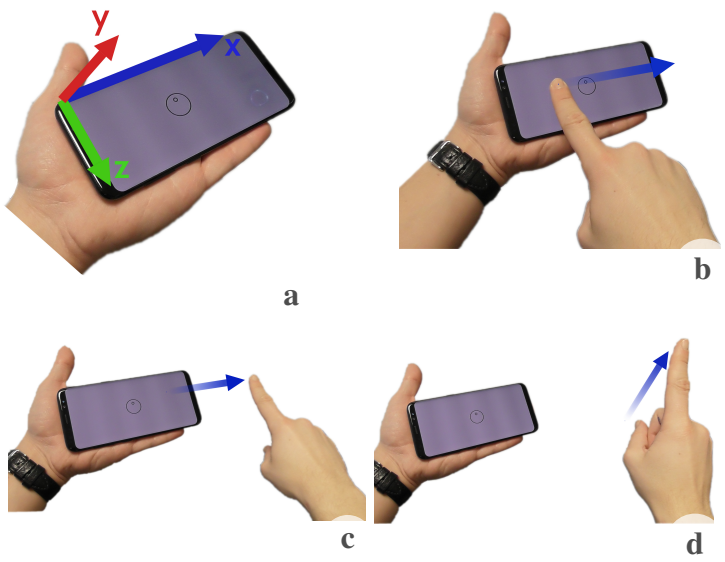


Figure 5.1: E-Pad functions: (a) Coordinates of E-pad; (b) pointing on the pad; (c) continuing in the air; and (d) releasing the cursor.

et al. [39] significant clutching to move the cursor degrades the performance. We should also state that, Nancel et al. [129] found that clutch-less movements were harder to perform and were not faster than clutch-enabled movements on touchpads.

Mixing touch and mid-air gestures is another approach to control large displays. In [29,107,195], authors benefit from switching between touch and air interaction for large tactile displays, but there is no continuity between the two modes. Yet, there exists a rich interaction space trapped between touch and air. Marquardt et al. [117] propose transition techniques that start on a touch surface and end in the air, and vice-versa. They generalize this type of gestures as *Extended Continuous Gestures*. From that point on, the community has started to employ the extended continuous gestures. For example, DeAraujo et al., [51] proposed an on-and-above-the-surface interaction techniques for creating and editing 3D models in a stereoscopic environment. Rateau et al., [146] proposed Talaria a drag & drop technique on a big wall display. Takashima et al., [169] introduced *Boundless Scroll* to decrease the number of clutches while performing scroll gestures on a touchscreen. *E-Pad*, the technique that we propose, builds upon this previous work to insure large display pointing in a continuous interaction space around a mobile device.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

5.2.1 Extended continuous relative pointing gesture

Marquardt et al. defined *Extended Continuous Gestures* [117] as "a gesture that a person starts through direct touch on the interactive surface can continue in the space above the surface to avoid occlusion of the digital content visible on the tabletop display". We propose an extended continuous gesture that continues in the space *around* a small hand-held touchscreen device. *Extended continuous relative pointing gestures* are complementary to regular pointing gestures, *i.e.*, they intervene when the limited physical surface of a hand-held touchscreen is insufficient to perform large cursor movements. These gestures start on the touchscreen and end in mid-air while the cursor is attached to the relative motion of the finger (Fig. 5.1.)

Preliminary Study: Finger Motion

During the implementation stage of *E-Pad*, we realized that in mid air, the dominant hand diverged upwards from the smartphone's plane as the motion continues. This observation is parallel to the the findings in *Boundless Scroll* [169]. Consequently, we decided to study the approximate imaginary surface on which the user performed the gestures when being around the smartphone surface. Thus, we conducted a preliminary study to observe the user's ability to interact spatially on and around a smartphone's touchscreen when pointing on a large display. We are mainly interested in determining the boundaries of the interaction volume and the pointing path in mid air.

Participants

Six men, average age 28. All participants were right handed and regular users of smartphone.

Apparatus

The preliminary experiment was conducted on two concatenated displays with a total resolution of 3840 $\times$ 1080 pixels resulting in a screen size of 130 $\times$ 37 centimeters. Participants were standing 2 meters away from the projection surface. We used a 5.8" Galaxy S8 smartphone (14,9$\times$37.8cm) to send the touch inputs via TUIO protocol. Additionally, four Optitrack cameras operating at 150Hz were in charge of tracking reflective markers on the pad and the user's hand as seen in

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

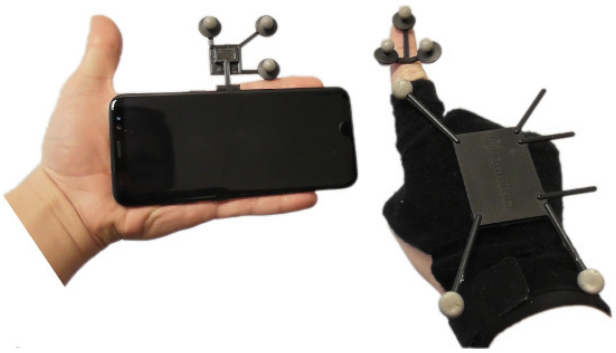


Figure 5.2: 3d printed markers on the smartphone and the user's dominant hand and the index finger.

Figure 5.2. The experiment was implemented in C++ and ran on two PCs (one for tracking and one for the display, each running on Windows 10.)

Procedure

We told participants that we were interested in determining users' performances when pointing on and around the smartphone surface for large display. This was intentionally misleading, since we were really studying how they unconsciously move their finger and the dominant hand around the smartphone surface as well as how they hold the smartphone. We then equipped our participants with reflective markers to create rigid bodies for the smartphone, the index finger and the dominant hand. This allowed us to obtain the position and the orientation of the smartphone, the finger's position with respect to the smartphone to move the cursor and the finger's position with respect to the hand to implement a reliable clicking technique independent from the smartphone's plane.

In the experiment phase, the participants were asked to stay standing while holding the smartphone comfortably using their non-dominant hand, and perform the gestures quickly using their index finger from the dominant hand. They were shown a start target they selected by tapping on the touchscreen. Once the start target was selected, an end target was displayed on the opposite side of the screen. The participants were required to select the end target with an air-click as a result of a continuous gesture (Figure 5.1). To simulate the motion on the cursor on the screen, we used the standard Windows 10 transfer function for both the finger position on the touchscreen and in mid-air. 3D finger positions were projected on the *Pad*'s plane to obtain 2D coordinates. In order to prevent the click gesture

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

from moving the cursor, we blocked the cursor when the finger accelerates away ($>$1cm/s) from the hand's plane. However a click is performed only if the finger reaches 15° from the hand. This threshold was empirically chosen.

The experiment was a 6×3 within-subjects design with two factors: pointing direction (east to west, north-east to south-west, north-west to south-east, west to east, south-west to north-east) and block (block1, block2, block3). We used a fixed amplitude (117 cm, with the amplitude corresponds to the distance between the start target and the end target) and a fixed tolerance (2.54 cm, with tolerance corresponds to the target circle radius). The experimental trials were administered as 3 blocks of 36 trials. Inside each block, 36 trials (6 directions $\times$ 6 repetitions) were randomly presented to each participant – a total of 108 trials per participant. The average duration of the experiment was 30 minutes.

Data collection

We recorded the position of the rigid bodies placed on the smartphone, the index finger and the dominant hand at 120Hz. We isolated the 108 extended relative pointing gestures per participant between the start target and the end target. The finger position captured by the Optitrack cameras was translated to the smartphone's coordinates. We did not log the finger position on the touchscreen as it can also be obtained from the Optitracks using the markers on the finger and the smartphone.

Results

First, all the participants held the smartphone almost parallel to the ground, grabbing it from the behind in order not to block the movements on different directions.

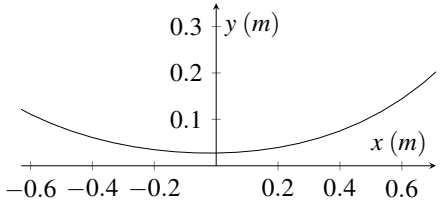
We constructed a point cloud of index finger positions from 648 trials. All the points were constrained in an imaginary box that exists between $-$63cm and 71cm on the x-axis, $-$12cm and 25cm on the y axis and $-$33cm and 22cm on the z axis.

We, then, fitted two forth order polynomials to the point cloud on the x-axis (see Fig. 5.3.a) and on the z-axis (see Fig. 5.3.b) to represent the average paths used by participants. The equations of the curves are as following:

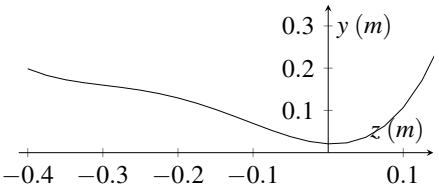
© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020



(a) on the x axis



(b) on the z axis

Figure 5.3: Estimated paths of the finger.

On the x-axis:

$$y = 0.03 + 0.01x + 0.23x^2 + 0.05x^3 + 0.12x^4$$

On the z-axis:

$$y = 0.02 - 0.008z + 6.38z^2 + 27.07z^3 + 33.45z^4$$

These curves confirm the hypothesis that the index finger deviates from the ideal smartphone plane as it moves away in mid-air. The deviation is similar on the dominant and non dominant hand sides, but as expected, the user has a larger reach on the dominant hand side. Because of the limited distance between the user and the smartphone, there is limited reach when user move downwards on the z-axis. On this direction, the finger deviates quickly.

DESIGN IMPLICATIONS.

Informed by our experimental findings, we outline two relevant guidelines for designing pointing techniques on and around a smartphone's touchscreen for large display:

- *Interaction volume.* The imaginary box obtained by the experiment shows that it's important to define an interaction volume rather than an interaction plane to perform the extended relative pointing. It is not important to limit the size of the interaction volume on the x and z axes as these dimensions depend on heavily on the arm's length. However, limiting the *depth* of the volume on the y-axis, we can present the users a mechanism to release the cursor, or to perform clutches in the air. Based on the extremities of the

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

point cloud we obtained, this depth can be chosen as 37cms, 12cms under and 25cms over the *Pad*.

- *Curved motions.* The curved nature of the obtained surface has to be taken into account in the control to display transfer function. On one hand, if the transfer function only uses the projected positions on the xz plane, the user wastes physical motion performed on the y axis while moving the cursor. On the other hand, we cannot add directly the y component of the motion into the equation. If the user deviates even more than the proposed curves, y component of the motion weighs heavily, causing stability issues. Also, using y component on both x and z axes result in interdependence between horizontal and vertical cursor motion. Therefore, we propose projecting the finger position on the xz plane, then using the length of the displacement vector on the curves (Figure 5.4).

5.2.2 E-PAD DESIGN

E-Pad is designed to overcome the limitations of indirect relative pointing system with a mobile device for large displays. We were inspired by extended continuous interaction techniques [117] that start the interaction on touch and continues in the air and we named our technique Extended-Pad (E-Pad). E-Pad is an indirect relative pointing system composed of a touch surface (smartphone) and the space around. A person can perform *extended continuous relative pointing gestures* starting on the pad, then continuing in the free space within the arm's reach.

The preliminary study we conducted allowed us to define the following functions for *E-Pad*:

Starting pointing on pad.

E-Pad contains the regular use of a trackpad (see Figure 5.1.b). Relative movements of the finger on the touch screen are mapped into the motion of the cursor on the screen with a conventional control to display transfer function. We use Windows' native pointer acceleration function to control the cursor [38].

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Exiting the *Pad* and creating the Imaginary Interaction Space (*IIS*).

When the size of the pad is not sufficient to move the cursor through a long distance, *E-Pad* permits the user to exceed the *Pad*'s frame. To enable the relative mid-air pointing, the finger should leave the touchscreen with a high velocity an should continue parallel to the smartphone's surface. The velocity threshold and the maximum angle between the direction of the finger on exit and the smartphone's plane were empirically chosen as 8 cm/s and 15° respectively. Once the transition to mid-air is completed, an *I*maginary *I*nteraction *S*pace (*IIS*) is created, its origin and orientation corresponds to that of the smartphone (see the axes in the Figure 5.1.a). The position and the orientation of the *IIS* are then fixed. This prevents variations of the coordinate system of the finger while it is in mid air and provides a better stability for the cursor on screen. *IIS* is open ended on smartphone's plane, but it has a height of 37cms (see 5.2.1). *IIS* encompasses an optimal surface for the finger motion while permitting deviations from the optimal path.

Pointing in mid-air.

In order to maintain the continuity between the *Pad* and around the *Pad*, we use Windows' native pointer acceleration function with the same gain multiplier for pointing in the mid-air. However, in the light of the preliminary study, instead of taking the planar components of finger's motion as the input displacement vector, we used the displacement on the obtained curves. More precisely, we first obtained the displacements on the x-axis and the z-axis, and then used the length of the curve corresponding to those displacements. Taking the y-component of the displacement would create a dependence between the x and z axis, meaning any explicit gesture to create an horizontal cursor motion would result in a vertical

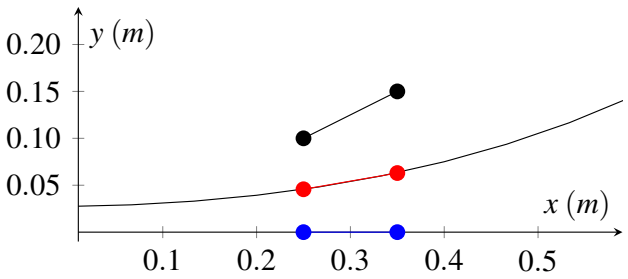


Figure 5.4: An example of displacement vector estimation on the x axis. Black: Displacement of the finger. Blue: Projected displacement. Red: Displacement obtained by our method.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Figure 5.5: Clutching in the air

cursor motion and vice-versa. Yet, using the direct projections of the displacement on the x and z axes would mean that only a part of the user's physical effort was used the move the cursor. Thus, the polynomials we obtained in the preliminary study permits us to compromise between the two options (Figure 5.4).

Clutching in mid-air.

The height of the *IIS* ensures that user's hand stays in the interaction volume without the guidance of a physical surface in mid-air. If the user wants to perform a clutch, in air, they can quickly leave the *IIS*, clutch and re-enter *IIS* in 2 seconds. The user can also land on the *Pad* once they re-enter the *IIS*. The cursor control method we described above permits the user to lift their hand without disturbing the cursor. By doing so, contrarily to a clutch on a physical device, in our case the smartphone, the users can made a clutch by lifting their hand over the upper limit but also by lowering it under the bottom limit of the E-Pad space, *i.e.*, the IIS limits.

Clicking in mid-air.

The air click technique we described in the procedure of the preliminary study can be used anywhere in the *IIS*.

Releasing the cursor.

In order to detach the cursor from the hand, the user simply leaves the *IIS*. The user can also lower the smartphone to invalidate the *IIS* and end the interaction. Of course, if the user needs then to use the E-Pad again, they can create a new *IIS* that corresponds to the position and orientation of the device when the user exits the *Pad* condition.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

5.2.3 EXPERIMENT

We conducted an experiment to compare performance between *Pad* and *E-Pad* techniques. Based on the theoretical ability (*i.e.*, the limited surface) of *Pad* technique to point on the large display, we hypothesize that *E-Pad* will **(H1)** improve selection speed while **(H2)** decreasing the number of clutches. The third hypothesis **H3** is that *Pad* will be more accurate than *E-Pad* as the touch click should have less systematic error than mid-air click.

PARTICIPANTS

12 participants (1 female) volunteered to take part in our experiment. Participants' ages varied between 24 and 32 years (mean age 26.66, sd=2.67 years). All participants were right-handed. All participants except one were regular users of smart phones or tablet devices with multi-touch screens, and 4 participants were regular users of Kinect games.

APPARATUS

The experiment uses the same apparatus as in the preliminary study.

TASK, PROCEDURE AND DESIGN

We used a reciprocal two dimensional pointing task (Figure 5.6) on 6 targets (only one visible at a time) that were positioned on an imaginary circle). The participants were instructed to stay standing while holding the smartphone with their non-dominant hand. The participants were, then, instructed to click on the targets in random order to complete a clicking sequence of 6 motions in different directions. Each direction consisted of a *start target* and an *end target*. Each trial began after the start target was successfully selected with a click on the touchscreen. The trial ended with the selection of the end target with a touchscreen click in the case of the *Pad* technique and an air click in the case of the *E-Pad* technique. The touchscreen click is potentially possible with *E-Pad* if the user does not leave the pad surface. After a *start target* was selected, it disappeared and the corresponding *end target* was displayed. In case a participant

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

missed the *end target*, it disappeared and the start target was displayed again, logging an error for the trial. Participants had to successfully select the *end target* before moving to the next *start target*, even if it required multiple attempts. Each pointing sequence was repeated 3 times.

Dependent measures are analyzed using a $2 \times 3 \times 3 \times 6$ repeated measures within-subjects analysis of variance for the factors *Technique* (*Pad* and *E-Pad*), *Block* ($1-3$, with the first block serving as opportunity for learning the new method), *tolerance* (*L*: 75px (5.08cm); *M*: 38px (2.54cm) and *S*: 13px (0.85cm), with *tolerance* corresponds to the target circle radius), *direction* (east to west (*EW*) , north-east to south-west (*NE-SW*), north-west to south-east (*NW-SE*), west to east (*WE*), south-west to north-east (*SW-NE*) and south-east to north-west *SE-NW*).

The amplitude (*i.e.*, the distance between the start target and the end target) was kept constant at 117 cm. We decided on this longer single distance because we think that the issue of clutching when using a smartphone is more recurrent when the target distance was longer even if we think that the benefits of *E-Pad* still consistent with short distance as E-pad can be used as pad for short amplitude. This decision was taken to reduce the duration of the experiment and to highlight the effect of the proposed technique. The rationale was also that if no effect was found with these settings, it would be likely that no such effect exists.

In the experiment phase, the two *techniques* were randomly presented to the participants. Inside each *technique*, participants completed three *blocks*. Inside each *block*, the three *tolerances* were randomly presented to the participants. For each *tolerance*, the pointing sequence (*i.e.*, the six directions) were repeated 3 times. The initial direction of each pointing sequence was randomized – a total of 324 (=2 techniques $\times$ 3 blocks $\times$ 3 tolerances $\times$ 6 directions $\times$ 3 repetitions) trials per participant. After each block of trials, participants were encouraged to take a pause.

After each technique, participants respond to 5-point Likert-scale questions (strongly disagree to strongly agree): i) I performed well, ii) I accomplish the task rapidly, iii) I need a lot effort to finish the task, iv) I need to concentrate to accomplish the task; v) I feel frustrated/stressed/irritated/annoyed; vi) I felt confident in my ability to hit the target; vii) I enjoyed interacting with the device(s)." At the end of the experiment, participants were asked to rank the two techniques according to their preferences. The average duration of the experiment was 1 hour and 10 minutes.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

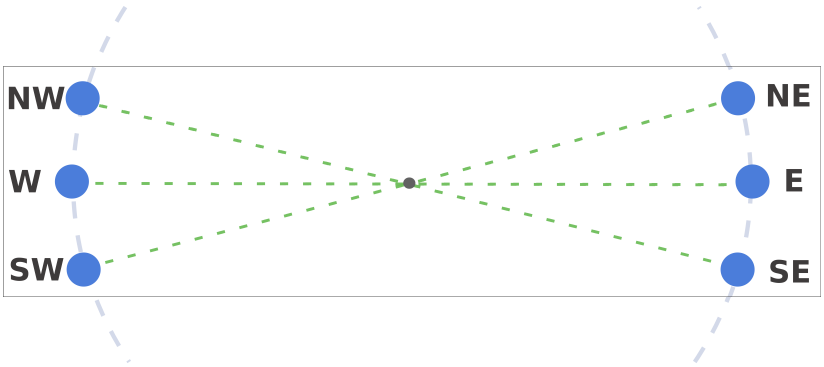


Figure 5.6: Target positions and movement directions.

5.2.4 RESULTS

The dependent measures are *movement time*, *number of clutches* and *error rate*. We also analyze the subjective responses. All analyses are multi-way ANOVA. Tukey tests are used post-hoc when significant effects are found.

MOVEMENT TIME

Movement time is the main dependent measure and is defined as the time between the click on the start target and the click on the end target.

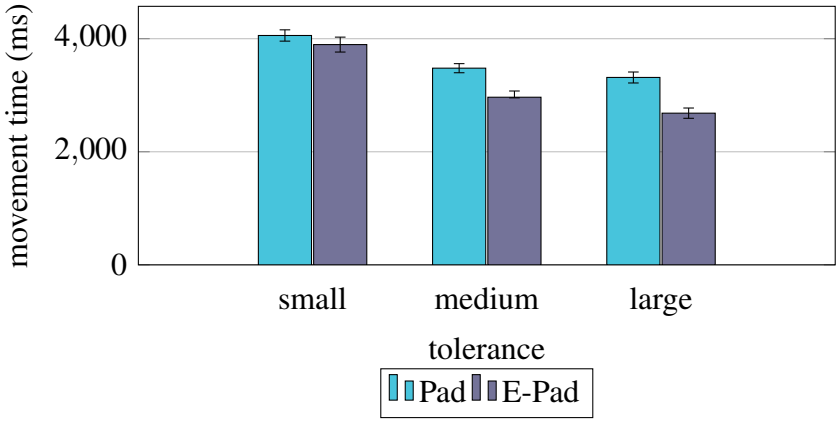


Figure 5.7: Effect of tolerance on movement time

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

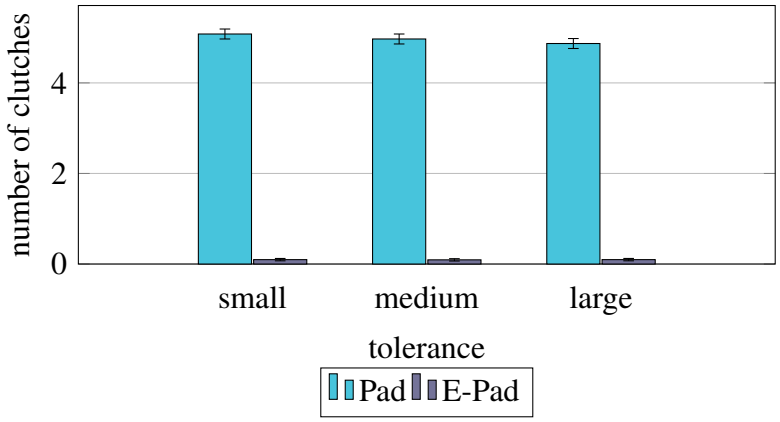


Figure 5.8: Effect of tolerance on number of clutches

Repeated-measures ANOVA revealed a significant effect of *block* on *movement time* ($F_{2,22}$=4.76, p<.0001). Post-hoc tests showed a significant decrease in the time between the first block and the two remaining (p<.05; block1: 3630 ms, block2: 3439ms, block3: 3359ms) due to a learning during the first block. As we are concerned with user performance after familiarization, the remaining analysis discards the first block.

There were significant main effects of *technique* ($F_{1,11} = 12.48$, p = .0047), *tolerance* ($F_{2,22} = 142.92$, p < .0001) and a significant *technique* × *tolerance* interaction ($F_{2,22} = 16.10$, p < .0001) on *movement time* (Fig. 5.7). Post-hoc tests revealed that *E-Pad* was significantly faster than *E-Pad* for the medium and the large tolerance sizes (p < .05) by respectively 14.78% and 19.06%, supporting partially **H1**. We also, found that with *Pad*, *movement time* is significantly higher with small tolerance size than both medium and large tolerance sizes (p<.05). While, with *E-Pad*, *movement time* increased significantly as the *tolerance* decreases (p<.05). There were no significant *technique* × *direction* × *tolerance* (p=.33) interaction, suggesting that the benefits of *E-Pad* with the medium and the large tolerance are consistent across directions.

CLUTCHING

We analyzed the *number of clutches* used on the pad surface, assuming that frequent clutching indicates high physical workload. Repeated measures ANOVA showed significant main effects of *technique* ($F_{1,11}$=1067, p<.0001) and a significant *technique* × *tolerance* interaction ($F_{2,22}$=3.88, p=.0360) on

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

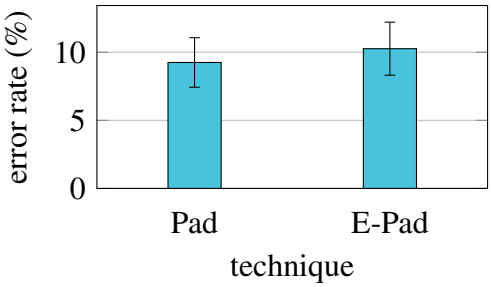


Figure 5.9: Effect of technique on error rate

number of clutches. As *E-Pad* (mean .09, s.d .01) allows continuous pointing to be maintained without movement, it is unsurprising that they have significantly less clutching than *Pad* (mean 4.97, s.d .06) with the three tolerance sizes ($p < .05$), supporting **H2**. Additionally, for *Pad* technique, Post-hoc tests revealed that the number of clutches increased significantly as the *tolerance* decreases ($p<.05$). Interestingly, we found that there was no significant *technique* $\times$ *direction* ($p=.31$) nor *technique* $\times$*tolerance* $\times$ *direction* ($p=.45$) interactions, suggesting that the benefits of *E-Pad* over *Pad* are consistent across the different *tolerances* and *distances*.

In the *E-Pad* technique, even though both the upper and lower limits could be used for a clutch, participants were observed making the clutch only by lifting their hand/finger using the upper limit of the IIS.

ERROR RATE

Targets that were not selected on the first attempt were marked as errors. There were no significant main effects nor interaction ($p>.05$) on *error rate* with *Pad* (mean 9.25%, s.d. 1.82), *E-Pad* (mean 10.26%, s.d. 1.95), suggesting that there was no significant difference between *Pad* and *E-Pad* and so leading to rejection of **H3**.

SUBJECTIVE RESULTS

We recall that participants were asked to rank the two techniques conditions after completing the experiment. Overall, *E-Pad* was ranked 100% first.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

	Pad		***E-Pad***		***Wilcoxon***
	Mean	s.d	Mean	s.d	Z
Performance	3.83	.47	3.5	.61	.96
Rapidity	3.08	.37	3.66	.65	-1.41
Physical	2.58	.65	2.41	.70	.32
Concentration	2.66	.81	2.83	.79	-.41
Frustration	2.41	.78	2	.59	.89
Confidence	4.41	.50	4.16	.63	1.73
Enjoyment	3.25	.48	4.16	.53	-2.75

Note: Wilcoxon-Signed-Rank tests are reported at p=.05 (*) significance levels. The significant tests are highlighted.

Table 5.1: Mean and s.d questionnaire responses, with 1=strongly disagree, and 5 = strongly agree.

Participants were also asked to rate each technique. Overall, questionnaire responses (Table 5.1) show that mean ratings for *E-Pad* were mostly more appreciative, but only significantly for enjoyment.

We correlate these findings with comments from participants who felt that the PAD technique was "*boring*" and "*repetitive*". Multiple participants stated that swiping continuously on the screen forced their wrists. One participant said: "*this really hurts my wrist*". Besides multiple participants felt that the friction on the touchscreen hurts the finger in the long term. One person said: "*I don't feel my fingertip anymore*".

In contrast, participants stated their satisfaction with *E-Pad*. Some quotes are: "*I'm happy... the precision is not an issue while my finger is in the air*" and "*I definitely prefer this technique*". Most of the participants affirmed, also, that clicking in mid-air was not difficult. One participant said: "*This is much easier than I imagined*". However, some participants stated a few concerns about the E-PAD. Two participants said that distinguishing the y-axis and the z-axis was difficult in the beginning. One of them said: "*How do you go upwards again?*". Three participants declared that pointing to their non-dominant side was more difficult. However, as mentioned above, we did not find a significant effect of *direction* on the *movement time*.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

5.2.5 DISCUSSION AND DESIGN GUIDELINES

Our key finding is that *E-Pad* improved selection speed, decreased the number of clutches and increased enjoyment over conventional *Pad*, without compromising accuracy. The performance benefits were consistent across different *tolerances* and *directions*.Our findings indicate that *E-Pad* is faster than *Pad* by up to 19.06%, decreases the number of clutches, without compromising accuracy. Our analysis also suggests an overwhelming preference for *E-Pad* instead of the *Pad*.

Informed by our experimental findings, we outline relevant guidelines for designing pointing techniques on large displays:

- Touchscreen of the smartphone should be used prudently for selecting distant targets on large display. Indeed, our participants often expressed dissatisfaction when making distant target selections feeling that it requires longer selection time and bigger number of clutches.
- Users should be provided with a physical reference when making mid-air indirect pointing on large display as our participants insisted on the confidence brought by having the smartphone as a reference when switching to mid-air modality.
- Designers should conceive flexible input that allow users to continue pointing in the space around the smartphone when the surface of the pad is not sufficient to continue the pointing task.

In our experiments, our participants were standing while taking the smartphone with their non-dominant hand which can increase the fatigue and change the behaviour of the user when compared to having the smartphone on a desk and the users set down. Thus, future work will investigate the effect of having the smartphone on a desk on both Extended continuous relative pointing gesture behaviour, the design of *E-Pad* and then on the performance of *Pad* and *E-Pad*. Additionally, one potential usability issue of our techniques is that when switching to the midair modality, there were no visual feedback for the pad extended surface, the only visual reference is actually the tablet. To visually help the user, the technique could add a visual feedback of the extended pad using revealed virtual objects [24]. Finally, to omit the dependency on external cameras, future work will explore the use of magnetic sensing [192] and microphone arrays [130] for *E-pad* technique.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

5.3 CONCLUSION

In this chapter, I proposed extending the physical boundaries of input devices by data fusion. To do so, I introduced an Extended Continuous Relative Pointing Gesture, a complementary pointing gesture to perform large cursor movements by allowing users to continue the pointing gesture in the air around the touchscreen when the physical surface is limited. Our preliminary study indicated that when switching to the air space, the hand motion deviates from the touchscreen plane and follow a curved trajectory. The combined interaction space utilized the new trajectory to compensate the physical effort lost in the traditional transfer functions.

I expect similar modifications in gesture trajectories in other combinations of interaction spaces which constitute a continuum and differ in physical support. This is due to two factors. First, the gestures performed in a larger space requires a bigger part of human body to move, therefore the kinematic chain is longer [69]. Second, the bias for existing interaction techniques for each device affect user behavior.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Six

Conclusion

In this dissertation I explored a data fusion approach for human computer interaction which combines both information and the physical spaces in which users perform gestures. I attempted to broaden the existing definition of data fusion to include humanly aspects of interaction. While reviewing the literature, I tried to be chronological to explain where data fusion originates and how advanced contextual input devices enabled promising heterogeneous environments which need particular attention.

Our contributions were divided into three chapters. First, we discussed the extraction of relevant information from individual sources so that devices are ready to be fused on characteristic level. The high abstraction level of cameras allowed us to show that even when the input is complex, it can be processed efficiently to obtain highly descriptive features in computationally constraint scenarios. We also saw that for the features of multiple sources to be meaningful, they should be put into the same coordinates. Second, by using the features obtained from contextual sensors, we discussed the how to add value to the observations that stay in the intersection of two interaction spaces. We exploited the intersection to filter data, to add visual feedback and to work at multiple scales. Third, instead of limiting the interaction to the intersection of interaction spaces, we expanded it. Extended continuous gestures helped us to increase the performance of large screen pointing and also proved that the gestures are reshaped at the extension of the new interaction space.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

Throughout this document, I tried to provide as many design guidelines as possible to underline the key points which should be considered when two inputs are combined. These points sum up to lightweight processing of inputs, distributing interaction modes around devices and taking gesture trajectories into account. It is my hope that my work gives insight to future research around multi-sensor interaction. Some of the open research topics I am interested in are resumed in the following section.

6.1 FUTURE RESEARCH

To begin with, there are two intended improvements to the work presented in this thesis. First, depth-camera calibration and the design space of multiscale skeletal fusion requires an application to test the concepts proposed in the respective sections. It is an ideal candidate for contactless interaction in an environment which contains a number of target objects. For instance, *Damassama* [31] features this kind of setup, but it only allows interaction from a longer distance with ample gestures. Second, E-Pad could benefit from a visual feedback mechanism that we introduced in section 4.3. It would guide the user by projecting gradual colors on the user's hand to indicate the optimal path and the limits of the interaction volume.

Furthermore, continuous interaction spaces demand further investigation. An overwhelming proportion of research conducted on this subject focuses on the combination of touch surfaces and mid-air. Even though we proposed similar gestures for mid-air to mid-air transition (section 4.4), transition between other types of devices and interaction spaces are open to investigation. For example, the motion information obtained from *independent* camera movement in *Phone with the Flow* (4.2.2) can be replaced with information obtained from the on-board IMU, and provide a transition between *fixed* and *adaptive* use of cameras. Additionally, the combination of hand gestures in virtual reality systems and real world objects are open to examination for various transitions.

Finally, the question that was asked in the introduction still stands. *Can we combine existing materials for a new interaction technique instead of waiting for new hardware to obsolete older technologies?* . In this dissertation, we relied on mostly consumer-grade contemporary technologies to demonstrate that novelty stems from data fusion without producing new devices. Combinations which were studied all permit dismantling and reuse of each component in accordance

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

with sustainable design guidelines. However, to have a mainstream impact, a more pragmatic question is: are these combined systems usable outside of university libraries? Are they easily configurable, maintainable and replaceable when they break down? Otherwise, our collective aim and effort becomes a mere retrofuturistic concept of bulky, messy, chaotic interactive environments. We say that human-computer interaction is an interdisciplinary field of research which incorporates engineering, design, cognitive sciences and more, but perhaps our research should include more elements from economics, sociology and environmental studies to create the interfaces of *tomorrow*.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Bibliography

[1] Difference between the terms movement and motion.

[2] Very nervous system. `http://www.davidrokeby.com/vns.html`. Accessed: 2018-3-28.

[3] N. Ahmad, R. A. R. Ghazilla, N. M. Khairi, and V. Kasi. Reviews on various inertial measurement unit (imu) sensor applications. *International Journal of Signal Processing Systems*, 1(2):256–262, 2013.

[4] B. Allaert, I. M. Bilasco, and C. Djeraba. Consistent optical flow maps for full and micro facial expression recognition. 2017.

[5] A. Altavilla, B. Caramiaux, and A. Tanaka. Towards gestural sonic affordances. In *Proceedings of NIME*, pages 61–64, May 2013.

[6] Y. A. Ameur and N. Kamel. A generic formal specification of fusion of modalities in a multimodal hci. In *Building the Information Society*, pages 415–420. Springer, 2004.

[7] T. Amiaz, E. Lubetzky, and N. Kiryati. Coarse to over-fine optical flow estimation. *Pattern recognition*, 40(9):2496–2503, 2007.

[8] M. Ananya, G. Essl, and M. Rohs. Microphone as sensor in mobile phone performance. In *Proceedings of NIME*, pages 185–188, Genoa, Italy, 2008.

[9] C. Arslan, F. Berthaut, J. Martinet, I. M. Bilasco, and L. Grisoni. The phone with the flow: Combining touch+ optical flow in mobile instruments. In *NIME 2018-18th International Conference on New Interfaces for Musical Expression*, 2018.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

[10] C. Arslan, I. M. Bilasco, and J. Martinet. Dynamic index finger gesture video dataset for mobile interaction. In *2018 International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–4. IEEE, 2018.

[11] C. Arslan, Y. Rekik, and L. Grisoni. E-pad: Large display pointing in a continuous interaction space around a mobile device. In *Proceedings of the 2019 on Designing Interactive Systems Conference*, pages 1101–1108, 2019.

[12] I. Aslan, I. Buchwald, P. Koytek, and E. André. Pen + mid-air: An exploration of mid-air gestures to complement pen input on tablets. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*, NordiCHI '16, New York, NY, USA, 2016. Association for Computing Machinery.

[13] D. Avrahami, J. O. Wobbrock, and S. Izadi. Portico: tangible interaction on and around a tablet. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 347–356, 2011.

[14] A. Baghaie, R. M. D'Souza, and Z. Yu. Dense descriptors for optical flow estimation: a comparative study. *Journal of Imaging*, 3(1):12, 2017.

[15] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*, 56(3):221–255, 2004.

[16] C. Baldé, V. Forti, V. Gray, R. Kuehr, and P. Stegmann. The global e-waste monitor 2017: Quantities, flows, and resources, 12 2017.

[17] T. Ballendat, N. Marquardt, and S. Greenberg. Proxemic interaction: Designing for a proximity and orientation-aware environment. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, page 121–130, New York, NY, USA, 2010. Association for Computing Machinery.

[18] P. Baudisch and G. Chu. Back-of-device interaction allows creating very small touch devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1923–1932, New York, NY, USA, 2009. ACM.

[19] M. Beaudouin-Lafon, S. Huot, M. Nancel, W. Mackay, E. Pietriga, R. Primet, J. Wagner, O. Chapuis, C. Pillias, J. Eagan, et al. Multisurface interaction in the wild room. *Computer*, 45(4):48–56, 2012.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

[20] H. Benko and A. D. Wilson. Depthtouch: Using depthsensing camera to enable freehand interactions on and above the interactive surface. In *In Proceedings of the IEEE Workshop on Tabletops and Interactive Surfaces*, page 2009.

[21] N. O. Bernsen. Modality theory in support of multimodal interface design. In *Proceedings of the AAAI Spring Symposium on Intelligent Multi-Media Multi-Modal Systems*, pages 37–44, 1994.

[22] N. O. Bernsen. A taxonomy of input modalities. *Information theory and information mapping, Amodeus project deliverable D*, 15, 1995.

[23] G. A. Berry, V. Pavlovic, and T. S. Huang. Battleview: A multimodal hci research application. In *Workshop on Perceptual User Interfaces*, pages 67–70. Citeseer, 1998.

[24] F. Berthaut, C. Arslan, and L. Grisoni. Revgest: Augmenting gestural musical instruments with revealed virtual objects. In *International Conference on New Interfaces for Musical Expression*, 2017.

[25] F. Berthaut and A. Jones. Controllar: Appropriation of visual feedback on control surfaces. In *Proceedings of ACM ISS*, pages 271–277, 2016.

[26] B. Blazica, D. Vladusic, and D. Mladenic. Ubiquitous personalization of a smartphone, used as a universal controller. 2012.

[27] E. Blevis. Sustainable interaction design: Invention & disposal, renewal & reuse. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, page 503–512, New York, NY, USA, 2007. Association for Computing Machinery.

[28] R. A. Bolt. "put-that-there" voice and gesture at the graphics interface. In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, pages 262–270, 1980.

[29] A. Bragdon, R. DeLine, K. Hinckley, and M. R. Morris. Code space: Touch + air gesture hybrid interactions for supporting developer meetings. In *ITS '11, ACM*, pages 212–221, New York, NY, USA. ACM.

[30] A. Bragdon, R. DeLine, K. Hinckley, and M. R. Morris. Code space: Touch + air gesture hybrid interactions for supporting developer meetings. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '11, page 212–221, New York, NY, USA, 2011. Association for Computing Machinery.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

[31] N. Bremard, L. Grisoni, and B. De Araujo. Interaction events in contactless gestural systems: from motion to interaction. In *Proceedings of the 2014 International Workshop on Movement and Computing*, pages 166–169, 2014.

[32] R. A. Brooks. The intelligent room project. In *Proceedings Second International Conference on Cognitive Technology Humanizing the Information Age*, pages 271–278. IEEE, 1997.

[33] D. Brown, N. Renney, A. Stark, C. Nash, and T. Mitchell. Leimu: Gloveless music interaction using a wrist mounted leap motion. In *Proceedings of NIME*, 2016.

[34] D. T. Buechler, N. N. Zyaykina, C. A. Spencer, E. Lawson, N. M. Ploss, and I. Hua. Comprehensive elemental analysis of consumer electronic devices: Rare earth, precious, and critical elements. *Waste Management*, 103:67–75, 2020.

[35] W. Buxton. Lexical and pragmatic considerations of input structures. *Computer Graphics*, 17(1):31–37, 1983.

[36] S. K. Card, J. D. Mackinlay, and G. G. Robertson. The design space of input devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 117–124. ACM, 1990.

[37] S. K. Card, J. D. Mackinlay, and G. G. Robertson. A morphological analysis of the design space of input devices. *ACM Transactions on Information Systems (TOIS)*, 9(2):99–122, 1991.

[38] G. Casiez and N. Roussel. No more bricolage!: Methods and tools to characterize, replicate and compare pointing transfer functions. In *UIST '11, ACM*, pages 603–614. ACM.

[39] G. Casiez, D. Vogel, Q. Pan, and C. Chaillou. Rubberedge: Reducing clutching by combining position and rate control with elastic feedback. In *UIST '07, ACM*, pages 129–138. ACM.

[40] F. Castanedo. A Review of Data Fusion Techniques. *The Scientific World Journal*, 2013:1–19, 2013.

[41] D. Cavens, F. Vogt, S. Fels, and M. Meitner. Interacting with the big screen: Pointers to ponder. In *CHI '02, ACM*, pages 678–679. ACM.

[42] P. Chakravorty. What is a signal? [lecture notes]. *IEEE Signal Processing Magazine*, 35(5):175–177, 2018.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

[43] W. Chang, C. Wu, R. Y. Tsai, K. C. Lin, and Y. Tseng. Eye on you: Fusing gesture data from depth camera and inertial sensors for person identification. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2021–2026, 2018.

[44] H. Chao, Y. Gu, and M. Napolitano. A survey of optical flow techniques for robotics navigation applications. *Journal of Intelligent & Robotic Systems*, 73(1-4):361–372, 2014.

[45] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1932–1939, June 2009.

[46] C. Chen, R. Jafari, and N. Kehtarnavaz. A survey of depth and inertial sensor fusion for human action recognition. *Multimedia Tools and Applications*, 76(3):4405–4425, 2017.

[47] X. Chen, J. Schwarz, C. Harrison, J. Mankoff, and S. E. Hudson. Air+ touch: interweaving touch & in-air gestures. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 519–525, 2014.

[48] V. T. Y. S. Claus Vielhauer, Sascha Schimke. Fusion strategies for speech and handwriting modalities in hci. volume 5684, 2005.

[49] P. Dalsgaard, K. Halskov, W. Mackay, N. Maiden, and J.-B. Martens. Supporting creative design processes in blended interaction spaces. In *Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition*, CandC 15, page 395–396, New York, NY, USA, 2015. Association for Computing Machinery.

[50] B. V. Dasarathy. *Decision fusion*, volume 1994. IEEE Computer Society Press Los Alamitos, CA, 1994.

[51] B. R. De Araùjo, G. Casiez, and J. A. Jorge. Mockup builder: Direct 3d modeling on and above the surface in a continuous interaction space. In *Proceedings of Graphics Interface 2012*, GI '12, pages 173–180, Toronto, Ont., Canada, Canada, 2012. Canadian Information Processing Society.

[52] Q. De Smedt, H. Wannous, and J.-P. Vandeborre. Skeleton-based dynamic hand gesture recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2016 IEEE Conference on*, pages 1206–1214. IEEE, 2016.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

[53] A. K. Dey, G. D. Abowd, and D. Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human–Computer Interaction*, 16(2-4):97–166, 2001.

[54] M. Donnarumma, B. Caramiaux, A. Tanaka, et al. Muscular interactions combining emg and mmg sensing for musical practice. In *Proceedings of NIME*. KAIST, 2013.

[55] B. Dumas, D. Lalanne, D. Guinard, R. Koenig, and R. Ingold. Strengths and weaknesses of software architectures for the rapid creation of tangible and multimodal interfaces. In *Proceedings of the 2nd international conference on Tangible and embedded interaction*, pages 47–54, 2008.

[56] H. F. Durrant-Whyte. Sensor models and multisensor integration. In *Autonomous robot vehicles*, pages 73–89. Springer, 1990.

[57] G. Essl and M. Rohs. Interactivity for mobile music-making. *Organised Sound*, 14(2):197–207, 2009.

[58] J. A. Fails and D. O. Jr. Light widgets: interacting in every-day spaces. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 63–69, 2002.

[59] G. Farneback. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pages 363–370. Springer, 2003.

[60] S. Fels, A. Gadd, and A. Mulder. Mapping transparency through metaphor: towards more expressive musical instruments. *Organised Sound*, 7(2):109–126, 2002.

[61] D. Freeman, H. Benko, M. R. Morris, and D. Wigdor. Shadowguides: Visualizations for in-situ learning of multi-touch and whole-hand gestures. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, pages 165–172, New York, NY, USA, 2009. ACM.

[62] A. L. Fuhrmann, J. Kretz, and P. Burwik. *Multi sensor tracking for live sound transformation*. Ann Arbor, MI: Michigan Publishing, University of Michigan Library, 2013.

[63] N. Fujimura. Remote furniture: interactive art installation for public space. In *ACM SIGGRAPH 2004 Emerging technologies*, page 23. 2004.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

[64] M. Funk, K. Kuwabara, and M. J. Lyons. Sonification of facial actions for musical expression. In *Proceedings of NIME*, pages 127–131, Vancouver, BC, Canada, 2005.

[65] W. W. Gaver. Technology affordances. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 79–84, 1991.

[66] G. Geiger. Using the touch screen as a controller for portable computer music instruments. In *Proceedings of NIME*, pages 61–64, Paris, France, 2006.

[67] J. Grubert, M. Heinisch, A. Quigley, and D. Schmalstieg. Multifi: Multi fidelity interaction with displays on and around the body. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3933–3942, 2015.

[68] T. Grzejszczak, M. Kawulok, and A. Galuszka. Hand landmarks detection and localization in color images. *Multimedia Tools and Applications*, 75(23):16363–16387, 2016.

[69] Y. Guiard. The kinematic chain as a model for human asymmetrical bimanual cooperation. In *Advances in Psychology*, volume 55, pages 205–228. Elsevier, 1988.

[70] J. Guna, G. Jakus, M. Pogačnik, S. Tomažič, and J. Sodnik. An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking. *Sensors*, 14(2):3702, 2014.

[71] M. Gurevich, P. Stapleton, and A. Marquez-Borbon. Style and constraint in electronic musical instruments. In *Proceedings of NIME*, pages 106–111, 2010.

[72] I. Guyon, V. Athitsos, P. Jangyodsuk, and H. J. Escalante. The chalearn gesture dataset (cgd 2011). *Machine Vision and Applications*, 25(8):1929–1951, Nov 2014.

[73] D. L. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.

[74] P. Hamilton and D. J. Wigdor. Conductor: enabling and understanding cross-device interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 2773–2782, 2014.

[75] R. I. Hartley and P. Sturm. Triangulation. *Computer vision and image understanding*, 68(2):146–157, 1997.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

[76] I. Hattwick and M. M. Wanderley. Interactive lighting in the pearl: Considerations and implementation. In *Proceedings of NIME*, pages 201–204, 2015.

[77] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1):185 – 203, 1981.

[78] S. Houben and N. Marquardt. Watchconnect: A toolkit for prototyping smartwatch-centric cross-device applications. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 1247–1256, New York, NY, USA, 2015. Association for Computing Machinery.

[79] E. Hovy and Y. Arens. When is a picture worth a thousand words? allocation of modalities in multimedia communication. In *AAAI Symposium on Human-Computer Interfaces, Stanford*, 1990.

[80] Z. Huang and J. Kong. A toolkit for prototyping tabletop-centric cross-device interaction. *International Journal of Human–Computer Interaction*, 36(6):536–552, 2020.

[81] J. Jakovich and K. Beilharz. Particletecture : Interactive granular soundspaces for architectural design. In *Proceedings of NIME*, pages 185–190, New York City, NY, United States, 2007.

[82] H. D. Jellinek and S. K. Card. Powermice and user performance. In *CHI '90, ACM*, pages 213–220, New York, NY, USA. ACM.

[83] A. R. Jensenius. Motion-sound interaction using sonification based on motiongrams. In *Proceedings of ACHI*, pages 170–175. IARIA, 2012.

[84] J. C. R. F. Jeremy Steward, Derek Lichti and S. Osis. Performance assessment and calibration of the kinect 2.0 time–of–flight range camera for use in motion capture applications. FIG Working Week 2015, May 2015.

[85] C. Jewitt, J. Bezemer, and K. O'Halloran. *Introducing multimodality*. Routledge, 2016.

[86] S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunner. The reactable: exploring the synergy between live music performance and tabletop tangible interfaces. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 139–146, 2007.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

[87] R. Jota, M. A. Nacenta, J. A. Jorge, S. Carpendale, and S. Greenberg. A comparison of ray pointing techniques for very large displays. In *GI'10, Canadian Information Processing Society*, pages 269–276. Canadian Information Processing Society.

[88] A. Just, O. Bernier, and S. Marcel. Hmm and iohmm for the recognition of mono- and bi-manual 3d hand gestures. IDIAP Research Report, 2004.

[89] M. Karam. A taxonomy of gestures in human computer interactions. 2005.

[90] T. Käster, M. Pfeiffer, and C. Bauckhage. Combining speech and haptics for intuitive and efficient navigation through image databases. In *Proceedings of the 5th international conference on Multimodal interfaces*, pages 180–187, 2003.

[91] Z. Kazi, S. Chen, M. Beitler, D. Chester, and R. Foulds. Multimodal hci for robot control: Towards an intelligent robotic assistant for people with disabilities.

[92] P. O. Keefe and G. Essl. The visual in mobile music performance. In *Proceedings of NIME*, pages 191–196, Oslo, Norway, 2011.

[93] M. Khamis, F. Alt, M. Hassib, E. von Zezschwitz, R. Hasholzner, and A. Bulling. Gazetouchpass: Multimodal authentication using gaze and touch on mobile devices. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, page 2156–2164, New York, NY, USA, 2016. Association for Computing Machinery.

[94] S. Kim and E. Paulos. Practices in the creative reuse of e-waste. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, page 2395–2404, New York, NY, USA, 2011. Association for Computing Machinery.

[95] T. K. Kim, S. F. Wong, and R. Cipolla. Tensor canonical correlation analysis for action classification. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[96] K. Kirkpatrick. Electronics need rare earths. *Commun. ACM*, 62(3):17–18, Feb. 2019.

[97] A. Kitsikidis, K. Dimitropoulos, S. Douka, and N. Grammalidis. Dance analysis using multiple kinect sensors. In *2014 international conference on computer vision theory and applications (VISAPP)*, volume 2, pages 789–795. IEEE, 2014.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

[98] B. Knowles, O. Bates, and M. Håkansson. This changes sustainable hci. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2018.

[99] R. Kopper, D. A. Bowman, M. G. Silva, and R. P. McMahan. A human motor behavior model for distal pointing tasks. *Int. J. Hum.-Comput. Stud.'10, Academic Press, Inc.*, 68(10):603–615.

[100] M. Korzetz, R. Kühn, K. Kegel, L. Georgi, F.-W. Schumann, and T. Schlegel. Milkyway: A toolbox for prototyping collaborative mobile-based interaction techniques. In *International Conference on Human-Computer Interaction*, pages 477–490. Springer, 2019.

[101] T. Kroeger, R. Timofte, D. Dai, and L. V. Gool. Fast optical flow using dense inverse search. In *Proceedings of ECCV*, 2016.

[102] T. Kroeger, R. Timofte, D. Dai, and L. Van Gool. Fast optical flow using dense inverse search. In *European Conference on Computer Vision*, pages 471–488. Springer, 2016.

[103] D. Lahat, T. Adali, and C. Jutten. Multimodal data fusion: an overview of methods, challenges, and prospects. *Proceedings of the IEEE*, 103(9):1449–1477, 2015.

[104] F. Landragin, N. Bellalem, and L. Romary. Referring to objects with spoken and haptic modalities. In *Fourth IEEE International Conference on Multimodal Interfaces*, pages 99–104, 2002.

[105] H.-K. Lee and J. H. Kim. An hmm-based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):961–973, Oct 1999.

[106] Z. Lin, Z. Jiang, and L. S. Davis. Recognizing actions by shape-motion prototype trees. In *2009 IEEE 12th International Conference on Computer Vision*, pages 444–451, Sept 2009.

[107] M. Liu, M. Nancel, and D. Vogel. Gunslinger: Subtle arms-down mid-air interaction. In *UIST'15, ACM*, pages 63–71. ACM.

[108] J. Llinas, C. Bowman, G. Rogova, A. Steinberg, E. Waltz, and F. White. Revisiting the jdl data fusion model ii. Technical report, SPACE AND NAVAL WARFARE SYSTEMS COMMAND SAN DIEGO CA, 2004.

[109] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

[110] R. C. Luo, C.-C. Yih, and K. L. Su. Multisensor fusion and integration: approaches, applications, and future research directions. *IEEE Sensors journal*, 2(2):107–119, 2002.

[111] S. Malik, A. Ranjan, and R. Balakrishnan. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In *UIST '05, ACM*, pages 43–52, New York, NY, USA. ACM.

[112] J. Malloch and M. M. Wanderley. The t-stick: From musical interface to musical instrument. In *Proceedings of NIME*, pages 66–70, 2007.

[113] T. Mankowski, J. Tomczynski, and P. Kaczmarek. Cie-dataglove, a multi-imu system for hand posture tracking. In *International Conference Automation*, pages 268–276. Springer, 2017.

[114] S. Marcel. Sebastien marcel - hand posture and gesture datasets. `http://www.idiap.ch/resource/gestures`, 2020. [Online; accessed 30-April-2020].

[115] S. Marcel, O. Bernier, J.-E. Viallet, and D. Collobert. Hand gesture recognition using input-output hidden markov models. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 456–461. IEEE, 2000.

[116] N. Marquardt, R. Diaz-Marino, S. Boring, and S. Greenberg. The proximity toolkit: Prototyping proxemic interactions in ubiquitous computing ecologies. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 315–326, New York, NY, USA, 2011. ACM.

[117] N. Marquardt, R. Jota, S. Greenberg, and J. A. Jorge. The continuous interaction space: Interaction techniques unifying touch and gesture on and above a digital surface. In *Proceedings of the 13th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part III*, INTERACT'11, pages 461–476, Berlin, Heidelberg, 2011. Springer-Verlag.

[118] A. Martinelli. Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. *IEEE Transactions on Robotics*, 28(1):44–60, 2011.

[119] D. C. McCallum and P. Irani. Arc-pad: Absolute+relative cursor positioning for large displays with a mobile touchscreen. In *UIST '09, ACM*, pages 153–156. ACM.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

[120] D. Merrill, J. Kalanithi, and P. Maes. Siftables: towards sensor network user interfaces. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 75–78, 2007.

[121] mi.mi. Mi.mu gloves. `http://mimugloves.com`, 2017. [Online; accessed 30-December-2016].

[122] F. M. Mirzaei and S. I. Roumeliotis. 1| a kalman filter-based algorithm for imu-camera calibration. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2427–2434. IEEE.

[123] T. Mitchell and I. Heap. Soundgrasp: A gestural interface for the performance of live music.

[124] T. P. Moran. The command language grammar: A representation for the user interface of interactive computer systems. *International journal of man-machine studies*, 15(1):3–50, 1981.

[125] A. Mulder and S. Fels. Sound sculpting: Manipulating sound through virtual sculpting. In *Proc. of the 1998 Western Computer Graphics Symposium*, pages 15–23, 1998.

[126] B. A. Myers, R. Bhatnagar, J. Nichols, C. H. Peck, D. Kong, R. Miller, and A. C. Long. Interacting at a distance: Measuring the performance of laser pointers and other devices. In *CHI '02, ACM*, pages 33–40. ACM.

[127] H.-H. Nagel. Displacement vectors derived from second-order intensity variations in image sequences. *Computer Vision, Graphics, and Image Processing*, 21(1):85–117, 1983.

[128] M. Nancel, O. Chapuis, E. Pietriga, X.-D. Yang, P. P. Irani, and M. Beaudouin-Lafon. High-precision pointing on large wall displays using small handheld devices. In *CHI '13, ACM*, pages 831–840, New York, NY, USA. ACM.

[129] M. Nancel, D. Vogel, and E. Lank. Clutching is not (necessarily) the enemy. In *CHI'15, ACM*, pages 4199–4202, New York, NY, USA. ACM.

[130] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota. Fingerio: Using active sonar for fine-grained finger tracking. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 1515–1525, New York, NY, USA, 2016. ACM.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

[131] L. Nigay and J. Coutaz. A design space for multimodal systems: Concurrent processing and data fusion. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, page 172–178, New York, NY, USA, 1993. Association for Computing Machinery.

[132] L. Nigay and J. Coutaz. A generic platform for addressing the multimodal challenge. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, page 98–105, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[133] A. Nijholt et al. Multimodal integration of haptics, speech, and affect in an educational environment. In *Proceedings of the International Conference on Computing, Communications and Control Technologies*, volume 2, pages 94–97, 2004.

[134] B. O'Flynn, J. T. Sanchez, J. Connolly, J. Condell, K. Curran, P. Gardiner, and B. Downes. Integrated smart glove for hand motion monitoring. In *The Sixth International Conference on Sensor Device Technologies and Applications*, 2015.

[135] J.-Y. Oh and W. Stuerzlinger. Laser pointers as collaborative pointing devices, 2002.

[136] M. Ojanen, J. Suominen, T. Kallio, and K. Lassfolk. Design principles and user interfaces of erkki kurenniemi's electronic musical instruments of the 1960's and 1970's. In *Proceedings of NIME*, pages 88–93, New York, NY, USA, 2007. ACM.

[137] A. V. Oppenheim, A. S. Willsky, and S. Hamid. Signals and systems, processing series, 1997.

[138] E. W. Pedersen and K. Hornbæk. Tangible bots: Interaction with active tangibles in tabletop interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, page 2975–2984, New York, NY, USA, 2011. Association for Computing Machinery.

[139] J.-M. Pelletier. Sonified motion flow fields as a means of musical expression. In *NIME*, pages 158–163, 2008.

[140] B. Penelle and O. Debeir. Multi-sensor data fusion for hand tracking using kinect and leap motion. In *Proceedings of the 2014 Virtual Reality International Conference*, VRIC '14, pages 22:1–22:7, New York, NY, USA, 2014. ACM.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

[141] J. S. Pierce, A. S. Forsberg, M. J. Conway, S. Hong, R. C. Zeleznik, and M. R. Mine. Image plane interaction techniques in 3d immersive environments. In *I3D, ACM*, pages 39–ff. ACM.

[142] K. Pietroszek and E. Lank. Clicking blindly: Using spatial correspondence to select targets in multi-device environments. In *MobileHCI '12, ACM*, pages 331–334, New York, NY, USA. ACM.

[143] P. K. Pisharady and M. Saerbeck. Recent methods and databases in vision-based hand gesture recognition: A review. *Computer Vision and Image Understanding*, 141:152 – 165, 2015. Pose & Gesture.

[144] P. K. Pisharady, P. Vadakkepat, and A. P. Loh. Attention based detection and recognition of hand postures against complex backgrounds. *International Journal of Computer Vision*, 101(3):403–419, Feb 2013.

[145] Qiaohui Zhang, K. Go, A. Imamiya, and Xiaoyang Mao. Designing a robust speech and gaze multimodal system for diverse users. In *Proceedings Fifth IEEE Workshop on Mobile Computing Systems and Applications*, pages 354–361, 2003.

[146] H. Rateau, Y. Rekik, L. Grisoni, and J. Jorge. Talaria: Continuous drag and drop on a wall display. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces*, ISS '16, pages 199–204, New York, NY, USA, 2016. ACM.

[147] S. S. Rautaray and A. Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, 43(1):1–54, Jan 2015.

[148] C. Rieger and T. A. Majchrzak. Conquering the mobile device jungle: Towards a taxonomy for app-enabled devices. In *WEBIST*, pages 332–339, 2017.

[149] O. Rodrigues. Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire. *Journal de mathématiques pures et appliquées 1re série*, pages 380–440, 1840.

[150] T. B. Rodrigues, C. Ó. Catháin, D. Devine, K. Moran, N. E. O'Connor, and N. Murray. An evaluation of a 3d multimodal marker-less motion analysis system. In *Proceedings of the 10th ACM Multimedia Systems Conference*, pages 213–221, 2019.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

[151] M. Rohs and G. Essl. Camus 2 – optical flow and collaboration in camera phone music performance. In *Proceedings of NIME*, pages 160–163, New York City, NY, United States, 2007.

[152] M. Rohs, G. Essl, and M. Roth. Camus: Live music performance using camera phones and visual grid tracking. In *NIME*, pages 31–36, Paris, France, 2006.

[153] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Computer Vision – ECCV 2006*, pages 430–443, Berlin, Heidelberg, 2006.

[154] S. Sabatelli, M. Galgani, L. Fanucci, and A. Rocchi. A double stage kalman filter for sensor fusion and orientation tracking in 9d imu. In *2012 IEEE Sensors Applications Symposium Proceedings*, pages 1–5, 2012.

[155] D. Salber, A. K. Dey, and G. D. Abowd. Ubiquitous computing: Defining an hci research agenda for an emerging interaction paradigm. Technical report, Georgia Institute of Technology, 1998.

[156] S. Schaffer and N. Reithinger. Conversation is multimodal: Thus conversational user interfaces should be as well. In *Proceedings of the 1st International Conference on Conversational User Interfaces*, CUI '19, New York, NY, USA, 2019. Association for Computing Machinery.

[157] G. Schall, D. Wagner, G. Reitmayr, E. Taichmann, M. Wieser, D. Schmalstieg, and B. Hofmann-Wellenhof. Global pose estimation using multi-sensor fusion for outdoor augmented reality. In *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, pages 153–162, 2009.

[158] A. Schmidt, M. Beigl, and H.-W. Gellersen. There is more to context than location. *Computers & Graphics*, 23(6):893–901, 1999.

[159] C. A. Scholes et al. Finite elements. *Chemistry in Australia*, (Sep/Oct 2019):24, 2019.

[160] N. Sebe. Multimodal interfaces: Challenges and perspectives. *Journal of Ambient Intelligence and smart environments*, 1(1):23–30, 2009.

[161] M. Serrano, K. Hasan, B. Ens, X.-D. Yang, and P. Irani. Smartwatches+ head-worn displays: the" new" smartphone. ACM SIGCHI, 2015.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

[162] R. Sharma, V. I. Pavlović, and T. S. Huang. Toward multimodal human–computer interface. In *Advances in image processing and understanding: a festschrift for Thomas S Huang*, pages 349–365. World Scientific, 2002.

[163] S. Shen, H. Wang, and R. Roy Choudhury. I am a smartwatch and i can track my user's arm. In *Proceedings of the 14th annual international conference on Mobile systems, applications, and services*, pages 85–96, 2016.

[164] X. Shen, G. Hua, L. Williams, and Y. Wu. Dynamic hand gesture recognition: An exemplar-based approach from motion divergence fields. *Image and Vision Computing*, 30(3):227 – 235, 2012. Best of Automatic Face and Gesture Recognition 2011.

[165] S. Siddhpuria, S. Malacria, M. Nancel, and E. Lank. Pointing at a distance with everyday smart devices. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 173:1–173:11, New York, NY, USA, 2018. ACM.

[166] R. Sodhi, H. Benko, and A. Wilson. Lightguide: Projected visualizations for hand movement guidance. In *Proceedings of ACM CHI*, pages 179–188, 2012.

[167] J. Song, G. Sörös, F. Pece, S. R. Fanello, S. Izadi, C. Keskin, and O. Hilliges. In-air gestures around unmodified mobile devices. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 319–329, 2014.

[168] M. Stevens and E. D'Hondt. Crowdsourcing of pollution data using smartphones. In *Workshop on Ubiquitous Crowdsourcing*, pages 1–4, 2010.

[169] K. Takashima, N. Shinshi, and Y. Kitamura. Exploring boundless scroll by extending motor space. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '15, pages 557–566, New York, NY, USA, 2015. ACM.

[170] T. Y. Tang, M. Y. He, and V. L. Cao. "one doesn't fit all": A comparative study of various finger gesture interaction methods. In A. Marcus, editor, *Design, User Experience, and Usability: Technological Contexts*, pages 88–97, Cham, 2016. Springer International Publishing.

[171] P. Truillet. A taxonomy of physical contextual sensors. In *International Conference on Human-Computer Interaction*, pages 982–989. Springer, 2007.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

[172] J. Uijlings, I. Duta, N. Rostamzadeh, and N. Sebe. Realtime video classification using dense hof/hog. In *ICMR*, 2014.

[173] D. Vieira, J. D. Freitas, C. Acarturk, A. Teixeira, L. Sousa, S. Silva, S. Candeias, and M. S. Dias. "read that article": Exploring synergies between gaze and speech interaction. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*, ASSETS '15, page 341–342, New York, NY, USA, 2015. Association for Computing Machinery.

[174] D. Vogel and R. Balakrishnan. Distant freehand pointing and clicking on very large, high resolution displays. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, UIST '05, pages 33–42, New York, NY, USA, 2005. ACM.

[175] C. Wacharamanotham, K. Todi, M. Pye, and J. Borchers. Understanding finger input above desktop devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, page 1083–1092, New York, NY, USA, 2014. Association for Computing Machinery.

[176] J. Wagner, M. Nancel, S. G. Gustafson, S. Huot, and W. E. Mackay. Body-centric design space for multi-surface interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, page 1299–1308, New York, NY, USA, 2013. Association for Computing Machinery.

[177] L. Wald. Some terms of reference in data fusion. *IEEE Transactions on geoscience and remote sensing*, 37(3):1190–1193, 1999.

[178] G. Wang. Ocarina: Designing the iphone's magic flute. *Computer Music Journal*, 38(2):8–21, 2014.

[179] M. Weiser. The computer for the 21st century. *ACM SIGMOBILE mobile computing and communications review*, 3(3):3–11, 1999.

[180] A. Wexelblat. Natural gesture in virtual environments. In *Virtual Reality Software And Technology*, pages 5–16. World Scientific, 1994.

[181] F. White. Data fusion lexicon: data fusion subpanel of the joint directors of laboratories technical panel for c3. *IEEE Trans.: San Diego, CA, USA*, 1991.

[182] A. D. Wilson and H. Benko. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proceedings of the*

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

23nd Annual ACM Symposium on User Interface Software and Technology, UIST '10, page 273–282, New York, NY, USA, 2010. Association for Computing Machinery.

[183] A. D. Wilson and E. Cutrell. Flowmouse: A computer vision-based pointing and gesture input device. In M. F. Costabile and F. Paternò, editors, *Human-Computer Interaction - INTERACT 2005*, pages 565–578, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[184] J. O. Wobbrock, B. A. Myers, and H. H. Aung. The performance of hand postures in front- and back-of-device interaction for mobile computing. *Int. J. Hum.-Comput. Stud.*, 66(12):857–875, Dec. 2008.

[185] P. C. Wong, H. Fu, and K. Zhu. Back-mirror: Back-of-device one-handed interaction on smartphones. In *SIGGRAPH ASIA 2016 Mobile Graphics and Interactive Applications*, SA '16, pages 13:1–13:2, New York, NY, USA, 2016. ACM.

[186] H. Wu, M. Siegel, and S. Ablay. Sensor fusion for context understanding. In *IMTC/2002. Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference (IEEE Cat. No. 00CH37276)*, volume 1, pages 13–17. IEEE, 2002.

[187] Y. Wu and T. S. Huang. Vision-based gesture recognition: A review. In A. Braffort, R. Gherbi, S. Gibet, D. Teil, and J. Richardson, editors, *Gesture-Based Communication in Human-Computer Interaction*, pages 103–115, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[188] X. Xiao, T. Han, and J. Wang. Lensgesture: Augmenting mobile interactions with back-of-device finger gestures. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, pages 287–294, New York, NY, USA, 2013. ACM.

[189] N. Xiong and P. Svensson. Multi-sensor management for information fusion: issues and approaches. *Information Fusion*, 3(2):163 – 186, 2002.

[190] W. Yamada, H. Manabe, H. Hakoda, K. Ochiai, and D. Ikeda. Camtrackpoint: Pointing stick for mobile device using rear camera. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, UbiComp '17, pages 321–324, New York, NY, USA, 2017. ACM.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr

Thèse de Cagan Arslan, Université de Lille, 2020

[191] S. Yang and Y. Guan. Audio–visual perception-based multimodal hci. *The Journal of Engineering*, 2018(4):190–198, 2018.

[192] S. H. Yoon, K. Huo, and K. Ramani. Tmotion: Embedded 3d mobile input using magnetic sensing technique. In *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '16, pages 21–29, New York, NY, USA, 2016. ACM.

[193] W. Yu and S. Brewster. Comparing two haptic interfaces for multimodal graph rendering. In *Proceedings 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. HAPTICS 2002*, pages 3–9. IEEE, 2002.

[194] J. Yves Bouguet. Pyramidal implementation of the lucas kanade feature tracker. *Intel Corporation, Microprocessor Research Labs*, 2000.

[195] Y. Zhai, G. Zhao, T. Alatalo, J. Heikkilä, T. Ojala, and X. Huang. Gesture interaction for wall-sized touchscreen display. In *UbiComp'13, ACM*, pages 175–178. ACM.

[196] Z. Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000.

© 2020 Tous droits réservés.

lilliad.univ-lille.fr